ABSTRACT
Bayesian optimisation provides an effective method to optimise expensive black box functions. It has recently been applied to problems in fluid dynamics. This paper studies and compares common Bayesian optimisation algorithms empirically on a range of synthetic test functions. It investigates the choice of acquisition function and number of training samples, exact calculation of acquisition functions and Monte Carlo based approaches and both single-point and multi-point optimisation. The test functions considered cover a wide selection of challenges and therefore serve as an ideal test bed to understand the performance of Bayesian optimisation and to identify general situations where Bayesian optimisation performs well and poorly. This knowledge can be utilised in applications, including those in fluid dynamics, where objective functions are unknown. The results of this investigation show that the choices to be made are less relevant for relatively simple functions, while optimistic acquisition functions such as Upper Confidence Bound should be preferred for more complex objective functions. Furthermore, results from the Monte Carlo approach are comparable to results from analytical acquisition functions. In instances where the objective function allows parallel evaluations, the multi-point approach offers a quicker alternative, yet it may potentially require more objective function evaluations.

Keywords Bayesian optimisation · Black box functions · Gaussian processes · Physical experiments · Computer experiments · Fluid dynamics

1 Introduction
Bayesian optimisation is a widely used optimisation strategy to locate global optima of objective functions that are expensive to evaluate and whose mathematical expression is unknown [1, 2]. Thus, they cannot be optimised with conventional methods such as optimisation strategies that utilise gradient information or that
require a large number of function evaluations. Examples of such objective functions include hyperparameter
tuning of neural networks and physical experiments. For example, Wu et al. [3] optimise the hyperparameters
of a selection of models such as random forests, convolutional neural networks, recurrent neural networks
and multi-grained cascade forests. In these low-dimensional cases (two to three parameters) the model
performance was increased each time while also reducing the time to find the values compared to manual
search. Recently, Bayesian optimisation has been used successfully in the field of computational fluid
dynamics (CFD). Most notably, Talnikar et al. [4] developed a Bayesian optimisation framework for the
parallel optimisation of large eddy simulations. They used this framework (i) on a one-dimensional problem to
determine the wave speed of a travelling wave that maximises the skin-friction drag reduction in a turbulent
channel flow and (ii) on a four-dimensional problem to find an efficient design for the trailing edge of a turbine
blade that minimises the turbulent heat transfer and pressure loss. For the former, Bayesian optimisation was
able to locate a wave speed to generate skin-friction drag reduction of 60%, while for the latter within 35
objective function evaluations a design was found that reduced the heat transfer by 17% and the pressure loss
by 21%. Mahfoze et al. [5] utilised Bayesian optimisation on a four dimensional problem to locate optimal
low-amplitude wall-normal blowing strategies to reduce the skin-friction drag of a turbulent boundary-layer
with a net power saving of up to 5%, within 20 optimisation evaluations. Morita et al. [6] considered three
CFD problems: the first two problems concerned the shape optimisation of a cavity and a channel flow. The
third problem optimised the hyperparameters of a spoiler-ice model. Lastly, Nabae et al. [7] maximised the
skin-friction drag reduction in a turbulent channel flow by optimising the velocity amplitude and the phase
speed of a traveling wave-like wall deformation. They achieved a maximum drag reduction of 60.5%.
While these examples show that Bayesian optimisation works on hyperparameter tuning and CFD problems,
there is no extensive empirical study on the many different types of acquisition functions proposed in the
literature. In particular, there is a research gap regarding the variability in performance when Bayesian
optimisation is applied to distinct challenges and varying numbers of initial training points. This paper
aims to address this gap by considering a wide range of acquisition functions and a wide range of problems
with increasing levels of complexity that are, thus, increasingly difficult to solve. For these experiments,
the Newcastle University Bayesian optimisation (NUBO) framework that is currently being developed for
optimising fluid flow problems is utilised.
The paper is structured as follows. First, the Bayesian optimisation algorithm, in particular the underlying
Gaussian process (GP) and different classes of acquisition functions, are reviewed. Then, synthetic benchmark
functions and their advantages over other types of common benchmarking methods are discussed. Finally, four
different sets of simulations are presented. These experiments compare (i) multiple analytical single-point
acquisition functions, (ii) varying numbers of initial training points, (iii) acquisition functions utilising Monte
Carlo sampling and (iv) various multi-point acquisition functions.

2 Bayesian optimisation

Bayesian optimisation aims to find the global optimum of an expensive to evaluate objective function, whose
mathematical expression is unknown or does not exist, in a minimum number of evaluations. It first fits a
surrogate model, most commonly a GP, to some initial training data. The GP reflects the current belief about
the objective function and is used to compute heuristics, called acquisition functions. When optimised, these
functions suggest the candidate point that should be evaluated next from the objective function. After the
response for the new candidate point is computed from the objective function, it is added to the training data
and the process is repeated until a satisfying solution is found or a predefined evaluation budget is exhausted [1].
The following sections provide an overview of the two main components of Bayesian optimisation, the GP and
the acquisition functions.

2.1 Gaussian process

Consider an objective function \( f \) that allows the evaluation of one point \( x \), or multiple points \( X \) yielding one
or multiple observations \( f(x) = y \) and \( f(X) = y \) respectively. A popular choice for the surrogate model to
represent such an objective function is a GP. A GP is a stochastic process for which any finite set of points
can be represented by a multivariate Normal distribution. This so called prior is defined by a mean function
\( \mu_0 : \mathcal{X} \rightarrow \mathbb{R} \) and a positive definite covariance function or kernel \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \). These induce a mean
vector \( \mathbf{m}_i := \mu_0(x_i) \) and a covariance matrix \( K_{i,j} := k(x_i, x_j) \) that define the prior distribution of the GP as
\[
 f \mid X \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{1}
\]
where $f \mid X$ are the unknown responses from the objective function given a collection of points $x_{1:n}$. Given a set of observed design points $D_n = \{(x_i, y_i)\}_{i=1}^n$ and a new point also called a candidate point $x$, the posterior predictive distribution of $x$ can be computed:

$$Y \mid D_n \sim \mathcal{N}(\mu_n(x), \sigma_n^2(x))$$

with the posterior mean and variance

$$\mu_n(x) = \mu_0(x) + k(x)^T(K + \sigma^2 I)^{-1}(y - m)$$

$$\sigma_n^2(x) = k(x, x) - k(x)^T(K + \sigma^2 I)^{-1}k(x)$$

where $K = k(x_{1:n}, x_{1:n})$ and $k(x) = k(x, x_{1:n})$.

For the experiments presented in this paper, the zero mean function is implemented as it fulfills the theoretical properties required to compute the variable hyperparameter $\beta$ for the Gaussian Process Upper Confidence Bound (GP-UCB) algorithm as derived in Srinivas et al. [9] (see Section II. B.). While there are many covariance kernels used in the literature, this paper implements the Matern kernel with $\nu = 5/2$ with an individual length scale parameter for each input dimension as proposed by Snoek et al. [8] for practical optimisation problems, as other kernels such as the squared-exponential kernel and radial basis function kernel can be too smooth for applied problems such as physical experiments. This kernel requires an additional parameter, the output scale $\tau^2$, to allow the kernel to be scaled above a certain threshold by multiplying the output scale with the posterior variance $\tau^2\sigma_n^2(x)$ [2]. A nugget term $\nu$ is added to the diagonal of the covariance matrix to take possible errors, measurement and other, into consideration where $I$ is the identity matrix. Nuggets have been shown to improve the computational stability and the performance of the algorithm such as the coverage and robustness of the results when using sparse data [10]. Additionally, Gramacy et al. [11] give a detailed reasoning why nuggets should be added to deterministic problems that exceeds solely technical advantages. For example, while there might not be a measurement error in a deterministic simulation, the simulation itself is biased as it only approximates reality. In such cases, it makes sense to include a model to take possible biases into account. The hyperparameters of the GP, in this case the nugget, the output scale and the length scales of the Matern kernel, can be estimated directly from the data, for example using maximum likelihood estimation (MLE) or maximum a posteriori (MAP) estimation.

