On Bayesian Search for the Feasible Space Under Computationally Expensive Constraints

Alma Rahat\textsuperscript{1} and Michael Wood\textsuperscript{2}

\textsuperscript{1} Department of Computer Science, Swansea University, Swansea, UK. 
\textsuperscript{2} ACT Acoustics, Exeter, UK.

\texttt{a.a.m.rahat@swansea.ac.uk}
\texttt{m.wood@actacoustic.co.uk}

\textbf{Abstract.} We are often interested in identifying the feasible subset of a decision space under multiple constraints. However, in cases where the constraints cannot be represented by analytical formulae, the cost of solving these problems can be prohibitive, since the only way to determine feasibility is to run computationally or financially expensive simulations. We propose a novel approach for this problem: we learn a surrogate classifier that can rapidly and accurately identify feasible solutions using only a very limited number of samples ($11n$, where $n$ is the dimension of the decision space) obviating the need for full simulations. This is a data-efficient active-learning approach using Gaussian processes (\textit{GPs}), a form of Bayesian regression models, and we refer to this method as Bayesian search. Using a small training set to begin with, we train a \textit{GP} model for each constraint. The algorithm then identifies the next decision vector to expensive evaluate using an acquisition function (a surrogate-assisted indicator of quality). We subsequently augment the training data set with each newly evaluated solution, improving the accuracy of the estimated feasibility on each step. This iterative process continues until the limit on the number of expensive evaluations is reached. Initially, we adapted acquisition functions from the reliability engineering literature for this purpose. However, these acquisition functions do not appropriately consider the uncertainty in predictions offered by the \textit{GP} models. We, therefore, introduce a new acquisition function to account for this. The new acquisition function combines the probability that a solution lies at the boundary between feasible and infeasible spaces representing exploitation) as well as the entropy in predictions (representing exploration). To test the efficacy of our approach, we selected five problems from the popular G test suite for constrained optimisation. The results show that the best classifier has a median informedness of at least $97.95\%$ across all problems, confirming the overall effectiveness of the approach.

1 Introduction

In engineering applications, we are often interested in determining in the feasible design space for a given problem. This requires estimating a set of decision

\textsuperscript{3} The feasible space is sometimes referred to as the level set \cite{levelset}.
variables that does not violate given constraint functions. This is a challenging task, particularly if the constraints cannot be expressed analytically. In these cases, computationally expensive simulations or physical experiments are required to explore the design space. For instance, in some nuclear power applications, keeping the neutron production ratio below the critical level is essential for safe operation [2]. This presents a significant design challenge. It is not practical to test each set of plant parameters by simulation, since each evaluation of the simulator takes between 5 and 30 minutes. An exhaustive search using Monte Carlo simulations is therefore not practical [3].

We propose a data-driven approach to such problems as surrogate-assisted methods are known to perform well for problems with a strict budget on the number of expensive evaluations [2,4].

Apart from design exploration applications, the ability to determine the feasible space may also be useful in constrained global optimisation of problems for finding an initial feasible solution [5] as well as integrating the probability of feasibility within the search [6,7].

In this paper, we focus on a surrogate-assisted sequential Bayesian search method. This method was inspired by the Efficient Global Optimisation (EGO) method [8] (often referred to as Bayesian Optimisation).

This method starts with a small training set of independent parameters. These parameters are expensively evaluated with a set of constraints functions. We then use the resulting dataset to train a Bayesian regression model (in this case, a Gaussian process, \( \mathcal{GP} \)) for each constraint [9]. Together, these models estimate the probability that a given solution is feasible. In this way, the combination of models act as a binary classifier.

The challenge in creating this model is to select the next sample such that it gives us the greatest improvement in the feasible space estimation. This sample is estimated by maximising an acquisition function (often referred to as an infill criterion or a utility function). We keep adding additional samples until the budget on additional expensive evaluations is exhausted.

We understand that using Bayesian search in this way is new. Although we are aware of one other paper on this topic by Knudde et al., their approach only works when there is a single constraint for determining feasibility [10]. Our approach address a scenario where there are multiple constraints. The novel contributions of our work are:

– A full investigation of a range of acquisition functions in Bayesian search method for rapidly estimating the feasible space imposed by multiple computationally expensive constraints.

– A new acquisition function \( \alpha_{PBE}(\cdot) \) based on the probability of a solution residing at the boundary (representing exploitation) and the entropy of predictive distribution (representing exploration). We use this function to con-
struct a classifier for predicting the feasibility of a solution. This function exhibits a high informedness using only a small number ($11n$, where $n$ is the dimension of the decision space) of function evaluations.

