

## EFFICIENT EXPECTED IMPROVEMENT ESTIMATION FOR CONTINUOUS MULTIPLE RANKING AND SELECTION

Michael Pearce  
Juergen Branke

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

### ABSTRACT

This paper considers the problem of identifying the best of a discrete set of alternatives for each of a set of correlated problem instances. We assume that the instances can be described by a set of continuous features and the performance of a particular alternative on a particular problem instance can only be estimated from noisy samples. A possible application is in manufacturing, where we would like to identify the best dispatching rule to be used depending on shop floor conditions, and performance is estimated via stochastic simulation. We propose three myopic sequential sampling methods to collect information, and in particular focus on an efficient estimation of the expected improvement which requires integration over the continuous space of problem instances. Empirical tests show that our method of estimating expected improvement is indeed more efficient than standard Monte Carlo integration, and that our method significantly outperforms a recently published alternative sampling method.

### 1 INTRODUCTION

Ranking and selection is the problem of identifying the alternative with the best expected performance from a set of alternatives, when the quality of an alternative can only be observed through expensive noisy sampling. The problem we consider is an extension of the standard ranking and selection setting, where there is a distribution of problem instances described by some continuous feature values, and the goal is to identify the best alternative,  $a$ , depending on the problem instance,  $x$ . In other words, we are looking for a mapping  $S(x) : X \rightarrow A$  from problem instance features to alternative that maximizes the expected performance over all instances,  $F = \int \mathbb{E}[f(S(x), x)] \mathbb{P}[x] dx$ , where  $f(a, x)$  is the stochastic performance of alternative  $a$  on problem instance  $x$ , and  $\mathbb{P}[x]$  is the probability of encountering a particular problem instance  $x$ . If there is only one  $x$  the problem reduces to traditional ranking and selection.

We assume that performance values are correlated over problem instances, i.e., that applying an alternative  $a$  to two similar problem instances results in similar performance, but that the performances of different alternatives are independent. To identify  $S$ , we can sequentially collect information by sampling performances  $f(a, x)$ , and we would like to allocate a given sampling budget  $N$  such that  $F$  is maximized.

In traditional ranking and selection under input uncertainty it is assumed the user aims to find the single optimal  $a$  averaged over all inputs  $x$ . The problem we consider may also be viewed as ranking and selection under Input Uncertainty where the user aims to find the optimal alternative  $a$  for each possible input  $x$  distributed according  $\mathbb{P}[x]$ .

This setting has various real-world applications. For example in manufacturing, we may want to understand which dispatching rule works best depending on the shop floor conditions, so that we can pick the best rule in each condition encountered (Heger et al. 2016). Or in online advertising, we may want

to understand which advert is most effective depending on some information on the viewer (such as their search history).

We model the problem using Gaussian Processes and extend an earlier paper (Pearce and Branke 2017) in which we have proposed an algorithm to identify the best alternatives for a *discrete* set of correlated problem instances to the case of *continuous* instance distributions. In particular, we focus here on ways to compute the expected improvement efficiently. We also demonstrate that our approach outperforms a recently published alternative method on a synthetic benchmark problem.

The paper is structured as follows. Section 2 surveys previous work, in Section 3 we provide a problem formulation, and our new sampling methods are proposed in Section 4. We evaluate the sampling methods on a synthetic benchmark problem in Section 5 and conclude in Section 6.

## 2 LITERATURE REVIEW

The problem of identifying the best alternative when the performance of an alternative must be inferred from noisy observations is usually called Ranking and Selection. Many methods have been proposed that sequentially allocate sample observations to the different alternatives in order to maximize the probability of correctly selecting the best alternative, or minimizing the expected opportunity cost. Branke, Chick, and Schmidt (2007) provide a review and comparison of several ranking and selection methods, a more recent survey focusing on algorithms derived under the Bayesian framework has been written by Chen et al. (2015). Building on Gupta and Miescke (1996), the Knowledge Gradient policy of Frazier, Powell, and Dayanik (2008) and the myopic sampling policy of Chick, Branke, and Schmidt (2010) allocate the next sample where they myopically expect the largest improvement of performance in the next step. This method was extended to the case when decision outcomes are correlated by Frazier, Powell, and Dayanik (2009) where observing a high outcome from one alternative suggests similar alternatives would also perform well.

When the search space is continuous, a Gaussian Process, or Kriging, are often used as a statistical model of the objective function given the data collected. This can then be used for optimization where a user has limited data and new observations are expensive to collect, see Shahriari et al. (2016) for a review of Bayesian Optimization methods. Possibly the most famous method in this category is the Efficient Global Optimization algorithm (EGO) of Jones, Schonlau, and Welch (1998) that uses a Gaussian Process Regression model and sequentially allocates samples by maximizing an expected improvement of a new function evaluation over the current best evaluation. The Sequential Kriging Optimization method of Huang et al. (2006) extends the EGO method to account for noise and Scott, Frazier, and Powell (2011) extend the discrete Knowledge Gradient policy to the continuous case which is myopically optimal under certain conditions.

When multiple finite Ranking and Selection problems have to be solved, correlations among these problems may be exploited to speed up convergence. Pearce and Branke (2017) provide myopic methods for allocating samples where there is correlation between tasks but they do not account for correlation across decision alternatives. Hu and Ludkovsk (2017) consider the same problem as this work where the range of tasks is continuous and provide a sampling method that maximizes the model's prediction of the upper bound.

A few papers extend the idea of Bayesian Optimization to the case where several related optimization problems have to be solved sequentially. Morales-Enciso and Branke (2015) consider the optimization of a changing objective function with EGO, and propose three different ways to re-use information gathered on previous objective functions to speed up optimization of the current objective function: using the old posterior mean as the new prior mean, treating old samples as noisy, and treating time as an extra input dimension to the learned function. The latter idea is also used by Poloczek, Wang, and Frazier (2016) to warm-start Bayesian Optimization, however with the Knowledge Gradient as the acquisition function for the current optimization task. A similar problem has been much studied in the field of Contextual Multi-Arm Bandits, an extension of the multi-armed bandit problem where the best arm depends on a context that is randomly changing with time and sampling policies must aim to maximize cumulative reward, see Zhou

(2015) for a survey. Krause and Ong (2011) use Gaussian Processes with inputs that are both decision variables and context variables and propose an upper confidence bound acquisition function to find the best arm for each context.