2.2 Acquisition functions

Acquisition functions utilise the posterior distribution of the GP, that is the surrogate model representing the objective function, to propose the next point to sample from the objective function. Shahriari et al. [1] group acquisition functions into four categories: improvement-based policies, optimistic policies, information-based policies and portfolios. This section presents some popular representatives of each group, which we shall focus on in this paper.

Improvement-based policies such as Probability of Improvement (PI) and Expected Improvement (EI) propose points that are better than a specified target, for example the best point evaluated so far, with a high probability. Given the best point so far $x^*$ and its response value $y^* = f(x^*)$, PI (5) and EI (6) can be computed as:

$$\alpha_{PI}(x; D_n) = \Phi\left(\frac{z}{\sigma_n(x)}\right)$$

$$\alpha_{EI}(x; D_n) = z\Phi\left(\frac{z}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{z}{\sigma_n(x)}\right)$$

respectively, where $z = \mu_n(x) - y^*$ and $\Phi$ and $\phi$ are the CDF and the PDF of the standard Normal distribution. The Upper Confidence Bound (UCB) acquisition function is an optimistic policy. It is optimistic with regards to the uncertainty (variance) of the GP, that is UCB assumes the uncertainty to be true. In contrast to the improvement-based methods, UCB [4] has a tuneable hyperparameter $\beta_n$ that balances the exploration-exploitation trade-off. This trade-off describes the decision the Bayesian optimisation algorithm has to make at each iteration. The algorithm could either explore and select new points in an area with high uncertainty or it could exploit and select points in areas where the GP makes a high prediction [1]. Thus, choosing how to set the trade-off parameter $\beta_n$ is an important decision. Common strategies for setting $\beta_n$ include fixing it to a particular value or varying it, as in the GP-UCB algorithm [9].

$$\alpha_{UCB}(x; D_n) = \mu_n(x) + \sqrt{\beta_n\sigma_n(x)}$$

Portfolios consider multiple acquisition functions at each step and choose the best-performing one. The Hedge algorithm outlined by Hoffmann et al. [13] is used in this paper. It requires the computation and optimisation

3
While there are various methods of benchmarking the performance of optimisation algorithms, such as with a proxy target, usually the minimal or maximal response value encountered so far. After one iteration of the so-called temperature parameter \(\tau\), where \(z\) would be preferred in a different iteration, the Hedge algorithm should in theory select the best acquisition function at each iteration, yielding a better solution than when just considering one individual acquisition function.

In addition to these analytical acquisition functions, Monte Carlo sampling can be implemented. While this is an approximation of the analytical functions, Monte Carlo sampling does not require the often-times strenuous explicit computations involved with the analytical functions, especially when considering multi-point approaches. Information-based policies, in particular those using entropy, aim to find a candidate point \(x\) that reduces the entropy of the posterior distribution \(p(x|D_n)\). While Entropy Search (ES) and Predictive Entropy Search (PES) are computationally expensive, Wang et al. \cite{14,15} introduce Max-value Entropy Search (MES), a method that uses information about simple to compute maximal response values instead of costly to compute entropies.

The analytical acquisition functions can be used with Monte Carlo sampling by reparameterising equations \cite{5,7,16}. Then, they can be computed by sampling from the posterior distribution of the GP instead of computing the acquisition functions directly:

\[
\alpha_{MC\ PI}(x; D_n) = \sigma \left( \frac{\max(\mu_n(x) + Lz) - y^*}{\tau} \right) \tag{8}
\]

\[
\alpha_{MC\ EI}(x; D_n) = \max (0, \max (\mu_n(x) + Lz) - y^*) \tag{9}
\]

\[
\alpha_{MC\ UCB}(x; D_n) = \max \left( \mu_n(x) + \sqrt{\beta \pi/2 |Lz|} \right) \tag{10}
\]

where \(z\) is a vector containing samples from a Standard Normal distribution \(z \sim N(0, I)\) and \(L\) is the lower triangular matrix of the Cholesky decomposition of the covariance matrix \(K = LL^T\). The softmax function \(\sigma\) with the so-called temperature parameter \(\tau\) enables the reparameterisation of PI by approximating necessary gradients that are otherwise almost entirely equal to 0. As \(\tau \to 0\), this approximation turns precise as described in Section 2 of Wilson et al. \cite{16}.

It is straightforward to extend the reparameterisations of the analytical acquisition functions to multi-point methods that propose more than one point at each iteration. In this case, samples are taken from a joint distribution (including training points and new candidate points) that is optimised with respect to the new candidate points. This optimisation can be performed jointly, i.e. optimising the full batch of points simultaneously, or in a sequential manner where a new joint distribution is computed for each point in the batch including the previously computed batch points, but only optimising it with respect to the newest candidate point \cite{17}. While analytical functions cannot be extended naturally to the multi-point case, there exist some frameworks that allow the computation of batches. The Constant Liar framework uses EI in combination with a proxy target, usually the minimal or maximal response value encountered so far. After one iteration of optimising EI, the lie is taken as the true response for the newly computed candidate point and the process is repeated until all points in the batch are computed. At this point, the true responses of the candidate points are evaluated via the objective function and the next batch is computed \cite{18}. Similarly, the Gaussian Process Batched Upper Confidence Bound (GP-BUCB) approach leverages the fact that the variance of the multivariate Normal distribution can be updated based solely on the inputs of a new candidate point. However, to update the posterior mean, the response value is required. GP-BUCB updates the posterior variance after each new candidate point, while the posterior mean stays the same for the full batch. After all points of a batch are computed, the response values are gathered from the objective function and the next batch is computed \cite{19}.

### 3 Benchmark functions

While there are various methods of benchmarking the performance of optimisation algorithms, such as sampling objective functions from Gaussian processes or optimising parameters of models \cite{8,13,14}, this paper focuses on using synthetic test functions. Synthetic test functions have the main advantages that their global optima and their underlying shape are known. Thus, when testing the algorithms outlined in section II on these functions, the distance from their results to the true optimum can be evaluated. For other benchmarking methods, this might not possible as, for example, in the case of optimising the hyperparameters of a model, the true optimum, i.e. the solution optimising the performance of the model, is rarely known. Hence, results can only be discussed relative to other methods, ignorant of the knowledge of how close any method actually is to the true optimum. Furthermore, knowing the shape of the test function is advantageous as it provides
information on how challenging the test function is. For example, knowing that a test function is smooth and has a single optimum indicates that it is less complex and thus less challenging than a test function that possesses multiple local optima, in each of which the algorithm potentially could get stuck.