We organise this paper as follows. In section 2 we review related work from the reliability engineering literature. In section 3 we discuss necessary concepts focusing on using GP's to model constraints functions, and the standard Bayesian search framework. Then we propose a range of acquisition functions suitable for Bayesian search of the feasible space in section 4. We present our results in section 5. Finally, we finish with general conclusions in section 6.

2 Related Work

The reliability engineering literature has much work devoted to system reliability analysis (SRA). SRA is applied when there are multiple failure modes in a system [11], and Yang et al. [12] provide a comprehensive review of work in this area. In these cases, a sequential search approach is adopted to constructing constraint models, which are then used to compute the probability of failure. Here, their ultimate goal is to estimate the total volume of the infeasible space or the excursion set [13].

The earliest approaches to modelling the boundary of the feasible space used either polynomials (typically first or second-order) [14–16] or support vector machines (SVM) [17]. However, these approaches are limited. Under multiple constraints, the boundary is often highly-non-linear, and may be even be discontinuous [18].

Polynomials and SVNs therefore perform poorly in modelling the boundary directly. To solve this problem, others have attempted to model the constraint functions instead. Attempts have been made using neural networks [19] and SVMs [20], but since these methods only produce point-predictions, there is no quantification of uncertainty. The predictions of the feasible space may therefore be misleading [21].

Recently, GP models have shown promise as a framework for active learning [11,12,22,25]. The GP approach is similar to Bayesian search. The difference is that a the GP approach maximises the acquisition function using a variant of Monte Carlo search to find the next promising sample to simulate. This is an important distinction, since the Monte Carlo search does not perform as well as evolutionary search methods. The Bi-population Covariance Matrix Adaptation Evolutionary Strategy (Bi-POP-CMA-ES) has been shown to perform better than Monte Carlo [26], so we propose this method for maximising the infill function.

Many of the popular approaches adopt a composite criterion approach [12]. In these approaches, an acquisition function is created with the aim of improve the estimation of each relevant constraint. Each model is selected based on the mean
predictions. These predictions determine which acquisition function should be used to select the parameters for the next expensive simulation. This approach is effective, but there are some drawbacks which mean that they are not suitable for our approach. Firstly, a model is selected by considering all Monte Carlo samples. The adaptation in our framework would therefore require a reformulation of the combined acquisition function (which we perform in (14)). Secondly, irrespective of the reformulation, the selection of the model requires reliance on the mean predictions. This may be misleading, particularly during the early stages of the search where data is sparse. Finally, the composite criterion approach tends to underperform if the constraint functions have a difference in scales and cannot be easily normalised [12]. Our proposed acquisition function does not require the model selection step. Instead, it combines predictive distributions from all models. This allows the computation of the utility of a candidate solution using the models without the need to normalise the value of individual constraints.

3 Background

Consider, a design vector \( \mathbf{x} \) in a design space \( \mathcal{X} \subseteq \mathbb{R}^n \). Without loss of generality, a constrained problem with \( L \) constraints can be defined as:

\[
G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_L(\mathbf{x}))^\top \leq \mathbf{t} = (t_1, \ldots, t_L)^\top,
\]

where, \( g_l : \mathbb{R}^n \to \mathbb{R} \) is the \( l \)th constraint function with a threshold for feasibility \( t_l \). To deal with equality constraints, we can add a small fixed constant \( \epsilon \). This converts the equation to an inequality constraint [27].

The \( l \)th constraint function \( g_l(\mathbf{x}) \) generates a feasible space \( \mathcal{F}_l \subseteq \mathcal{X} \). The infeasible set of solutions for this constraint is therefore \( \mathcal{I}_l = \mathcal{X} \setminus \mathcal{F}_l \). The total infeasible set of solutions becomes \( \mathcal{I} = \bigcup_{l=1}^L \mathcal{I}_l \). If all constraints are considered, the feasible space is at the intersection of all feasible sets: \( \mathcal{F} = \bigcap_{l=1}^L \mathcal{F}_l \).

From a reliability engineering perspective, such a combination of constraints is considered as a parallel combination of multiple failure modes [12]. We confirmed this to be true by a small test, and we exclude the approach from our investigation.