In this work we approach the same problem considered by Hu and Ludkovsk (2017), i.e., given a finite set of independent response surfaces, one for each alternative, over a continuous domain, such as problem features, the aim is to learn about the best surface/alternative for each point in the domain. We provide three new algorithms that either perform equivalently or better than the best method proposed by Hu and Ludkovsk (2017). We especially focus on how to efficiently estimate the expected improvement over a continuous domain.

### 3 PROBLEM FORMULATION

There exists a continuous distribution of problem instances with features  $x \in X \subset \mathbb{R}^D$ , and the distribution  $\mathbb{P}[x]$  of instances in the feature space is known. There exists a discrete set of alternatives  $A$ . Executing a given alternative  $a \in A$  on a given problem instance  $x \in X$  yields a stochastic performance measurement  $Y_a(x) = f(a, x) = \theta_a(x) + \varepsilon$  where  $\theta_a(x) : \mathbb{R}^D \rightarrow \mathbb{R}$  is a deterministic latent function and  $\varepsilon \sim N(0, \sigma_{\varepsilon, a})$  is independent and identically distributed white noise with known variance (which in practice is measured). The objective of the user is to find a classifier, or mapping, from problem instance to the best alternative  $S(x) : X \rightarrow A$  that approximates  $S^*(x) = \operatorname{argmax}_a \theta_a(x)$  given a finite budget of  $N$  performance measurements. The aim is to find a mapping that maximizes the overall expected performance across all problem instances

$$\text{True Performance} = F = \int_X \theta_{S(x)}(x) \mathbb{P}[x] dx. \tag{1}$$

Individual performance measurements may be collected sequentially so that after  $n$  measurements the user may determine the alternative and problem instance for the  $(n + 1)^{th}$  sample.

If there was only a single problem instance, the problem reduces to finding the alternative  $a$  with the highest  $\theta_a$ , a standard ranking and selection problem. The formulation above extends ranking and selection to a continuous range of correlated ranking and selection problems. The distribution of problems  $\mathbb{P}[x]$  ensures that frequent problems (high  $\mathbb{P}[x]$ ) contribute to performance more than rare problems (low  $\mathbb{P}[x]$ ) and as such a user would prefer to have a more accurate mapping  $S(x)$  for common problems. We refer to  $\mathbb{P}[x]$  as the problem instance density however, without loss of generality, any user defined weight function,  $W(x)$ , can be incorporated to give priority to particular instances  $x$ .

A practical application may be the algorithm selection problem where one would like to derive a mapping that specifies the most suitable algorithm depending on some features of the problem instance. In this case,  $x$  are the features of a given problem instance,  $\mathbb{P}[x]$  is the distribution of problem instances throughout the feature space. There is a set of  $A$  alternative algorithms and  $\theta_a(x)$  is the average performance of an algorithm  $a$  on instance with features  $x$ . A user aims to construct the mapping  $S(x)$  from problem features to best algorithm.

### 4 SAMPLING METHODS

We propose to use independent Gaussian Process regression models for each  $a$  to predict expected performance  $\theta_a(x)$  given instance features  $x$ . Gaussian Process regression is a popular Bayesian regression method that has been used successfully to optimize expensive black box functions where only abstract prior information is known about the black box such as smoothness or periodicity.

Given a dataset of (instance, alternative, performance) triplets  $(x^1, a^1, y^1), \dots, (x^n, a^n, y^n)$ , we define the subset of (instance, performance) pairs associated with alternative  $a$  as  $X_a^n, Y_a^n$ . We define the sigma algebra generated by the data as  $\mathcal{F}^n$  that defines the sequence of filtrations used to condition the distributions of  $\theta_1(x), \dots, \theta_A(x)$  on the data. For a single alternative  $a$ , a user defines prior mean and covariance functions  $\mu_a^0(x), k_a^0(x, x')$  that can be used to incorporate prior knowledge about the underlying performance  $\theta_a(x)$ ,

however there also exists common default settings such as zero mean and squared exponential kernel. After observing data the posterior distribution of  $\theta_a(x)$ , the Gaussian Processes Regression, is given by

$$\mathbb{E}[\theta_a(x)|\mathcal{F}^n] = \mu_a^0(x) + k_a^0(x, X_a^n)(k_a^0(X_a^n, X_a^n) + \mathbb{I}\sigma_{a\varepsilon}^2)^{-1}(Y_a^n - \mu^0(X_a^n)), \quad (2)$$

$$\text{Cov}[\theta_a(x), \theta_a(x')|\mathcal{F}^n] = k_a^0(x, x') - k_a^0(x, X_a^n)(k_a^0(X_a^n, X_a^n) + \mathbb{I}\sigma_{a\varepsilon}^2)^{-1}k_a^0(X_a^n, x'), \quad (3)$$

where  $k_a^0(X_a^n, X_a^n)$  is the  $n \times n$  matrix with elements determined by evaluating the prior covariance function between all possible pairs of elements in  $X_a^n$  and  $\mathbb{I}$  is the  $n \times n$  identity matrix. We can then construct a mapping for any instance  $x$  by taking the alternative with the best predicted performance

$$S(x) = \operatorname{argmax}_a \mu_a^n(x)$$

and the model estimate of performance,  $P^n$ , of such a mapping is computed by

$$P^n = \int_X \mu_{S(x)}^n(x) \mathbb{P}[x] dx = \int_X \max_a \mu_a^n(x) \mathbb{P}[x] dx \quad (4)$$

which we aim to optimize as a surrogate for optimizing  $F$ , however this cannot be written analytically for arbitrary  $\mathbb{P}[x]$ . To derive sampling methods we next consider the marginal improvement in the surrogate objective. Given the next data point  $x^{n+1}$ ,  $a^{n+1}$  and  $y^{n+1}$ , the observed improvement due to the new sample is

$$I((x, a, y)^{n+1}) = P^{n+1}((x, a, y)^{n+1}) - P^n = \int_X \left( \max_a \mu_a^{n+1}(x) - \max_{a'} \mu_{a'}^n(x) \right) \mathbb{P}[x] dx. \quad (5)$$