This paper assesses the Bayesian optimisation algorithms on nine different synthetic test functions spanning a wide range of challenges. Table 1 presents the test functions and gives details on the number of input dimensions, the shape and the number of optima to give an idea about the complexity of the individual functions, while Figure 1 shows some of the functions in three-dimensional space and indicates the increasing complexity of the functions. The number of input dimensions for the test functions was chosen to be equal to or greater than the number in the simulations recently published in the fluid dynamics community, which is typically three to eight [4–7]. The 10D Sphere and 10D Dixon-Price functions are less challenging problems with a high degree of smoothness and only one optimum. They are therefore considered for a higher number of input dimensions. The 8D Griewank function adds a layer of complexity by introducing oscillatory properties. The 6D Hartmann function increases the level of difficulty further through its multi-modality. It has six optima with only one global optimum. This function is also considered in two noisy variants to simulate typical measurement uncertainty encountered during experiments in fluid dynamics. In particular, the standard deviation of the added Gaussian noise is calculated so that it represents the measurement errors of state-of-the-art Micro-Electro-Mechanical-Systems (MEMS) sensors which directly measure time-resolved skin-friction drag in turbulent air flows (e.g. the flow over an aircraft); that is, 1.4 to 2.4% in an experimental setting [20]. Factoring in the range of the 6D Hartmann function, the corresponding standard deviations for the Gaussian noise, taken as a 99% confidence interval, are therefore 0.0155 and 0.0266 respectively. The most complex test functions are the 5D Michalewicz and the 6D Ackley functions. While the former has 120 optima and steep ridges, the latter is mostly flat with a single global minimum in the centre of the space. The gradient of the function close to the optimum and the large flat areas represent a great level of difficulty, as illustrated in the bottom row of Figure 1.

---

1The number of input dimensions for the functions was reduced to two to allow the representation in three-dimensional space.

2This paper considers a modified Ackley function with $a = 20.0$, $b = 0.5$ and $c = 0.0$.

---
| Test function   | Number of input dimensions D | Shape       | Number of optima |
|-----------------|-----------------------------|-------------|-----------------|
| Sphere          | 10                          | Bowl-shaped | 1               |
| Dixon-Price     | 10                          | Valley-shaped | 1              |
| Griewank        | 8                           | Oscillatory | 1               |
| Hartmann        | 6                           | Multi-modal | 6               |
| Noisy Hartmann  | 6                           | Noisy       | 6               |
| Michalewicz     | 5                           | Steep edges | 120             |
| Ackley          | 6                           | Mostly flat | 1               |

4 Results

This section focuses on the results for four of the nine test functions that were considered in this paper. Results for the other four test functions, as well as more extensive tables, can be found appendix. The 8D Griewank function, the 6D Hartmann function in variations without noise and with the high noise level and the 6D Ackley function are examined here, as they represent increasing levels of complexity and come with unique challenges as illustrated in section III. Mirroring real world applications, a budget for the number of evaluations was imposed on the test functions: 200 evaluations for the Griewank and Hartmann functions and 500 for the more complex Ackley function. If not otherwise stated, the number of initial training points is equivalent to five points per input dimension of the given test function, i.e. 40 training points for the Griewank function and 30 training points for the Hartmann and Ackley functions. For each of the methods in the following sections, 50 different optimisation runs were computed to investigate how sensitive the methods are to varying initial training points. Each run was initialised with a different set of initial training points sampled from a Maximin Latin Hypercube design \cite{21}. However, the points were identical for all methods for a specific test function. All experiments were run on container instances on the cloud with the same specifications (two CPU cores and 8GB of memory) to make runs comparable.

Overall, four different sets of experiments are considered. First, analytical single-point acquisition functions are compared. Second, the effects of a varying number of initial training points are investigated. Third, analytical methods are compared with Monte Carlo methods and, lastly, multi-point or batched methods are compared to the single-point results. As a baseline of performance, space filling designs were sampled from a Latin Hypercube.

4.1 Analytical single-point acquisition functions

Section II describes four different groups of acquisition functions: improvement-based, optimistic, portfolios, and entropy-based. In this section, representative functions from the first three groups are tested on the synthetic test functions outlined above \cite{1}. The focus lies on analytical single-point acquisition functions as they are widely used and thus present a natural starting point. Overall, seven different methods are considered: PI, EI, UCB with a variable and fixed \( \beta \) (5.0 and 1.0) and a Hedge portfolio that combines the PI, EI and variable UCB acquisition functions. For a detailed discussion of these functions see section II.

Figure 2 presents the best solutions, defined by the output value closest to the global optimum, found so far at each evaluation. The outputs are normalised to the unit range where 1.0 represents the global optimum. Most methods perform very well on the Griewank function, all reaching 1.00, and both variations of the Hartmann function with and without added noise (all reaching 1.00 and >0.97 respectively). The acquisition functions all find the optimum or a solution fairly close to the optimum within the allocated evaluation budget. However, PI and variable UCB typically take more evaluations to find a solution close to the optimum. All acquisition functions perform noticeably better than the Latin Hypercube benchmark and also exhibit much less variation over the 50 runs, as indicated by the 95% confidence intervals. There is little difference between the Hartmann function with and without added noise, indeed the results are almost identical. The Ackley function on the other hand is more challenging. While the UCB methods still perform very well (all > 0.96) the performance of the portfolio and PI decrease slightly (0.91 and 0.88 respectively) and EI performs considerably worse.

\footnote{The entropy-based approach is considered in section II. C. of this paper.}
Figure 2: Performance plots for analytical single-point acquisition functions with five initial starting points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.

Similar conclusions can be drawn from Table 2, which presents the area under the curve (AUC) of each method. The AUC indicates how quickly the individual methods find solutions near the optimum. A perfect score (1.0) would indicate that the algorithm finds the optimum perfectly at the first iteration. The lower the score (a) the further the algorithm is away from the optimum and (b) the more evaluations are required for the algorithm to find promising solutions. While all methods score at least 0.98 for the Griewank function with a standard error of at most 0.01, the scores worsen slightly for the Hartmann function and significantly for the Ackley function. In particular, the AUC for EI for the Ackley function is very poor (0.59) and exhibits a high degree of variability between the 50 runs (standard error: 0.15). At the opposite end of the performance spectrum lie the optimistic methods, i.e. the UCB with variable and fixed $\beta$, that perform very well on all test functions with low standard errors. While the choice of acquisition function is less relevant for simple and moderately complex objective functions such as the Griewank or Hartmann functions, it is instrumental for solving challenging problems such as the Ackley function, especially if they are characterised by large flat areas. Here the optimistic acquisition functions are advantageous and should be preferred over the Hedge portfolio and improvement-based approaches.

---

This may only be true for the specific portfolio defined previously. Implementing a different collection of acquisition functions could yield different results.
4.2 Varying number of initial training points

The training points used to initialise the Bayesian optimisation algorithm directly affect the surrogate model, i.e. the Gaussian process, that in turn is a representation of the objective function. A larger number of training points yields a Gaussian process that will typically represent the objective function more closely as it incorporates more points and thus more information. However, the more evaluations of the total budget are allocated for these initial training points, the fewer points can be evaluated as part of the Bayesian optimisation algorithm. This trade-off indicates that the number of training points (and by extension their selection) is an important choice in Bayesian optimisation and should be considered thoroughly. This section explores this trade-off by taking the same experimental setup as the previous section and varying the number of initial training points. Overall, setups with one, five and ten initial points per input dimension of the objective function are considered.