We can use a scalarisation approach as an alternative method for dealing with the multiplicity of constraints. This approach could encapsulate all constraints into a single function so that any violation of the scalarised constraint is equivalent to infeasibility [5]:

\[
s(\mathbf{x}) = \max_{l=1}^L (g_l(\mathbf{x}) - t_l) \leq 0.
\]

Here, the response of \( s : \mathbb{R}^n \to \mathbb{R} \) is only greater than 0 for a design vector resulting in an infeasible solution, iff at least one of the component \( l \)th constraints is violated \((g_l(\mathbf{x}) > t_l)\). From a reliability engineering perspective, mono-surrogate approaches like this are known to be inferior [12]. We confirmed this to be true by a small test, and we exclude the approach from our investigation.
If constraint functions are cheap to evaluate, we can determine feasibility by brute force using Monte Carlo methods \[28\]. However, where each constraint function \( g_l(x) \) requires an independently evaluated and computationally expensive simulation, this approach would be prohibitively slow.

### 3.1 Modelling Constraints with Gaussian Processes

Gaussian processes (GP) are commonly used to construct surrogate models for constraints \( g_l(x) \). GPs produce a Normal predictive distribution for any arbitrary solution, meaning that both a mean and confidence distribution are produced\(^4\). The information provided by this predictive distribution can be used by the acquisition function to locate promising solutions.

In essence, a GP is a field of joint Gaussian distributions \[9\]. If the GP is based on sample data comprising \( g_l(x) \) evaluated at \( M \) locations \( D_l = \{(x_m, g_l(x_m))\}_{m=1}^{M} \) of \( M \) samples, the predictive probability for \( g_l \) at \( x \) is a Gaussian distribution with mean \( \mu_l(x) \) and variance \( \sigma^2_l(x) \):

\[
p(g_l \mid x, D_l, \theta_l) = \mathcal{N}(\mu_l(x), \sigma^2_l(x) \mid x, D_l, \theta_l),
\]

where the mean and variance are

\[
\mu_l(x) = \kappa(x, X)K^{-1}g_l
\]

\[
\sigma^2_l(x) = \kappa(x, x) - \kappa(x, X)^\top K^{-1}\kappa(X, x).
\]

Here \( X \in \mathbb{R}^{M \times n} \) is the matrix of design locations and \( g_l \in \mathbb{R}^M \) is the corresponding vector of the true function evaluations using \( g_l(\cdot) \); thus \( D_l = \{(X, g_l)\} \). The covariance matrix \( K \in \mathbb{R}^{M \times M} \) represents the covariance function \( \kappa(x, x'; \theta_l) \) evaluated for each pair of observations and \( \kappa(x, X) \in \mathbb{R}^M \) is the vector of covariances between \( x \) and each of the observations; \( \theta_l \) denotes the kernel hyperparameters.

We use the Matern 5/2 kernel as the covariance function as this approach is recommended for modelling realistic functions \[29\]. To train a GP model we estimating the hyperparameters \( \theta_l \) by maximising the log likelihood of the data:\(5 \)

\[
\log p(D_l \mid \theta_l) = -\frac{1}{2} \log |K| - \frac{1}{2} S^\top K^{-1} S - \frac{M}{2} \log(2\pi).
\]

In the following equations, we omit \( \theta_l \) for simplicity as the elements in \( \theta_l \) are set by maximum log likelihood estimates.

\(^4\) A comprehensive introduction may be found in \[9\].

\(^5\) We use the limited memory BFGS algorithm with 10 restarts to estimate the hyperparameters \[30\].
We train a model for each constraint independently. For each $l$th constraint function and an arbitrary design vector $x$, we derive a posterior predictive distribution $p(g_l \mid x, \mathcal{D}_l) = \mathcal{N}(\mu_l(x), \sigma_l^2(x))$. Thus, the combined posterior predictive distribution across all component models is a multi-variate Gaussian:

$$p(G \mid x, \mathcal{D}) = \mathcal{N}(\mu(x), \Sigma(x)) = \prod_{l=1}^L p(g_l \mid x, \mathcal{D}_l),$$

where, the training data set is $\mathcal{D} = \{(x_m, g_1(x_m)), \ldots, g_L(x_m))\}_{m=1}^M$, the mean prediction vector is $\mu(x) = (\mu_1(x), \ldots, \mu_L(x))^\top$, and the predictive covariance matrix is $\Sigma(x) = \text{diag}(\sigma_1^2(x), \ldots, \sigma_L^2(x))$ with no cross-covariances due to the independence between models. With this combined predictive distributions, we can compute the probability of feasibility.