After  $n$  samples,  $x^{n+1}$  and  $a^{n+1}$  are chosen by the user and a random  $y^{n+1}$  is observed. Inspired by myopic optimality we aim to allocate samples  $(x, a)^{n+1}$  to maximize the expectation of  $I$  over  $y^{n+1}$ . For a given  $(x, a)^{n+1}$  the Gaussian Process gives a predictive distribution of  $y^{n+1}$  thus the statistical distribution of  $\mu_a^{n+1}(x)$ ,  $P^{n+1}$  and  $I$  can in theory be calculated, we next derive these distributions.

Given only the  $n$  samples and the user's choice of  $(x, a)^{n+1}$ , the prior predictive distribution of the next observation is

$$y^{n+1} \sim N(\theta_a(x^{n+1}), \sigma_{a,\varepsilon}^2) = N(\mu_a^n(x^{n+1}), k_a^n(x^{n+1}, x^{n+1}) + \sigma_{a,\varepsilon}^2) \quad (6)$$

Equation 2 gives the formula for the current posterior mean after  $n$  samples, the formula for the posterior after one sample allocated to alternative  $a$  is

$$\mathbb{E}[\theta_a(x)|\mathcal{F}^1] = \mu_a^0(x) + \frac{k_a^0(x, x_a^1)}{k_a^0(x_a^1, x_a^1) + \sigma_{a\varepsilon}^2} (y_a^1 - \mu^0(x_a^1)). \quad (7)$$

Replacing  $\mu_a^0(x)$  and  $\mu_a^1(x)$  with  $\mu_a^n(x)$  and  $\mu_a^{n+1}(x)$ , and the same for covariances, yields a consistent updating formula for the posterior mean. For each new observed  $y^{n+1}$ , only the sampled alternative  $a^{n+1}$  is changed,

$$\mathbb{E}[\theta_a(x)|\mathcal{F}^{n+1}] = \mu_a^{n+1}(x) = \begin{cases} \mu_a^n(x) + \frac{k_a^n(x, x_a^{n+1})}{k_a^n(x_a^{n+1}, x_a^{n+1}) + \sigma_{a\varepsilon}^2} (y_a^{n+1} - \mu^n(x_a^{n+1})) & a = a^{n+1} \\ \mu_a^n(x) & a \neq a^{n+1} \end{cases} \quad (8)$$

Combining the distribution of  $y^{n+1}$  with the updating formula for the mean yields the predictive distribution

$$\mu_a^{n+1}(x) = \mu_a^n(x) + \tilde{\sigma}_a^n(x; x^{n+1})Z \quad (9)$$

where  $Z \sim N(0, 1)$  is the stochastic z-score of  $y^{n+1}$  value on its predictive distribution and

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

is a deterministic function of  $x$ , parametrized by  $x^{n+1}$ , that is an additive update to the posterior mean caused by the new sample and the size of the update is scaled by the stochastic  $Z$ . The improvement  $I$  is an integral over all  $x \in X$ . Next, we calculate the improvement for a single instance in the integral due to the new sample  $y^{n+1}$ . Given instance  $x$ , we define the best predicted alternative as  $\max_a \mu_a(x) = \mu_{(1)}^n(x)$  and for the following equation we drop  $(x)$ . The expected improvement in estimated performance is given by the integrand of Equation 5

$$\mathbb{E}[\max_{a'} \mu_{a'}^{n+1} - \mu_{(1)}^n] = \mathbb{E}[\max\{\max_{a' \neq a^{n+1}} \mu_{a'}^n - \mu_{(1)}^n, \mu_{a^{n+1}}^n - \mu_{(1)}^n + Z\tilde{\sigma}_{a^{n+1}}\}] \quad (10)$$

$$= \begin{cases} \mathbb{E}[\max\{\Delta_a, Z\tilde{\sigma}_a\}] & a^{n+1} = (1) \\ \mathbb{E}[\max\{0, \Delta_a + Z\tilde{\sigma}_a\}] & a^{n+1} \neq (1) \end{cases} \quad (11)$$

$$= \Delta_a \Phi(\Delta_a/\tilde{\sigma}_a) - \tilde{\sigma}_a \phi(\Delta_a/\tilde{\sigma}_a) \quad (12)$$

where  $\Delta_a(x) = -|\mu_a^n(x) - \max_{a' \neq a^{n+1}} \mu_{a'}^n(x)|$ ,  $\tilde{\sigma}_a = \tilde{\sigma}_a^n(x, x^{n+1})$  and  $\phi(x)$  and  $\Phi(x)$  are the standard normal density and cumulative functions respectively. The above expected improvement for a single  $x$  is the same as the well known expected improvement for the EGO algorithm which uses the prior predictive distribution of a new sample to give expected improvement over past samples. Here it is measuring the expected improvement in the mean of one Gaussian process over the mean of other independent Gaussian Processes. Finally the expected improvement, EI, over all  $x \in X$  is the integral of the single  $x$  improvement

$$EI(x, a) = \mathbb{E}_{y^{n+1}}[I(x, a, y^{n+1})] \quad (13)$$

$$= \int_{x' \in X} (\Delta_a(x')\Phi(\Delta_a(x')/\tilde{\sigma}_a^n(x', x)) - \tilde{\sigma}_a^n(x', x)\phi(\Delta_a(x')/\tilde{\sigma}_a^n(x', x))) \mathbb{P}[x'] dx' \quad (14)$$

In the following subsections, we propose several sampling policies for determining  $x^{n+1}$  and  $a^{n+1}$  based on the above expected improvement calculation.