Figures 3 and 4 depict the performance plots for the case with one and ten training points per dimension respectively. If the number of points is reduced to one point per dimension, the individual methods find solutions that are virtually identical to the results with five training points per dimension. EI still performs much worse than other methods for the Ackley function. Furthermore, most methods seem to find their best solution in a comparable or just slightly higher number of evaluations than before, as the AUC values in Table 3 show. This is expected, as using fewer initial training points means that the Bayesian optimisation algorithm has less information at the earlier iterations than when using more initial training points. However, the difference in mean AUC is small and the results suggest that Bayesian optimisation makes up for this lack of information quickly. For example, on average the PI and Hedge portfolio perform worse for the Hartmann function with a mean AUC that is 0.09 and 0.06 smaller. The variability between runs of the Hedge algorithm also rises, as indicated by a AUC standard error that is 0.08 higher. PI and the portfolio also perform worse for the Noisy Hartmann function, where the mean AUC decreased by 0.05 for both methods. Intuitively, this makes sense as the surrogate model includes less information than before and thus it takes more evaluations to find a good solution. In early iterations, the individual methods deviate from one-another more than when five training points per dimension are used.

If the number of starting points is increased to ten points per dimension, there is essentially no change to the case with just five starting points per dimension for the Griewank and the two Hartmann functions. For the Ackley function, however, the performance in terms of both the best solution found and the AUC worsen (Table 4) for EI and, most significantly, for PI. The average best solution for the latter decreases by 0.37 to only 0.51 while EI decreases by 0.10 to 0.53. Both policies struggle with the large area of the test function that gives the same response value and is hence flat (see section III.). The optimistic policies and the portfolio perform much better and no real change is noticeable to results of section II. A.

---

5See appendix

---
These results suggest that choosing a larger number of training points to initialise the Bayesian optimisation algorithm cannot necessarily be equated with better performance and solutions. This is particularly true considering that there was no improved performance when increasing from five to ten points per input dimension. On the other hand, reducing the number of training points did not yield results that were much worse. Overall, a similar picture as in the previous section emerges: While all methods perform well on simpler problems, optimistic policies achieve the best results on the more challenging problems independent of the number of starting points. For the test functions we considered, five training points per dimension appeared to be sufficient, with no discernible improvement when moving to ten training points, and a small loss of performance when reducing to one training point per dimension.
Figure 4: Performance plots for analytical single-point acquisition functions with ten initial starting points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.

Table 3: Averaged area under the curve with standard error for analytical single-point acquisition functions with one initial training point per input dimension.

| Method      | Griewank | Hartmann | Noisy Hartmann | Ackley |
|-------------|----------|----------|----------------|--------|
| PI          | 0.97     | 0.82     | 0.90           | 0.73   |
|             | (± 0.01) | (± 0.09) | (± 0.06)       | (± 0.11) |
| EI          | 0.99     | 0.95     | 0.95           | 0.58   |
|             | (± 0.00) | (± 0.03) | (± 0.03)       | (± 0.13) |
| UCB (variable) | 0.97     | 0.91     | 0.91           | 0.82   |
|             | (± 0.01) | (± 0.04) | (± 0.03)       | (± 0.01) |
| UCB (β=5)   | 0.98     | 0.94     | 0.95           | 0.87   |
|             | (± 0.00) | (± 0.03) | (± 0.03)       | (± 0.02) |
| UCB (β=1)   | 0.99     | 0.93     | 0.93           | 0.85   |
|             | (± 0.00) | (± 0.08) | (± 0.09)       | (± 0.06) |
| Hedge       | 0.98     | 0.89     | 0.92           | 0.76   |
|             | (± 0.01) | (± 0.11) | (± 0.05)       | (± 0.09) |
Table 4: Averaged area under the curve with standard error for analytical single-point acquisition functions with ten initial training points per input dimension.

| Method          | Griewank (±0.00) | Hartmann (±0.05) | Noisy Hartmann (±0.02) | Ackley (±0.25) |
|-----------------|------------------|------------------|------------------------|---------------|
| PI              | 1.00             | 0.93             | 0.96                   | 0.43          |
| EI              | 1.00             | 0.97             | 0.97                   | 0.41          |
| UCB (variable)  | 1.00             | 0.95             | 0.96                   | 0.87          |
| UCB (β=5)       | 1.00             | 0.97             | 0.98                   | 0.91          |
| UCB (β=1)       | 1.00             | 0.96             | 0.98                   | 0.86          |
| Hedge           | 1.00             | 0.96             | 0.97                   | 0.78          |

4.3 Monte Carlo single-point acquisition functions

The previous experiments considered analytical acquisition functions. This section goes one step further and assesses the Monte Carlo approach outlined in section II. As not all acquisition functions can be rewritten to suit such an approach, the experiments are restricted to PI, EI and UCB with variable and fixed β. Additionally, MES is introduced as a new method.

For the Griewank and both Hartmann functions all results are almost identical to the analytical case except for the variability between runs which increased slightly for some of the methods. However, Figure 5 clearly shows that the performance for PI and EI tested on the Ackley function decreased significantly. The average best solutions decreased by 0.68 and 0.56 and the average AUC (Table 5) decreased by 0.59 and 0.54 respectively. Table 5 shows that MES performs well on the Griewank and Hartmann functions, all reaching an AUC of above 0.97 with low standard errors. When it comes to the more complex Ackley function, however, MES performs much worse. With a mean AUC of 0.49 its performance ranks below the optimistic acquisition functions but still above the improvement-based methods. Earlier sections showed that improvement-based policies (in particular EI) perform poorly on the Ackley function when using analytical acquisition functions, but the results from this comparison show that their performance suffers even more severely when using the Monte Carlo approach. One reason for this could be that the Monte Carlo variants are essentially an approximation of the analytical acquisition functions as discussed in section II. However, there seems to be little to no change when using optimistic policies. This suggests that, similar to previous sections, optimistic policies should be preferred when optimising complex and challenging objective functions with large flat areas using Monte Carlo acquisition functions.
Figure 5: Performance plots for Monte Carlo single-point acquisition functions with five initial starting points per input dimension. Solid lines represent the mean over the 50 runs while the shaded areas represent the 95% confidence intervals.

Table 5: Averaged area under the curve with standard error for Monte Carlo single-point acquisition functions with five initial training points per input dimension.

| Method           | Griewank | Hartmann | Noisy Hartmann | Ackley |
|------------------|----------|----------|----------------|--------|
|                  |          |          |                |        |
| MC PI            | 0.99     | 0.92     | 0.95           | 0.17   |
|                  | (± 0.00) | (± 0.06) | (± 0.04)       | (± 0.12)|
| MC EI            | 1.00     | 0.97     | 0.97           | 0.05   |
|                  | (± 0.00) | (± 0.02) | (± 0.04)       | (± 0.03)|
| MC UCB (variable)| 0.98     | 0.95     | 0.95           | 0.86   |
|                  | (± 0.01) | (± 0.02) | (± 0.04)       | (± 0.01)|
| MC UCB (β=5)    | 1.00     | 0.97     | 0.98           | 0.91   |
|                  | (± 0.00) | (± 0.02) | (± 0.03)       | (± 0.01)|
| MC UCB (β=1)    | 1.00     | 0.96     | 0.97           | 0.89   |
|                  | (± 0.00) | (± 0.07) | (± 0.07)       | (± 0.03)|
| MES              | 0.99     | 0.97     | 0.97           | 0.49   |
|                  | (± 0.00) | (± 0.02) | (± 0.03)       | (± 0.12)|
4.4 Multi-point acquisition functions

The previous sections focused on single-point approaches, where each iteration of the Bayesian optimisation algorithm yields one new point that is sampled from the objective function before the next iteration is started. While this makes sense for objective functions that can be evaluated quickly or do not allow parallel evaluations, it might draw out the optimisation process needlessly when objective functions take a long time to evaluate and allow parallel evaluations. Thus, this section implements multi-point acquisition functions that propose a batch of points at each iteration, which are then evaluated simultaneously before the next iteration.