### 3.2 Classifying the Feasible Space

Given that the predictive distributions are Gaussian, we can compute the probability of any violation of the individual constraints. For the $l$th constraint, the probability of feasibility is $\Phi\left(\frac{t_l - \mu_l(x)}{\sigma_l(x)}\right)$.

$$p(x \in F_l) = p\left( p(g_l \mid x, \mathcal{D}_l) \leq t_l \right) = \Phi\left(\frac{t_l - \mu_l(x)}{\sigma_l(x)}\right).$$

The overall probability of feasibility is therefore:

$$p(x \in F) = \prod_{l=1}^L p(x \in F_l) = \prod_{l=1}^L \Phi\left(\frac{t_l - \mu_l(x)}{\sigma_l(x)}\right).$$

Due to symmetry, the probability of infeasibility is $p(x \in I) = 1 - p(x \in F)$. Using these probabilistic estimations, a decision vector $x$ is feasible iff $p(x \in F) > p(x \in I)$. Figure 1 illustrates the predicted feasible spaces for two constraints modelled with two GPs.

### 3.3 Bayesian Search Framework

Bayesian search is a surrogate-assisted active learning framework. This method takes inspiration from Efficient Global Optimisation (EGO), first proposed by Kushner [34] and later improved by Jones et al. [8]. The framework can be used to minimise the mean squared error in the sequential design of experiments, and is particularly useful where there are few observations [35]. It has also been used to compute the volume of infeasible space [2, 33, 36] and to locate the feasible space for single constraint problems [10].
Bayesian search is a global search strategy that sequentially samples the design space to efficiently determine the boundary between the feasible and infeasible space. The algorithm has two stages: initial sampling, and sequential improvement.

The initial sampling phase samples parameters using a space filling design, typically with Latin Hypercube design [37]. The parameters are then evaluated by the true function. The initial design set is formed of the parameter samples and their true-function output. Each design set is used to create a set of models, one for each constraint, $\hat{G} = \{\hat{g}_1, \ldots, \hat{g}_L\}$.

For the sequential improvement phase, we can use $\hat{G}$ to locate promising samples. $\hat{G}$ provides a multi-dimensional posterior distribution $p(G \mid x, D)$. For any design vector, $G$ provides a mean prediction (a vector) and uncertainty (a covariance matrix).

The predictive distribution permits a closed form calculation of probabilistic queries. We use it to predict whether or not a constraint function value will be exceed a threshold. Since our goal is to minimise the uncertainty around the threshold that bounds the infeasible space, we can design our acquisition function $\alpha(x, \hat{G}, t)$ accordingly. The aim is to strike a balance between exploitation (through mean predictions) and global exploration (through prediction uncer-
In this way, the acquisition function will drive the search towards the areas we are interested in. We present candidate acquisition functions in section 4.

The most promising solution is where $x^* = \arg\max_x \alpha(x, \hat{G}, t)$. We then determine $x^*$ expensively and use the results to augment the data and retrain $\hat{G}$. We repeat this process until we exhaust the simulation budget. When training is complete, we use $\hat{G}$ to estimate the feasible space. For an arbitrary $x$ a probability of feasibility is returned using (9). Algorithm 1 summarises the method.

Algorithm 1 Bayesian search framework.

Inputs
- $M$: Number of initial samples
- $T$: Budget on expensive function evaluations
- $t$: Threshold vector

Steps
1: $X \leftarrow \text{LatinHypercubeSampling}(X)$ \hfill $\triangleright$ Generate initial samples
2: $\Gamma \leftarrow \{G(x \in X)\}$ \hfill $\triangleright$ Expensively evaluate all initial samples
3: for $i = M \to T$ do
4: $\hat{G} \leftarrow \text{TrainGP}(X, \Gamma)$ \hfill $\triangleright$ Train a mono- or multi-surrogate model of constraints
5: $x^* \leftarrow \arg\max_x \alpha(x, \hat{G}, t)$ \hfill $\triangleright$ Optimise acquisition function
6: $X \leftarrow X \cup \{x^*\}$ \hfill $\triangleright$ Augment data set with $x^*$
7: $\Gamma \leftarrow \Gamma \cup \{G(x^*)\}$ \hfill $\triangleright$ Expensively evaluate $x^*$
8: end for
9: return $\hat{G}$ \hfill $\triangleright$ Return trained models for feasibility classification using (9)

4 Acquisition Functions

Some of the most popular acquisition functions can be adapted for Bayesian search. In this section, we describe how popular acquisition functions can be adapted. We also propose a new acquisition function.