#### 4.1 Local Expected Value of Improvement

As a cheap simple heuristic for maximizing  $EI(x, a)$  we propose the Local Expected Value of Improvement (LEVI) procedure which determines the next (instance, alternative) pair to sample based on the improvement that the sampled instance alone contributes to the global improvement  $EI(x, a)$ . The LEVI method allocates the sample to maximize the argument to the integral:

$$LEVI(x, a) = \mathbb{E}[\max_{a'} \mu_{a'}^{n+1}(x) - \max_{a''} \mu_{a''}^n(x)] \mathbb{P}[x] \quad (15)$$

$$= \left( \Delta_a(x)\Phi(\Delta_a(x)/\tilde{\sigma}_a(x; x)) - \tilde{\sigma}_a(x)\phi(\Delta_a(x)/\tilde{\sigma}_a(x; x)) \right) \mathbb{P}[x] \quad (16)$$

The LEVI method treats  $x$  as a single ranking and selection problem and samples the single problem that has the joint largest improvement and probability  $\mathbb{P}[x]$ . It tends to allocate a sample to (instance, alternative) pairs for which the difference in prediction between the chosen alternative and the best alternative is small,  $\Delta_a(x)$  is small, where the possible change in prediction,  $\tilde{\sigma}_a(x; x)$ , is high and where the instance density,  $\mathbb{P}[x]$ , is high. Each call to  $LEVI(x, a)$  requires a call to all posterior means  $\mu_1^n(x), \dots, \mu_A^n(x)$  which have complexity  $O(n_1), \dots, O(n_A)$  and one call to the posterior covariance  $k_a^n(x, x)$  which has complexity  $O(n_a^2)$  where  $n_a$  is the number of the  $n$  samples so far allocated to alternative  $a$ . This leading complexity is equivalent to one call for the EGO expected improvement, however for this problem the acquisition function must be optimized once for each alternative in order to allocate a single sample. Optimization of  $LEVI(x, a)$  can be done by many methods and in Section 5 we use the squared exponential kernel that gives smooth predictions and Nelder-Mead algorithm with multiple starts.

The LEVI sampling method is asymptotically optimal meaning that given an infinite sampling budget the true optimal mapping  $S^*(x)$  will be determined perfectly.

### 4.2 Regional Expected Value of Improvement

The Regional Expected Value of Improvement (REVI) sampling procedure determines the next (instance, alternative) pair to sample based on a Monte-Carlo estimate of the global improvement  $EI(x, a)$ . By replacing the continuous integral over all instances with a Monte-Carlo integral over a fixed set of  $N_X$  instances,  $X_{MC} = \{x_1, \dots, x_{N_X}\}$ , that are randomly distributed according to  $\mathbb{P}[x]$ . The integral becomes

$$EI(x, a) \approx \hat{EI}(x, a) = \frac{1}{N_X} \sum_{x_i \in X_{MC}} \mathbb{E} \left[ \max_a \mu_a^{n+1}(x_i) - \max_{a'} \mu_{a'}^n(x_i) \right] \quad (17)$$

$$= \frac{1}{N_X} \sum_{x_i \in X_{MC}} \Delta_a(x_i) \Phi(\Delta_a(x_i) / \tilde{\sigma}_a(x_i; x)) - \tilde{\sigma}_a(x_i; x) \phi(\Delta_a(x_i) / \tilde{\sigma}_a(x_i; x)). \quad (18)$$

The above integral  $\hat{EI}(x, a)$  gives an estimate of the total improvement  $EI(x, a)$ .  $LEVI(x, a)$  gives the single point improvement therefore  $LEVI(x, a)V_X$  is also an estimate of the overall improvement where  $V_X = \int_x dx$  is the volume of the instance space. Combining these two estimates of improvement weighted according their sample sizes yields

$$REVI(x, a) = \frac{1}{N_X + 1} (N_X \hat{EI}(x, a) + LEVI(x, a)V_X). \quad (19)$$

The REVI method allocates samples to (instance, alternative) pairs for which the difference between the chosen alternative and the best remaining alternative is close across correlated instances, where the prediction uncertainty is high across those instances and where the instance density is high across correlated instances. After each new sample, the set of fixed points,  $X_{MC}$ , are reproduced such that the mapping does not overfit to a single discretization.  $N_X$  may be seen a a parameter to be chosen by the user, and setting  $N_X = 0$  the REVI method becomes equivalent to LEVI method. In Section 5 we choose  $N_X = n$  such that the discretization density grows with the sample density.

A single call to  $REVI(x, a)$  requires one call to  $LEVI(x, a)$  and also the means of the  $X_{MC}$  instances for all alternatives and also all posterior covariances  $k_a^n(X_{MC}, x)$  to calculate  $\tilde{\sigma}_a^n(X_{MC}; x)$ . The means for all  $N_X$  instances and alternatives can be precomputed and stored between  $REVI(x, a)$  calls. Also the final terms of the matrix multiplication in Equation 3 for the posterior covariances may be precomputed reducing each  $k_a^n(x_i, x)$  call from  $O(n_a^2)$  to  $O(n_a)$ . Therefore after the first call, each additional call to  $REVI(x, a)$  has leading order complexity  $O(n_a^2 + N_X n_a)$ . Assuming  $N_X = n \approx An_a$  the leading complexity is  $O(n_a^2)$  which is equivalent to  $LEVI(x, a)$  and EGO. Again optimization of  $REVI(x, a)$  can be done by many methods and in Section 5 we use the same Nelder-Mead algorithm with multiple starts.

If the instance distribution is discrete,  $X = \{x_1, \dots\}$  and  $\mathbb{P}[x_i] = p_i$ , then  $\lim_{N_X \rightarrow \infty} \hat{I} = EI(x_i, a)$  may be computed exactly by summing instances weighted according to their probabilities and the  $LEVI(x, a)$  term may be dropped from  $REVI(x, a)$ . If all instances were equally likely  $p_i = 1/|X|$ , the simplified REVI method is equivalent to the REVI method given in (Pearce and Branke 2017) which is asymptotically and myopically optimal and also has a suboptimality bound.