Section II outlined how Monte Carlo approaches using the reparameterisation trick can be extended to compute batches naturally\(^6\). Hence, we considered the same acquisition functions as in the previous section but this time for a batch size of five points. Each of the acquisition functions is optimised with two different methods, sequentially and jointly. The latter optimises the acquisition function based on a joint distribution that includes training points and new points and optimises all new points in a single step. The former recomputes the joint distribution each time a new point is found and only optimises with respect to the newest point. For example, for a batch of five points this process is repeated five times\(^7\). This approach, also known as greedy optimisation, might be preferable and can yield better results\(^{22}\). While analytical functions cannot be extended to the multi-point case as easily as the Monte Carlo evaluations, there are some frameworks that allow the computation of batches. This section considers the Constant Liar approach with a lie equivalent to the minimal (CL min) and maximal (CL max) value so far\(^{18}\) and the GP-BUCB algorithm, an extension of the UCB function\(^{19}\). For more details see Section II.

Figure 6 shows the results of some multi-point acquisition functions. They essentially find identical solutions, on average, to the single-point approach for the Griewank function and Hartmann function with and without noise\(^7\). While there are no changes to the AUC for the Griewank function, there are some differences for the Hartmann functions that suggest that methods require a different number of evaluations to find the best value, as Table 6 shows. Although the joint Monte Carlo approach for PI and UCB (variable and \(\beta=5\)) have a lower mean AUC, the decrease is less severe than for the GP-BUCB methods that all decrease by 0.08 to 0.09. The other methods perform comparably to the single-point case and the variability between the runs of the sequential and joint Monte Carlo UCB with \(\beta=1\) even decreases. For the noisy Hartmann test function, the results are very similar to the Hartmann function without noise. In general, over all experiments there was no real drop in performance that can be declared as a result of adding noise up to 2.4\% of the overall objective function range.

Most methods find similar best solutions to the Ackley test function as their single-point counterparts. However, there are some changes in the improvement-based policies: the sequential Monte Carlo EI and PI, and CL max, find solutions that are better than before (by 0.33, 0.04 and 0.11 respectively). At the opposite end, CL min and joint Monte Carlo PI provide inferior solutions in the batched case (0.13 and 0.08 respectively). Looking at how quickly the individual methods find good values on average, i.e. the AUC, it is clear that the analytical multi-point and joint Monte Carlo methods perform worse than the single-point implementations for the Ackley function: the AUC of all analytical multi-point methods worsen by 0.19 – 0.24, except for CL max that improved by 0.08. Similarly, all joint methods provide poorer performance than in the sequential single-point optimisation where UCB with \(\beta=1\) sees the largest drop of 0.11. EI is the exception and stays about the same. The biggest improvement is provided by the sequential Monte Carlo EI with an increase of 0.32. However, this improvement is mainly caused by the very poor performance of the single-point Monte Carlo EI acquisition function. Furthermore, this approach comes with a higher variability, as the standard error of the AUC rises by 0.19.

In terms of the best solutions found, multi-point methods present a good alternative to single-point acquisition functions. They generally find best solutions comparable to the single-point approach, but for a slightly larger number of objective function evaluations. However, while it requires more evaluations, the multi-point approach would still be faster when computing batches in parallel. Multi-point acquisition functions will, therefore, be most beneficial for expensive to evaluate objective functions that can be evaluated in parallel. With some exceptions, this benefit requires a higher number of evaluations until solutions comparable to single-point methods are found. Overall, the optimistic methods, combined with sequential optimisation, appear to be favourable over the rest, as the red lines in Figure 6 clearly show.

---

\(^6\)MES does not use reparameterisation and is thus not naturally extendable to the multi-point setting.

\(^7\)Only selected methods are shown in the plot. Particularly, two variations of the UCB approach are not included as they are almost identical to the UCB with \(\beta=5\). See appendix for all methods.
Table 6: Averaged area under the curve with standard errors for multi-point acquisition function with five initial training points per input dimension.

| Type               | Method | Griewank | Hartmann | Noisy Hartmann | Ackley |
|--------------------|--------|----------|----------|----------------|-------|
| Sequential Monte Carlo | PI     | 0.99     | 0.92     | 0.94           | 0.22  |
|                    |        | (± 0.00) | (± 0.03) | (± 0.02)       | (± 0.08) |
|                    | EI     | 1.00     | 0.95     | 0.97           | 0.37  |
|                    |        | (± 0.00) | (± 0.03) | (± 0.02)       | (± 0.22) |
|                    | UCB (variable) | 0.98   | 0.94     | 0.94           | 0.84  |
|                    |        | (± 0.01) | (± 0.03) | (± 0.03)       | (± 0.01) |
|                    | UCB (β=5) | 0.99   | 0.95     | 0.96           | 0.88  |
|                    |        | (± 0.00) | (± 0.02) | (± 0.02)       | (± 0.02) |
|                    | UCB (β=1) | 1.00   | 0.96     | 0.97           | 0.86  |
|                    |        | (± 0.00) | (± 0.02) | (± 0.02)       | (± 0.02) |
| Joint Monte Carlo | PI     | 0.99     | 0.86     | 0.89           | 0.09  |
|                    |        | (± 0.00) | (± 0.05) | (± 0.04)       | (± 0.06) |
|                    | EI     | 1.00     | 0.95     | 0.96           | 0.06  |
|                    |        | (± 0.00) | (± 0.03) | (± 0.02)       | (± 0.03) |
|                    | UCB (variable) | 0.97   | 0.89     | 0.90           | 0.81  |
|                    |        | (± 0.02) | (± 0.04) | (± 0.04)       | (± 0.02) |
|                    | UCB (β=5) | 0.99   | 0.94     | 0.95           | 0.87  |
|                    |        | (± 0.00) | (± 0.03) | (± 0.03)       | (± 0.02) |
|                    | UCB (β=1) | 1.00   | 0.96     | 0.97           | 0.78  |
|                    |        | (± 0.00) | (± 0.02) | (± 0.02)       | (± 0.06) |
| Analytical | CL min | 0.99     | 0.94     | 0.95           | 0.40  |
|                    |        | (± 0.00) | (± 0.02) | (± 0.02)       | (± 0.08) |
|                    | CL max | 1.00     | 0.95     | 0.96           | 0.67  |
|                    |        | (± 0.00) | (± 0.03) | (± 0.03)       | (± 0.09) |
|                    | BUCB (variable) | 0.98   | 0.85     | 0.85           | 0.65  |
|                    |        | (± 0.02) | (± 0.05) | (± 0.05)       | (± 0.03) |
|                    | BUCB (β=5) | 0.98   | 0.88     | 0.90           | 0.66  |
|                    |        | (± 0.00) | (± 0.05) | (± 0.05)       | (± 0.05) |
|                    | BUCB (β=1) | 0.99   | 0.88     | 0.90           | 0.65  |
|                    |        | (± 0.00) | (± 0.06) | (± 0.06)       | (± 0.05) |
5 Discussion

Six main conclusions can be drawn from the simulation results presented in detail above. This section discusses these findings and relates them back to the applied problems that motivated this work, i.e. optimising hyperparameters of neural networks and conducting experiments in fluid dynamics.