In reliability engineering, acquisition functions were developed for a single constraint. These were first used in an active learning framework by Ranjan et al. [38] and Bichon et al. [39], and later popularised by Picheny et al. [2, 40] for computing the volume of infeasible space.

For Bayesian search of the feasible space, the acquisition function’s aim is to locate the boundary between feasible and infeasible spaces: $\mathcal{F}_I$ and $\mathcal{I}_I$. A solution based on $\hat{g}_I$ is often identified with $p(x \in \mathcal{F}_I) = 0.5$. If we add the sample at this to the training set, the estimation of feasibility with $\hat{g}_I$ in maximally improved. In this way, we achieve maximal exploitation of the latest knowledge of the model.
When data is limited, the uncertainty in predictions may be high, especially in areas with few samples. We should, therefore, promote exploration by sampling areas with high uncertainty.

However, if we only prioritise uncertain areas for sampling, we may miss areas near the threshold of interest. We therefore need to consider areas where both the uncertainty and the probability of being at the boundary are high. This strikes a balance between myopic exploitation and global exploration, and aims to gain as much knowledge as possible about the boundary with every new addition to the training dataset.

The most popular acquisition functions for single constraint are:

\[
\alpha_T^l(x, \hat{g}_l, t_l) = \sigma(x)\phi(z), \tag{10}
\]

\[
\alpha_B^l(x, \hat{g}_l, t_l) = \sigma(x)[z^+\Phi(z^+)+z^-\Phi(z^-)+\phi(z^+)+\phi(z^-)-2z\Phi(z)-2\phi(z)], \tag{11}
\]

\[
\alpha_R^l(x, \hat{g}_l, t_l) = \sigma^2(x)[z^2(\Phi(z^-)-\Phi(z^+))+z^+\phi(z^-)-z^-\phi(z^+)]. \tag{12}
\]

Here, \(z = \frac{\mu_l(x)-t_l}{\sigma_l(x)}\), \(z^+ = z + 1\), \(z^- = z - 1\), and \(\Phi(\cdot)\) and \(\phi(\cdot)\) are standard Gaussian cumulative and probability density functions respectively. \(\alpha_T^l(\cdot)\) is the targeted mean squared error and was defined by Picheny et al. \[33\]. \(\alpha_B^l(\cdot)\) and \(\alpha_R^l(\cdot)\) are functions that compute a form of average positive difference between uncertainty and the predictive distance from the threshold, defined by Bichon et al. \[39\] and Ranjan et al. \[38\]. Further details of these can be found in \[33,36\].

A similar acquisition function proposed by Echard et al. can also be used. This is written as \[12,22\]:

\[
\alpha_E^l(x, \hat{g}_l, t_l) = -\left|\frac{\mu_l(x)-t_l}{\sigma_l(x)}\right|. \tag{13}
\]

This is the negative of the probability of wrongly predicting feasibility. Maximising this function is likely to find solutions that reduce the misclassification error.

We refer the reader to Lv et al. \[41\] and Sun et al. \[42\] for further work on single constraints.

To determining areas of system failure under multiple constraints, a composite-criterion approach is commonly taken. This approach calculates the acquisition function for each model, selecting a single model based on the best individual mean prediction \[12,22,23\]. A generalised version without using Monte Carlo samples is:

\[
\alpha_Y(x, \hat{G}, t) = \alpha_{Y[k]}(x, \hat{g}_k, t_k) \mid k = \arg\max_{l=1}^L (\mu_l(x) - t_l), \tag{14}
\]
where, $\alpha_{Y|k}(x, g_k, t_k)$ is the acquisition function for $k$th constraint $g_k(x)$, with $Y \in \{T, B, R, E\}$.

Using the acquisition function in (14) only improves individual boundaries between feasible and infeasible spaces for each constraint $g_k(x)$. However, this approach does not directly account for the true boundary under multiple constraints. For multiple constraints, any violation is treated as infeasible, and since equation (14) may sample infeasible space, it will likely introduce unnecessary redundancy. A further weakness is that the model selection $k = \arg\max_{l=1}^L (\mu_l(x) - t_l)$ does not consider prediction uncertainty. The result can therefore be misleading. The scale of the function value in each constraint can also cause problems, since the magnitude differences in $\mu_l(x) - t_l$ may be inverse to relative importance. Our new acquisition function aims to solve these shortcomings.