In Section 5 we show that the REVI method significantly outperforms LEVI and another method from the literature in our tests.

### 4.3 Neighbors Only Regional Expected Value of Improvement

The Monte-Carlo integral  $\hat{EI}$  with  $N_X$  samples requires computing  $\tilde{\sigma}_a^n(x_i, x)$ , the change in the predicted performance of an instance  $x_i \in X_{MC}$  caused by the new sample at  $x$ , which requires computing  $k_a^n(x_i, x)$ . When covariance between  $x_i$  and  $x$  is low, the instance expected improvement is very small. As a result the summation in Equation 17 may be dominated by a few large terms and many small negligible terms. To improve efficiency, one may use importance sampling, for example with a stationary kernel, setting  $X_{MC}$  focused around  $x$  would significantly reduce error in  $\hat{EI}(x, a)$ . However in our experiments this lead

to slower computation, it's not possible to precompute means and partial covariances, and the  $REVI(x, a)$  is very rugged and not easily optimized.

Instead we propose the Neighbors Only Regional Expected Value of Improvement method (NREVI). The Monte-Carlo instances,  $X_{MC}$ , are generated as with REVI, however, only instances whose prior covariance is above a threshold are included in the Monte-Carlo integral. By filtering points according to the cheaply computed prior covariance, we can avoid the costly matrix multiplication in posterior covariance in Equation 3 for points that do not significantly contribute to  $\hat{EI}$ . The expected improvement is given by

$$EI(x, a) \approx \hat{EI}_N(x, a) = \frac{1}{N_X} \sum_{x_i \in X_{MC}} \mathbb{E}[\max_a \mu_a^{n+1}(x_i) - \max_{a'} \mu_{a'}^n(x_i)] \mathbb{1}\{k_a^0(x_i, x) > \delta\} \quad (20)$$

where  $\mathbb{1}\{\}$  is the indicator function returning unity if the argument is true. Otherwise the NREVI acquisition function is the same as the REVI function

$$NREVI(x, a) = \frac{1}{N_X + 1} (N_X \hat{EI}_N(x, a) + V_X LEVI(x, a)) \quad (21)$$

After each new sample the  $X_{MC}$  is randomly regenerated. This sampling method has the same advantages as the LEVI and REVI methods however it gives a similarly accurate estimate of improvement to REVI while significantly reducing computational cost. The N-REVI method has two parameters,  $N_X$  and  $\delta$ . When using a stationary kernel,  $k^0(x, x') = h(|x - x'|)$ , such as the Matern class of kernels or squared exponential, points may be filtered according to  $|x - x'| < r$  and the two free parameters are  $N_X$  and  $r$ . We show in Section 5 that computation time of N-REVI is significantly less than REVI without any significant loss in performance. By setting  $N_X = 0$  or  $\delta = r = 0$  the N-REVI method is equivalent to LEVI, also by setting  $\delta = \infty$  or  $r = \infty$  the N-REVI method is equivalent to REVI, therefore computational complexity is between REVI and LEVI depending on the choice of parameters, the kernel and  $\mathbb{P}[x]$ .

## 5 NUMERICAL EXPERIMENTS

We apply our methods to a set of four ( $A = 4$ ) synthetic functions sampled from a Gaussian process prior with a squared exponential kernel,  $\theta_1(x), \dots, \theta_4(x) \sim \mathcal{GP}(\mu^0(x) = 0, k^0(x, x') = \exp(-0.5\|x - x'\|_2^2/7^2))$ . The input domain is  $X = [0, 100]^2$  and the output domain is real numbers. In practice, for each function, a multivariate normal sample is produced from the prior evaluated at a discretized  $X$  and the sample is used to condition the continuous prior, yielding a continuous posterior mean. The posterior mean functions are used as synthetic functions  $\theta_1, \dots, \theta_4$  and plotted with the optimal mapping in Figures 1 (a) and (b). The noise is set to  $\sigma_{a,\epsilon}^2 = 0.1^2$  for all alternatives. Given these four functions we use two instance distributions: a) A uniform distribution (UNI)  $\mathbb{P}_U[x] = 1/10,000$ , so that we can compare directly with the Gap-SUR method proposed by Hu and Ludkovsk (2017), and b) A "Wedge" instance distribution,  $\mathbb{P}_W[x] = x_1/5 * 10^5$ , linearly increasing only in the  $x_1$  direction.

To initialize sampling, a random design is used as described below with 10 samples per alternative, 40 samples in total, Figure 1 e) show one such the sample allocation for the wedge distribution. In practice these samples would be used to learn hyperparameters for the Gaussian Processes however we treat them as constant throughout sampling for all methods such that the only difference between all methods is the acquisition function. Figures 1 c) and d) give the estimated performance functions and mapping after fitting a GP to the initial 40 samples.

After the initialization, the sequential methods are then applied until 200 samples in total have been allocated. Figure 1 f) shows the  $LEVI(x, a)$  after the initialization and it can be seen that LEVI prioritizes high  $\mathbb{P}[x]$ , which is on the boundary of the domain. However  $REVI(x, a)$  shown in Figure 1 g) does not prioritize the highest  $\mathbb{P}[x]$  but is closer to the mean  $x$  where the regional benefit of a new sample is greater as there are many instances surrounding the new sample.