The first findings concern the choice of acquisition functions related on the complexity of the problem. The results show that this choice is less important when optimising simpler objective functions. Improvement-based, optimistic and information-based acquisition functions, as well as portfolios, performed well on a wide range of synthetic test functions with up to ten input dimensions. However, for more complex functions, such as the Ackley function, the optimistic methods performed significantly better than the rest and thus should be favoured. For the Bayesian optimisation algorithm, this means that all acquisition functions considered in this paper can yield good results. But the choice of acquisition function must be considered more carefully with increasing complexity of the objective function. This indicates the importance of expert knowledge when applying Bayesian optimisation to a specific problem, such as a drag reduction problem in fluid dynamics. Basing the choice of acquisition function on specific knowledge about the likely complexity of the objective function could potentially increase the performance of the Bayesian optimisation algorithm significantly. In cases where no expert knowledge, or other information, about the objective function is accessible, results suggest that the optimistic methods are a good starting point.

Secondly, the results suggest that the number of initial training points is not critical in achieving good performance from the algorithm. In fact, there is only little difference when increasing the number of starting points from one point per input dimension to ten points per dimension. While the performance of the acquisition functions differs initially for the former case, they still find comparable results to the runs with five or ten points per input dimension over all iterations. This means that Bayesian optimisation explores the space efficiently even when provided with only a few training points. Deciding on the number of starting points is an important decision in applied problems where evaluating a point is expensive. For example, when evaluating a set of hyperparameters for a deep neural network, the model has to be trained each time, racking up costs in

§See appendix for more examples to reinforce this result.
time and computing resources. In most cases the problem boils down to dividing a predefined budget into two parts: evaluations to initialise the algorithm and evaluations for points proposed by the Bayesian optimisation algorithm. The decision of how many points to allocate for the training data is important as using too many could mean that the Bayesian optimisation algorithm does not have enough evaluations available to find a good solution, i.e., the budget is exhausted before a promising solution is found. The simulations in this paper suggest that this decision might not be as complex as it initially seems, as only a few training points are necessary for the algorithm to achieve good results. Taking this approach of using only a small number of evaluations for training points saves more evaluations for the optimisation itself.

The third finding regards the information-based acquisition functions. Through the range of different simulations PI and EI failed to find good results for the Ackley function. The reason for this is the large area of the parameter space where all response values are identical and thus the response surface is flat. As discussed in section II., improvement-based acquisition functions propose a point that is most likely to improve upon the previous best point. With a flat function like the Ackley function it is likely that all of the initial starting points fall into the flat area (especially in higher dimensions). The posterior mean of the Gaussian process will then be very flat, and will predict that the underlying acquisition function is flat. This leads to a very flat acquisition function, as most input points will have a small likelihood of improving upon the previous best points. Gramacy et al. [2] mention this problem when optimising a flat EI acquisition function. This results in the evaluation of points that are not optimal, which is especially problematic when the shape of the objective function is not known and flat regions cannot be ruled out a priori. If such properties are possible in a particular applied problem, the results suggest that using a different acquisition function, such as an optimistic policy, achieves better results.

Fourthly, the simulations show that Monte Carlo acquisition functions yield comparable results to analytical functions. These functions use Monte Carlo sampling to compute the acquisition functions, instead of analytically solving them. Utilising sampling to compute a function that can also be solved analytically might not appear useful at first glance as it is essentially just an approximation of the analytical results. However, this approach makes it straightforward to compute batches of candidate points (as outlined in section II.), which presents the foundation for the next finding.

Fifth, multi-point acquisition functions perform comparably to single-point approaches. All methods found good solutions across the range of problems considered, with the exception of improvement-based methods for the Ackley function as discussed previously. Multi-point approaches are particularly beneficial for cases in which the objective function is expensive to evaluate and allows parallel evaluations (e.g., high-fidelity turbulence resolving simulations [5]). For these problems, the total time required to conclude the full experiment can be reduced as multiple points are evaluated in parallel each time. However, more evaluations might be required in total to achieve results comparable to the single-point approach.

Lastly, the results showed no decrease in performance when adding noise to the objective function, in this case the 6D Hartmann function. When optimising the function with low and high noise (corresponding to the measurement error range of MEMS sensors to mirror the circumstances of an applied example in fluid dynamics) the same solutions were found, in a comparable number of evaluations, as for the deterministic case. This result can be attributed to the nugget that is added to the deterministic and noisy cases, as discussed in section II. The results are promising, as they show that Bayesian optimisation can handle noisy objective functions just as efficiently as deterministic functions. This enables the use of Bayesian optimisation for physical experiments where measurements cannot be taken without noise, but also for simulations where other noise can be introduced unknowingly.

While these findings show that the use of Bayesian optimisation to optimise unknown systems is promising, some limitations should be noted. Firstly, the acquisition functions considered in this paper represent only a subset of those available. The general results inferred from this selection might not extend to all acquisition functions. Secondly, the dimensionality of the test functions was selected to be not greater than ten. While Bayesian optimisation is generally considered to work best in this range, and even up to 20 input parameters, it would appear unlikely these results would hold in much higher-dimensional space [23]. It is more likely that optimisation for these more complex problems would require more advanced algorithms. Moreover, the conclusions drawn in this paper are based on the synthetic test functions chosen, and their underlying behaviour. When encountering objective functions with shapes, and challenges, that are not similar to those considered in this paper, the findings might not hold. Lastly, the noise added to the Hartmann function could be too low to represent every possible experiment. It is possible that the measurement error, or other sources of noise, from an experiment or simulation, are too large for Bayesian optimisation to work effectively. More investigation is need to find the maximal noise levels that Bayesian optimisation can tolerate and give good performance.
6 Conclusion

In this paper, Bayesian optimisation algorithms implemented with different types of acquisition function, were benchmarked on synthetic test functions inspired by applications in CFD and in the selection of hyperparameters for statistical models. Synthetic test functions have the advantage that their shape and their global optimum are known. This allows the algorithms to be evaluated on (a) how close their best solutions are to the global optimum and (b) how well they perform on specific challenges such as oscillating functions or functions with steep edges. This evaluation can indicate advantages and shortcomings of the individual acquisition functions and can inform researchers on the best approach to choose for their specific problem.

Four sets of experiments were conducted in this paper. First, analytical single-point acquisition functions were compared to each other. Second, the effect of varying the number of initial training data points was investigated. Third, the analytical approach was contrasted with acquisition functions based on Monte Carlo sampling. Fourth, the single-point approach was compared to the multi-point or batched approach.

From these experiments six main conclusion could be drawn: (i) While all acquisition functions performed well on simple test functions, optimistic policies, such as the Upper Confidence Bound, dealt best with challenging problems. (ii) Varying the number of initial training data points did not have a significant effect on the performance of the individual methods. (iii) Improvement-based acquisition functions struggled with flat test functions. (iv) The results showed that Monte Carlo acquisition functions and multi-point acquisition functions present a good alternative to the widely used analytical single-point methods. (v) The multi-point approach is particularly advantageous when the objective function takes a long time to evaluate and allows parallel evaluations. (vi) Bayesian optimisation performs equally well on noisy objective function.