4.1 Probability of Being at the Boundary and Entropy (PBE)

So far we have discussed how single-constraint acquisition functions can be combined to create an acquisition function for multiple constraints. However, since our aim is to find solutions with a high probability of being at the boundary of the feasible space (exploitation), whilst minimising the overall uncertainty in the models (exploration), we combine these two objectives as a product.

The probability that a solution is at the boundary $\beta$ between the feasible and infeasible spaces, given a multi-surrogate model $\hat{G}$, is:

$$p(x \in \beta) = p(x \in F) \cdot p(x \in I) = \prod_{l=1}^L \Phi\left(\frac{t_l - \mu_l(x)}{\sigma_l(x)}\right) - \prod_{l=1}^L \Phi^2\left(\frac{t_l - \mu_l(x)}{\sigma_l(x)}\right).$$

(15)

If we maximise the probability over the design space, we will locate solutions at the boundary, thereby exploiting the current knowledge.

To evaluate the overall uncertainty for a multi-surrogate model $\hat{G}$, we compute the differential entropy of a multi-variate Gaussian distribution:

$$H(x | \hat{G}) = \frac{L}{2} \ln(2\pi e) + \frac{1}{2} \ln(|\Sigma|) \propto \prod_{l=1}^L \sigma_l^2(x).$$

(16)

The extremes of the above equation identify the solutions with most overall uncertainty across the models. These extremes identify the most informative samples.

To maximise both quantities, we combine these two measures together as a product. This creates our multi-surrogate acquisition function; the Probability
of Boundary and Entropy (PBE):

\[
\alpha_{PBE}(x, \hat{G}, t) = p(x \in \beta) \ H(x \mid \hat{G}).
\] (17)

This function addresses the true boundary \( \beta \) directly. It is particularly useful, since no explicit model selection is required. Further, since the probability and entropy are being computed via an intra-constraint model (rather than between constraints), we expect it to perform better for unscaled function responses.

5 Experiments

To test the performance of our approach, we used the test suite for constrained single-objective optimisation problems from CEC2006 [27]. We restricted the suite of test problems to those with a feasible space volume greater than 0.5% (Table 1).

We ran each method 21 times on each problem, starting from \( n \) initial training samples and a total budget of \( 11n \) evaluations. The initial evaluations are matched between acquisition functions, i.e. for each pair of problem and simulation run, the same initial design was used. The exception to this is the LHS with \( 11n \) samples.

Since the acquisition function landscape is (typically) multi-modal, we used Bi-POP-CMA-ES to search the space, as it is known to solve multi-modal problems effectively [26]. We set the maximum number of evaluations of the acquisition function to 5000\( n \).

We use informedness as a performance indicator for the classifier. The informedness estimates the probability that a prediction is informed, compared to a chance guess. We chose informedness as it performs well for imbalanced class sizes, which are common when comparing the sizes of feasible and infeasible spaces for real-world constrained problems [44,45].

To test statistical significance, due to matched samples, we used the one-sided Wilcoxon Signed Rank test with Bonferroni correction, identifying the best method at the level of \( p \leq 0.05 \) [46]. We used Mann-Whitney-U test to compare the LHS and other methods (Table 2).

Table 1: A range of test problems with a feasible space volume \( \rho \geq 0.5\% \) from the test suite defined in [27] and implemented in PyGMO [43]. Here, \( n \) is the dimension of the decision space, and \( L \) is the number of constraints.

| ID | \( n \) | \( \rho(\%) \) | \( L \) |
|----|--------|-------------|------|
| G4 | 5      | 26.9953     | 6    |
| G8 | 2      | 0.8727      | 2    |
| G9 | 7      | 0.5218      | 4    |
| G19| 15     | 33.4856     | 5    |
| G24| 2      | 44.2294     | 2    |
Table 2: Performance of different acquisition functions in terms of median informedness (%) and the median absolute deviation from the median (MAD). The red cells show the best median performance, while the blue cells depict the equivalent methods to the best.