To measure the quality of the mapping, for each budget size,  $N \in [40, \dots, 200]$ , the mapping  $S(x) = \operatorname{argmax}_a \mu_a^N(x)$  is evaluated for 1000 points in the space  $X$  and the total opportunity cost is recorded. The

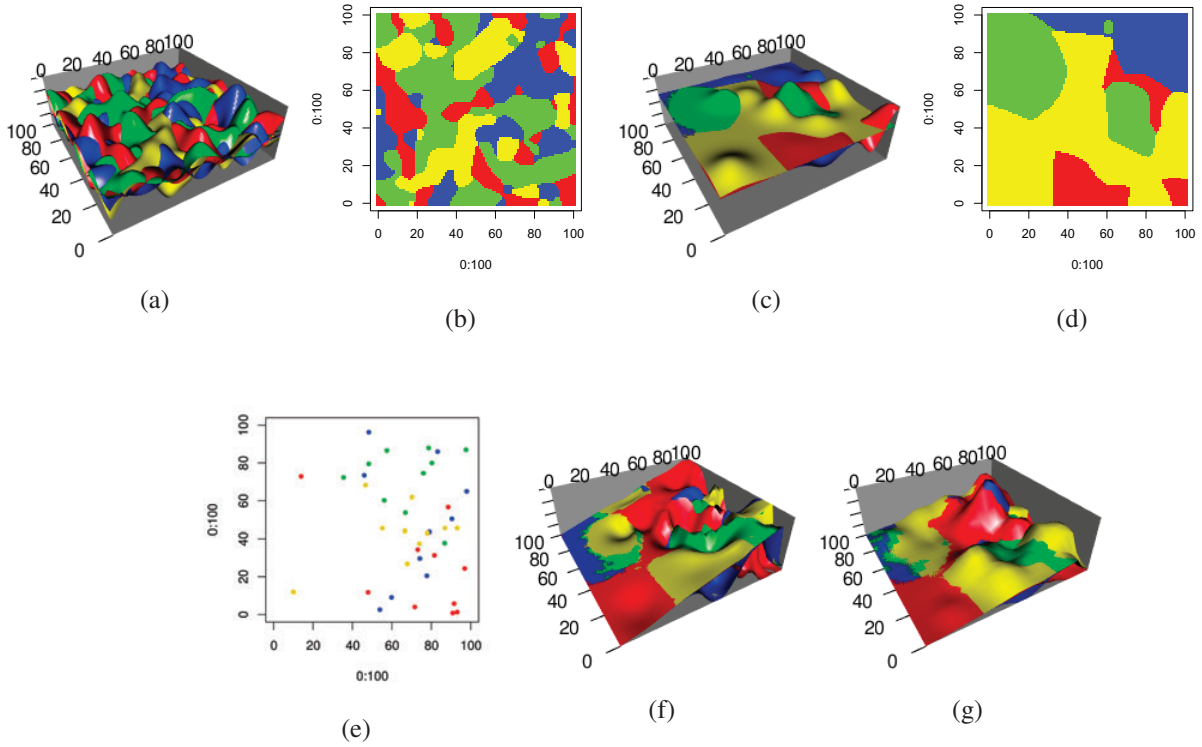


Figure 1: In all plots:  $x$  and  $y$  axis are instance features, each color represents an alternative. (a),(b) the true performance functions and the true optimal mapping. (c)(d) the predicted functions and mapping after the initial 40 samples when using the wedge distribution. (e) the allocation of the 40 samples. (f) the  $LEVI(x,a)$  and (g)  $REVI(x,a)$ , note how  $LEVI(x,a)$  is greatly influenced by the  $\mathbb{P}[x]$  and peaks on the edge of the domain, whereas  $REVI(x,a)$  is smoother and peaks further within the domain where more instances can benefit from a new measurement.

1000 test instances,  $X_T$ , are distributed according to  $\mathbb{P}_U[x]$  and  $\mathbb{P}_W[x]$  for uniform and wedge respectively. For each distribution they are generated once and the same instances are used to measure opportunity cost for all experiments. They are never used for the random sampling method nor in  $X_{MC}$ , it is a left-out test set.

$$\text{Opportunity Cost} = \sum_{x_i \in X_T} \max_a \theta_a(x_i) - \theta_{S(x_i)}(x_i). \quad (22)$$

Each sampling method is applied 400 times with different initial samples and noise values. For the REVI methods we run experiments for  $N_X \in \{0.5n, n, 2n, 4n\}$ , and  $r \in \{1l_X, 2l_X, 3l_X, \infty\}$  where  $l_X = 7$  is the kernel length scale. In order to investigate the influence of the varying sample size  $N_X$  and overfitting, we also run experiments with REVI where the Monte-Carlo sample size is constant:  $N_X = 100$  and  $N_X = 200$  and where the MC samples are fixed for all time or randomized at each time step which we refer to as "jittering". We compare the final opportunity cost of these methods with original REVI where the MC sample size grows as the budget is consumed.

To measure the computational cost of the methods, the time taken to optimize the acquisition function is measured, all parameters for the Nelder-Mead algorithm are kept constant across all experiments  $\alpha = 1$ ,  $\beta = 0.5$ ,  $\gamma = 2$  with 15 random restarts seeded by points sampled from the appropriate  $\mathbb{P}[x]$ .



## 5.1 Alternative Methods

### 5.1.1 Random Sampling

Given a budget of  $N$  samples,  $N$  instances are chosen by Latin Hypercube (uniform case) and by sampling  $\mathbb{P}_W[x]$  (wedge case). The points are randomly divided into  $A$  groups whose sizes differ by at most 1. For all (instance,alternative) pairs performance,  $f(x,a)$ , is measured and a Gaussian Process is fitted to the resulting dataset. This method represents the naive approach that does not exploit past evaluations to inform new ones.

### 5.1.2 Gap-SUR

Hu and Ludkovsk (2017) consider the same problem with a uniform instance distribution. The proposed method allocates samples to the (instance, alternative) pair that maximizes the reduction in the upper bound of the highest performing alternative:

$$\text{Gap} - \text{SUR}(x, a) = \left( \mathbb{E}[\max_{a'} \theta_{a'}(x) | \mathcal{F}^n] - \mathbb{E}[\max_{a'} \theta_{a'}(x) | \mathcal{F}^{n+1}] \right) \mathbb{P}[x]$$

where  $\theta_a \sim N(\mu^n(a), k_a^n(x, x))$  and  $\mathcal{F}^{n+1}$  is the next filtration constructed by assuming the new sample will be equal to the current mean  $(x^{n+1}, a^{n+1}, y^{n+1}) = (x, a, \mu_a^n(x))$ . The maximum of multiple Gaussian random variables is calculated by approximating the maximum of any two Gaussian random variables as another Gaussian, for which first and second moments are known, and iterating over  $A$ . One evaluation of  $\text{Gap} - \text{SUR}(x, a)$  requires one call each to  $k_1^n(x, x), \dots, k_A^n(x, x)$ , however once the posterior variances are known, evaluating  $\text{Gap} - \text{SUR}(x, a)$  for all  $a$  with the same  $x$  is relatively cheap. Therefore we optimize the acquisition function over  $x$  once using Nelder-Mead with the same parameters as other methods and for each  $x$ ,  $\text{Gap} - \text{SUR}(x, a)$  is optimized over  $a$  by inspection, thereby reducing the number of posterior covariance calls.