Acknowledgments

The work has been supported by the Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/T020946/1, and the EPSRC Centre for Doctoral Training in Cloud Computing for Big Data under grant number EP/L015358/1.
References

[1] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104:148–175, 1 2016.

[2] Robert B. Gramacy. Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences. Chapman Hall/CRC, Boca Raton, Florida, 2020.

[3] Jia Wu, Xiuyun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. Journal of Electronic Science and Technology, 17(1):26–40, 2019.

[4] Chaitanya Talnikar, Patrick Blonigan, Julien Bodart, and Qi Qi Wang. Parallel optimization for large eddy simulations. arXiv preprint arXiv:1410.8859, 2014.

[5] O. A. Mahfoze, A. Moody, A. Wynn, R. D. Whalley, and S. Laizet. Reducing the skin-friction drag of a turbulent boundary-layer flow with low-amplitude wall-normal blowing within a bayesian optimization framework. Physical Review Fluids, 4, 2019.

[6] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter. Applying bayesian optimization with gaussian process regression to computational fluid dynamics problems. Journal of Computational Physics, 449, 2022.

[7] Yusuke Nabae and Koji Fukagata. Bayesian optimization of traveling wave-like wall deformation for friction drag reduction in turbulent channel flow. Journal of Fluid Science and Technology, 17(1):26–40, 2019.

[8] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25, 2012.

[9] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv preprint arXiv:0912.3995, 12 2009.

[10] Ioannis Andrianakis and Peter G. Challenor. The effect of the nugget on gaussian process emulators of computer models. Computational Statistics and Data Analysis, 56:4215–4228, 2012.

[11] Robert B Gramacy and Herbert KH Lee. Cases for the nugget in modeling computer experiments. Statistics and Computing, 22(3):713–722, 2012.

[12] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. MIT Press, 2006.

[13] Matthew Hoffman, Eric Brochu, Nando De Freitas, et al. Portfolio allocation for bayesian optimization. In UAI, pages 327–336, 2011.

[14] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In International Conference on Machine Learning, pages 3627–3635. PMLR, 2017.

[15] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity bayesian optimization with max-value entropy search and its parallelization. In International Conference on Machine Learning, pages 9334–9345. PMLR, 2020.

[16] James T Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions. arXiv preprint arXiv:1712.00424, 2017.

[17] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 21524–21538. Curran Associates, Inc., 2020.

[18] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In Computational intelligence in expensive optimization problems, pages 131–162. Springer, 2010.

[19] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. Journal of Machine Learning Research, 15:4053–4103, 2014.

[20] N Ebrahimzade, J. Portoles, M. Wilkes, P. Cumpson, and R.D. Whalley. Optical MEMS sensors for instantaneous wall-shear stress measurements in turbulent boundary-layer flows. In: 12th Symposium on Turbulence and Shear Flow Phenomena (TSFP), pages 1–6, 12 2022.
[21] Bart G.M. Husslage, Gijs Rennen, Edwin R. van Dam, and Dick den Hertog. Space-filling latin hypercube designs for computer experiments. *Optimization and Engineering*, 12:611–630, 12 2011.

[22] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. *Advances in neural information processing systems*, 31, 2018.

[23] Riccardo Moriconi, Marc P. Deisenroth, and K. S. Sesh Kumar. High-dimensional bayesian optimization using low-dimensional feature spaces, 2019.
A Appendix

A.1 Analytical single-point acquisition functions

Figure 7: Performance plots for analytical single-point acquisition functions with five initial training points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Table 7: Best solutions found for analytical single-point acquisition functions with five initial training points per input dimension.

| Method   | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|----------|--------|-------------|----------|----------|--------------------|---------------------|--------------|--------|
| PI       | 1.00   | 1.00        | 1.00     | 0.97     | 0.99               | 1.00                | 0.78         | 0.88   |
| EI       | 1.00   | 1.00        | 1.00     | 0.99     | 0.99               | 1.00                | 0.71         | 0.63   |
| UCB (variable) | 0.98   | 1.00        | 1.00     | 0.98     | 0.99               | 1.00                | 0.80         | 0.96   |
| UCB (β=5) | 1.00   | 1.00        | 1.00     | 0.99     | 1.00               | 1.00                | 0.85         | 0.99   |
| UCB (β=1) | 1.00   | 1.00        | 1.00     | 0.97     | 1.00               | 1.00                | 0.89         | 0.97   |
| Hedge    | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 1.00                | 0.85         | 0.91   |

Table 8: Averaged area under the curve with standard error for analytical single-point acquisition functions with five initial training points per input dimension.

| Method   | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|----------|--------|-------------|----------|----------|--------------------|---------------------|--------------|--------|
| PI       | 0.99   | (± 0.00)    | 0.99     | 0.91     | 0.94               | 0.95                | 0.73         | 0.76   |
| EI       | 1.00   | (± 0.00)    | (± 0.00) | (± 0.04) | (± 0.02)           | (± 0.02)            | (± 0.10)     | (± 0.08)|
| UCB (variable) | 0.98   | (± 0.01)    | (± 0.01) | (± 0.03) | (± 0.02)           | (± 0.02)            | (± 0.06)     | (± 0.01)|
| UCB (β=5) | 1.00   | (± 0.00)    | (± 0.00) | (± 0.02) | (± 0.02)           | (± 0.02)            | (± 0.08)     | (± 0.02)|
| UCB (β=1) | 1.00   | (± 0.00)    | (± 0.00) | (± 0.04) | (± 0.03)           | (± 0.02)            | (± 0.08)     | (± 0.05)|
| Hedge    | 1.00   | (± 0.00)    | (± 0.00) | (± 0.03) | (± 0.03)           | (± 0.02)            | (± 0.08)     | (± 0.11)|
A.2 Varying number of initial training points

Figure 8: Performance plots for analytical single-point acquisition functions with one initial training point per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Table 9: Best solutions found for analytical single-point acquisition functions with one initial training point per input dimension.

| Method | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|--------|--------|-------------|----------|----------|--------------------|--------------------|-------------|--------|
| PI     | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 1.00               | 0.78        | 0.88   |
| EI     | 1.00   | 1.00        | 1.00     | 0.99     | 0.99               | 0.99               | 0.66        | 0.63   |
| UCB (variable) | 0.98 | 1.00        | 1.00     | 0.99     | 1.00               | 1.00               | 0.79        | 0.96   |
| UCB (β=5) | 1.00 | 1.00        | 1.00     | 0.99     | 1.00               | 1.00               | 0.81        | 0.99   |
| UCB (β=1) | 1.00 | 1.00        | 1.00     | 0.97     | 0.98               | 0.99               | 0.89        | 0.96   |
| Hedge  | 1.00   | 1.00        | 1.00     | 0.96     | 1.00               | 0.99               | 0.84        | 0.94   |

Table 10: Averaged area under the curve with standard error for analytical single-point acquisition functions with one initial training point per input dimension.