|       | LHS | α_T | α_R | α_E | α_PBE |
|-------|-----|-----|-----|-----|-------|
| **G4** | Median | 99.83% | 99.95% | 99.94% | 99.93% | 99.95% | 99.99% |
|       | MAD   | 7.9 × 10^{-4} | 2.7 × 10^{-4} | 2.5 × 10^{-4} | 4.3 × 10^{-4} | 2.1 × 10^{-4} | 3.6 × 10^{-4} |
| **G8** | Median | 97.85% | 99.99% | 98.85% | 98.85% | 98.86% | 100% |
|       | MAD   | 1.4 × 10^{-2} | 6.0 × 10^{-3} | 1.0 × 10^{-2} | 1.0 × 10^{-2} | 9.0 × 10^{-3} | 5.9 × 10^{-3} |
| **G9** | Median | 20.26% | 81.62% | 80.55% | 76.19% | 97.95% | 81.24% |
|       | MAD   | 2.0 × 10^{-1} | 1.7 × 10^{-1} | 2.3 × 10^{-1} | 2.5 × 10^{-1} | 1.6 × 10^{-2} | 5.6 × 10^{-2} |
| **G19** | Median | 99.89% | 99.92% | 99.91% | 99.92% | 99.94% | 99.91% |
|       | MAD   | 4.0 × 10^{-4} | 2.7 × 10^{-4} | 2.4 × 10^{-4} | 2.1 × 10^{-4} | 2.5 × 10^{-4} | 2.6 × 10^{-4} |
| **G24** | Median | 99.59% | 99.66% | 99.66% | 99.63% | 99.63% | 99.71% |
|       | MAD   | 1.4 × 10^{-3} | 1.1 × 10^{-3} | 6.1 × 10^{-4} | 8.5 × 10^{-4} | 1.8 × 10^{-2} | 3.9 × 10^{-2} |

The results show that the acquisition functions proposed in this paper outperform naive LHS. G9 has the worst median performance of 20.26% for LHS, where the feasible space volume is extremely small (about 0.5218%). The acquisition function α_E from Echard et al. outperforms all other methods with a median informedness of 97.95%. In three out of the five problems, α_PBE achieves the best median performance, while α_E performs best in the rest of the two problems. The best median for any problem is at least 97.95% with small MAD, demonstrating the efficacy of the methods.

6 Conclusions

This paper has examined the problem of feasible space identification for computationally expensive problems. We have demonstrated an active learning approach using Bayesian models (Bayesian search) and developed a range of acquisition functions for this purpose. Our experiments show that Bayesian search outperforms naive LHS, achieving a median performance of at least 97.95% across all problems. We propose that future work focusses on batch Bayesian search when it is possible to evaluate multiple solutions in parallel.

Acknowledgements

We acknowledge the support of the Supercomputing Wales project, which is part-funded by the European Regional Development Fund (ERDF) via Welsh Government.
References

1. Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pages 908–918, 2017.
2. Clément Chevalier, Julien Bect, David Ginsbourger, Emmanuel Vazquez, Victor Picheny, and Yann Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.
3. Wei-Chih Hsu and Jianye Ching. Evaluating small failure probabilities of multiple limit states by parallel subset simulation. *Probabilistic Engineering Mechanics*, 25(3):291–304, 2010.
4. Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
5. Yaohui Li, Yizhong Wu, Jianjun Zhao, and Liping Chen. A kriging-based constrained global optimization algorithm for expensive black-box functions with infeasible initial points. *Journal of Global Optimization*, 67(1-2):343–366, 2017.
6. Victor Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pages 787–795, 2014.
7. Samineh Bagheri, Wolfgang Konen, Richard Allmendinger, Jürgen Branke, Kalyanmoy Deb, Jonathan Fieldsend, Domenico Quagliarella, and Karthik Sindhya. Constraint handling in efficient global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 673–680. ACM, 2017.
8. Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
9. C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.
10. Nicolas Knudde, Ivo Couckuyt, Kohei Shintani, and Tom Dhaene. Active learning for feasible region discovery. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 567–572. IEEE, 2019.
11. Barron J Bichon, John M McFarland, and Sankaran Mahadevan. Efficient surrogate models for reliability analysis of systems with multiple failure modes. *Reliability Engineering & System Safety*, 96(10):1386–1395, 2011.
12. Xufeng Yang, Caixing Mi, Dingyuan Deng, and Yongshou Liu. A system reliability analysis method combining active learning kriging model with adaptive size of candidate points. *Structural and Multidisciplinary Optimization*, 60(1):137–150, 2019.
13. Emmanuel Vazquez and Julien Bect. A sequential bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10):546–550, 2009.
14. Alfred M Freudenthal. Safety and the probability of structural failure. *American Society of Civil Engineers Transactions*, 1956.
15. R Rackwitz and B Fiessler. Structural reliability under combined load sequence. *Computer & Structures ASCE*, pages 2195–2199, 1978.
16. Enrique Castillo, José María Sarabia, Cristina Solares, and Patricia Gómez. Uncertainty analyses in fault trees and bayesian networks using form/sorm methods. *Reliability Engineering & System Safety*, 65(1):29–40, 1999.
17. Jorge Eduardo Hurtado. *Structural reliability: statistical learning perspectives*, volume 17. Springer Science & Business Media, 2013.
18. Guillaume Perrin. Active learning surrogate models for the conception of systems with multiple failure modes. *Reliability Engineering & System Safety*, 149:130–136, 2016.