## 5.2 Results

First we have examined the influence of the REVI parameters  $N_X$  (Monte Carlo sample size) and  $r$  (filtering radius) on the opportunity cost. Figure 2 shows the results for (a) the uniform distribution and (b) the wedge distribution. Clearly, increasing the  $N_X$  parameter significantly improves performance in all cases, and the improvement is larger for the wedge distribution than for the uniform distribution. Apparently larger  $N_X$  improves accuracy of the estimate of improvement more in a more complicated instance distribution. Varying  $r$  from  $r = 1$  to  $r = \infty$  does not provide significant performance benefit but a using a smaller  $r$  allows for a very large reduction in computation time. Computation times are presented in Figure 3. As can be seen, LEVI is fastest, followed by  $\text{Gap} - \text{SUR}$ . REVI still scales linearly with the number of samples, but the slope is steeper than for the other methods. Standard REVI with  $r = \infty, N_X = n$  has about the same runtime curve as NREVI with  $r = 1, N_X = 4n$ . But as we have seen earlier in Figure 2, the latter version performs much better in terms of opportunity cost. Thus, for our final comparison with the other benchmark methods, we limit ourselves to NREVI with  $r = 1, N_X = 4n$ .

Figure 4 displays how the opportunity cost decreases with the number of samples, for the different sampling methods. As expected, random sampling is significantly worse than all sequential methods for both instance distributions. LEVI performs as good as or better than  $\text{Gap} - \text{SUR}$ , and as seen in Figure 3 is slightly faster to optimize in this implementation. In all benchmarks, NREVI is significantly better than the local methods, LEVI and  $\text{Gap} - \text{SUR}$ , suggesting that even the simplest accounting of correlated instances,  $N_X = 4n, r = l_X$ , does yield a significant performance benefit for increased computational cost. For most real world simulation applications the running time is dominated by the simulation, here we see that *REVI* improves performance and yet scales poorly to large sampling budgets  $N$ . Whereas *NREVI*, in our experiments, has the performance benefits of *REVI* and scales to larger sampling budgets without the risk of consuming running time.

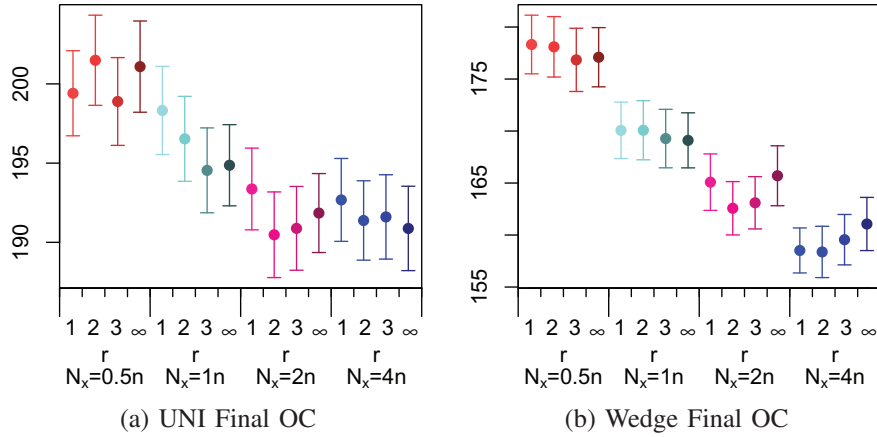


Figure 2: Opportunity cost of REVI depending on  $N_X$  and  $r$  at the end of the 200 samples taken.

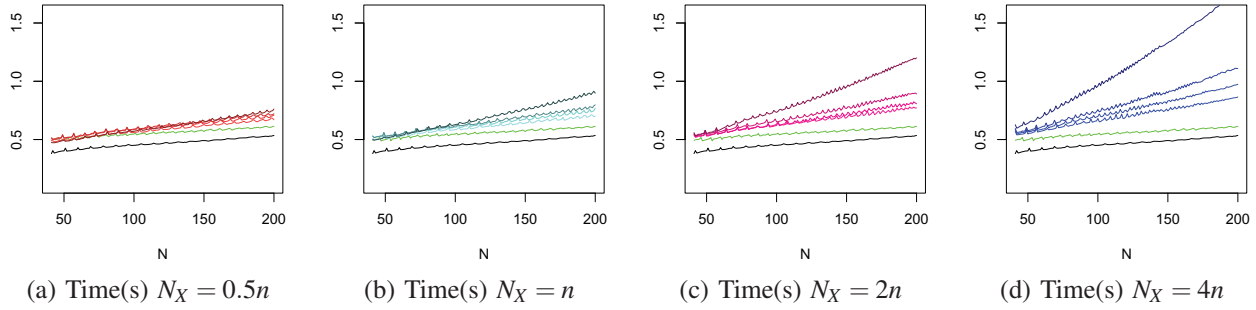


Figure 3: Optimization time in seconds per iteration for different sample sizes. Green is *Gap – SUR*, black is LEVI. REVI with  $r$  values increasing from light to dark.