19. Vissarion Papadopoulos, Dimitris G Giovanis, Nikos D Lagaros, and Manolis Papadrakakis. Accelerated subset simulation with neural networks for reliability analysis. *Computer Methods in Applied Mechanics and Engineering*, 223:70–80, 2012.

20. J-M Bourinet, François Deheeger, and Maurice Lemaire. Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety*, 33(6):343–353, 2011.

21. Francesco Cadini, Francisco Santos, and Enrico Zio. An improved adaptive kriging-based importance technique for sampling multiple failure regions of low probability. *Reliability Engineering & System Safety*, 131:109–117, 2014.

22. B Echard, N Gayton, and M Lemaire. Ak-mcs: an active learning reliability method combining kriging and monte carlo simulation. *Structural Safety*, 33(2):145–154, 2011.

23. William Fauriat and Nicolas Gayton. Ak-sys: an adaptation of the ak-mcs method for system reliability. *Reliability Engineering & System Safety*, 123:137–144, 2014.

24. Zhen Hu, Saideep Nannapaneni, and Sankaran Mahadevan. Efficient kriging surrogate modeling approach for system reliability analysis. *AI EDAM*, 31(2):143–160, 2017.

25. Yao Wang, Dongpao Hong, Xiaodong Ma, and Hairui Zhang. A radial-based centralized kriging method for system reliability assessment. *Journal of Mechanical Design*, 140(7):071403, 2018.

26. Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbo-b2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1689–1696. ACM, 2010.

27. JJ Liang, Thomas Philip Runnarson, Efren Mezura-Montes, Maurice Clerc, Ponnuthurai Naganathan Suganthan, CA Coello Coello, and Kalyanmoy Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics*, 41(8):8–31, 2006.

28. Yasuhiro Mori and Bruce R Ellingwood. Time-dependent system reliability analysis by adaptive importance sampling. *Structural safety*, 12(1):59–73, 1993.

29. J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

30. GPy. GPy: A gaussian process framework in python. [http://github.com/SheffieldML/GPy](http://github.com/SheffieldML/GPy) since 2012.

31. E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 329–343. Springer, 2001.

32. J. E. Fieldsend and R. M. Everson. Multi-objective optimisation in the presence of uncertainty. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 243–250. IEEE, 2005.

33. Clément Chevalier, Victor Picheny, and David Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 71:1021–1034, 2014.

34. H. J. Kushner. 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, 1964.
35. Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.
36. Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.
37. M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
38. Pritam Ranjan, Derek Bingham, and George Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008.
39. Barron J Bichon, Michael S Eldred, Laura Painion Swiler, Sandaran Mahadevan, and John M McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, 46(10):2459–2468, 2008.
40. Victor Picheny, David Ginsbourger, Olivier Roustant, Raphael T Haftka, and Nam-Ho Kim. Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7):071008, 2010.
41. Zhaoyan Lv, Zhenzhou Lu, and Pan Wang. A new learning function for kriging and its applications to solve reliability problems in engineering. *Computers & Mathematics with Applications*, 70(5):1182–1197, 2015.
42. Zhili Sun, Jian Wang, Rui Li, and Cao Tong. Lif: A new kriging based learning function and its application to structural reliability analysis. *Reliability Engineering & System Safety*, 157:152–165, 2017.
43. Francesco Biscani, Dario Izzo, Wenzel Jakob, GiacomoAcciarini, Marcus Märtens, Micky C, Alessio Mereta, Cord Kallemeyer, Sergey Lyskov, Sylvain Corlay, acxz, Benjamin Pritchard, Kishan Manani, Johan Mabille, GiacomoAcciarini, Oliver Webb, Axel Huebl, Moritz v. Looz, Manuel López-Ibáñez, jakirkham, Jeongseok Lee, hulucc, polygon, John Travers, Jakob Jordan, Ivan Smirnov, Huu Nguyen, Felipe Lema, Erik O’Leary, and Andrea Mambrini. esa/pagmo2: pagmo 2.15.0, April 2020.
44. David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
45. Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2018.
46. Carlos M Fonseca, Joshua D Knowles, Lothar Thiele, and Eckart Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 216, page 240, 2005.