The difference between the REVI and local methods is larger for the wedge distribution. The local methods prioritize the mode of  $\mathbb{P}[x]$ , which is at  $x_1 = 100$  in the wedge case. The REVI methods prioritize where there are many correlating instances, which will not be at  $x_1 = 100$  due to lack of  $X_{MC}$  points to contribute to  $\hat{EI}$  for a sample near a boundary. Therefore REVI methods outperform LEVI even in the uniform case as LEVI is not disincentivized from sampling near a boundary that yield marginal global improvement.

Finally, results comparing standard REVI ( $N_X = n/2$ ) with REVI where  $X_{MC}$  is fixed and where  $X_{MC}$  is jittered (both  $N_X = 100$ ) are given in Figure 5 a). Standard REVI performs worst presumably due to the small  $N_X$  for small budgets. This is followed by the fixed samples and then jittered samples suggesting the mapping  $S(x)$  was overfitting to the fixed  $X_{MC}$  and therefore jittering is necessary. When the  $X_{MC}$  set is equal to the sampling budget of 200, Figure 5, there is no significant difference between the three versions, presumably there are not enough samples allocated to overfit, this may not be the case for larger budgets. Although not shown the standard REVI requires less computation while the sample size is still small.

## 6 CONCLUSION

We have considered the problem of finding the best or near best alternative from a set of alternatives for each point across a problem instance domain described by continuous features. We have proposed new myopic sampling methods that allow to efficiently derive an accurate mapping from problem instance features to the best alternative. In two synthetic benchmark problems, our methods show significantly better

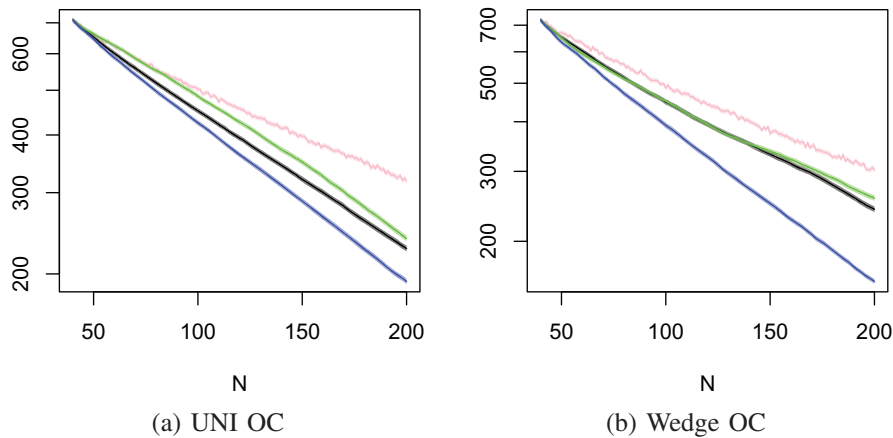


Figure 4: Opportunity cost over the course of the sampling, pink is random sampling, green is  $Gap - SUR$ , black is LEVI, and blue for REVI with  $r = 1, N_X = 4n$ .

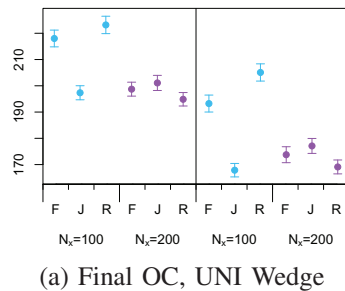


Figure 5: Final opportunity cost, uniform problems left, wedge problems right, for fixed points (F), jittering points (J) and standard REVI (R). Blue for  $N_X = 100$  and purple for  $N_X = 200$  ( $N_X = 0.5n$  and  $n$  for REVI).

performance than random sampling as well as a state-of-the-art algorithm from the literature. Furthermore, we have compared various ways of numerically integrating the expected improvement over the problem instance domain, and found that using a large sample size, but only taking into account samples with correlation yields the best results at lower computational cost. Future work includes applying the method to a real world problem, and applying the same adaptation to the case where there is correlation between the performance of different alternatives for a given problem instance.

## REFERENCES

- Branke, J., S. E. Chick, and C. Schmidt. 2007. “Selecting a Selection Procedure”. *Management Science* 53 (12): 1916–1932.
- 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.
- Chick, S. E., J. Branke, and C. Schmidt. 2010. “Sequential Sampling to Myopically Maximize the Expected Value of Information”. *INFORMS Journal on Computing* 22 (1): 71–80.
- Frazier, P., W. Powell, and S. Dayanik. 2009. “The Knowledge-Gradient Policy for Correlated Normal Beliefs”. *INFORMS journal on Computing* 21 (4): 599–613.
- Frazier, P. I., W. B. Powell, and S. Dayanik. 2008. “A Knowledge-Gradient Policy for Sequential Information Collection”. *SIAM Journal on Control and Optimization* 47 (5): 2410–2439.

- 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.
- Heger, J., J. Branke, T. Hildebrandt, and B. Scholz-Reiter. 2016. "Dynamic Adjustment of Dispatching Rule Parameters in Flow Shops with Sequence-Dependent Set-up Times". *International Journal of Production Research* 54 (22): 6812–6824.
- Hu, R., and M. Ludkovsk. 2017. "Sequential Design for Ranking Response Surfaces". *SIAM/ASA Journal on Uncertainty Quantification* 5 (1): 212–239.
- 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.
- Krause, A., and C. S. Ong. 2011. "Contextual Gaussian Process Bandit Optimization". In *Advances in Neural Information Processing Systems*, 2447–2455.
- Morales-Enciso, S., and J. Branke. 2015. "Tracking Global Optima in Dynamic Environments with Efficient Global Optimization". *European Journal of Operational Research* 242 (3): 744–755.
- Pearce, M., and J. Branke. 2017. "Efficient Information Collection on Portfolios". *Warwick Research Archive Portal*.
- Poloczek, M., J. Wang, and P. I. Frazier. 2016. "Warm Starting Bayesian Optimization". 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, 770–781. 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.
- Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. 2016. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE* 104 (1): 148–175.
- Zhou, L. 2015. "A survey on contextual multi-armed bandits". *arXiv preprint arXiv:1508.03326*.

## 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](mailto: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](mailto:Juergen.Branke@wbs.ac.uk).