| Method | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|--------|--------|-------------|----------|----------|--------------------|--------------------|-------------|--------|
| PI     | 0.97   | 0.99        | 0.97     | 0.82     | 0.87               | 0.90               | 0.68        | 0.73   |
| EI     | (± 0.01) | (± 0.00)   | (± 0.01) | (± 0.09) | (± 0.07)           | (± 0.06)           | (± 0.07)    | (± 0.11) |
| UCB (variable) | 0.99 | 1.00        | 0.99     | 0.95     | 0.94               | 0.95               | 0.62        | 0.58   |
| UCB (β=5) | (± 0.00) | (± 0.00)   | (± 0.00) | (± 0.03) | (± 0.04)           | (± 0.03)           | (± 0.09)    | (± 0.13) |
| UCB (β=1) | 0.96 | 1.00        | 0.97     | 0.91     | 0.90               | 0.91               | 0.66        | 0.82   |
| Hedge  | (± 0.01) | (± 0.00)   | (± 0.01) | (± 0.04) | (± 0.05)           | (± 0.03)           | (± 0.06)    | (± 0.01) |
|        | 0.98   | 1.00        | 0.98     | 0.94     | 0.94               | 0.95               | 0.72        | 0.87   |
|        | (± 0.00) | (± 0.00)   | (± 0.00) | (± 0.03) | (± 0.04)           | (± 0.03)           | (± 0.05)    | (± 0.02) |
|        | 0.99   | 1.00        | 0.99     | 0.93     | 0.92               | 0.93               | 0.77        | 0.85   |
|        | (± 0.00) | (± 0.00)   | (± 0.00) | (± 0.08) | (± 0.10)           | (± 0.09)           | (± 0.07)    | (± 0.06) |
|        | 0.98   | 0.99        | 0.98     | 0.89     | 0.91               | 0.92               | 0.73        | 0.76   |
|        | (± 0.00) | (± 0.00)   | (± 0.01) | (± 0.11) | (± 0.05)           | (± 0.05)           | (± 0.08)    | (± 0.09) |
Figure 9: Performance plots for analytical single-point acquisition functions with ten initial training points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Table 11: Best solutions found for analytical single-point acquisition functions with ten initial training points per input dimension.

| Method     | Sphere | Dixon-Price | Griewank | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|------------|--------|-------------|----------|-------------------|--------------------|-------------|--------|
| PI         | 1.00   | 1.00        | 1.00     | 0.97              | 0.99               | 0.99        | 0.73   | 0.51   |
| EI         | 1.00   | 1.00        | 1.00     | 0.98              | 0.99               | 0.99        | 0.68   | 0.53   |
| UCB (variable) | 0.99  | 1.00        | 1.00     | 0.98              | 0.98               | 0.99        | 0.77   | 0.96   |
| UCB (β=5)  | 1.00   | 1.00        | 1.00     | 0.98              | 0.99               | 1.00        | 0.81   | 0.98   |
| UCB (β=1)  | 1.00   | 1.00        | 1.00     | 0.98              | 0.99               | 1.00        | 0.88   | 0.95   |
| Hedge      | 1.00   | 1.00        | 1.00     | 0.98              | 0.99               | 1.00        | 0.79   | 0.87   |

Table 12: Averaged area under the curve with standard error for analytical single-point acquisition functions with ten initial training points per input dimension.

| Method     | Sphere | Dixon-Price | Griewank | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|------------|--------|-------------|----------|-------------------|--------------------|-------------|--------|
| PI         | 1.00   | 1.00        | 1.00     | 0.93              | 0.96               | 0.96        | 0.69   | 0.43   |
| EI         | 1.00   | 1.00        | 1.00     | 0.97              | 0.97               | 0.97        | 0.64   | 0.41   |
| UCB (variable) | 0.99  | 1.00        | 1.00     | 0.96              | 0.96               | 0.96        | 0.66   | 0.87   |
| UCB (β=5)  | 1.00   | 1.00        | 1.00     | 0.97              | 0.97               | 0.97        | 0.74   | 0.91   |
| UCB (β=1)  | 1.00   | 1.00        | 1.00     | 0.96              | 0.98               | 0.98        | 0.74   | 0.86   |
| Hedge      | 1.00   | 1.00        | 1.00     | 0.96              | 0.97               | 0.97        | 0.71   | 0.78   |
A.3 Monte Carlo single-point acquisition functions

Figure 10: Performance plots for Monte Carlo single-point acquisition functions with five initial training points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Table 13: Best solutions found for Monte Carlo single-point acquisition functions with five initial training points per input dimension.

| Method | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|--------|--------|-------------|----------|----------|--------------------|--------------------|--------------|--------|
| MC PI  | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 0.99               | 0.81         | 0.20   |
| MC EI  | 1.00   | 1.00        | 1.00     | 0.99     | 0.99               | 0.99               | 0.68         | 0.07   |
| MC UCB (variable) | 0.98 | 1.00 | 1.00 | 0.98     | 0.99               | 0.99               | 0.77         | 0.96   |
| MC UCB ($\beta=5$) | 1.00 | 1.00 | 1.00 | 0.99     | 1.00               | 1.00               | 0.84         | 0.99   |
| MC UCB ($\beta=1$) | 1.00 | 1.00 | 1.00 | 0.97     | 1.00               | 1.00               | 0.86         | 0.97   |
| MES    | 1.00   | 1.00        | 1.00     | 0.99     | 0.99               | 1.00               | 0.83         | 0.55   |

Table 14: Averaged area under the curve with standard error for Monte Carlo single-point acquisition functions with five initial training points per input dimension.

| Method | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|--------|--------|-------------|----------|----------|--------------------|--------------------|--------------|--------|
| MC PI  | 0.99   | 1.00        | 0.99     | 0.92     | 0.94               | 0.95               | 0.73         | 0.17   |
|        | (± 0.00) | (± 0.00) | (± 0.00) | (± 0.06) | (± 0.02)           | (± 0.04)           | (± 0.09)     | (± 0.12) |
| MC EI  | 1.00   | 1.00        | 0.97     | 0.95     | 0.97               | 0.97               | 0.64         | 0.05   |
|        | (± 0.00) | (± 0.00) | (± 0.02) | (± 0.02) | (± 0.02)           | (± 0.04)           | (± 0.08)     | (± 0.03) |
| MC UCB (variable) | 0.98 | 1.00 | 0.98 | 0.95     | 0.95               | 0.95               | 0.67         | 0.86   |
|        | (± 0.01) | (± 0.00) | (± 0.01) | (± 0.02) | (± 0.02)           | (± 0.04)           | (± 0.06)     | (± 0.01) |
| MC UCB ($\beta=5$) | 1.00 | 1.00 | 1.00 | 0.97     | 0.98               | 0.98               | 0.75         | 0.91   |
|        | (± 0.00) | (± 0.00) | (± 0.00) | (± 0.02) | (± 0.02)           | (± 0.03)           | (± 0.06)     | (± 0.01) |
| MC UCB ($\beta=1$) | 1.00 | 1.00 | 1.00 | 0.96     | 0.97               | 0.97               | 0.77         | 0.89   |
|        | (± 0.00) | (± 0.00) | (± 0.00) | (± 0.07) | (± 0.03)           | (± 0.07)           | (± 0.08)     | (± 0.03) |
| MES    | 0.99   | 0.99        | 0.99     | 0.97     | 0.97               | 0.97               | 0.74         | 0.49   |
|        | (± 0.00) | (± 0.00) | (± 0.02) | (± 0.02) | (± 0.03)           | (± 0.05)           | (± 0.05)     | (± 0.12) |
A.4 Multi-point acquisition functions

Figure 11: Performance plots for improvement-based multi-point acquisition functions with five initial training points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Figure 12: Performance plots for optimistic multi-point acquisition functions with five initial training points per input dimension. Solid lines represent the mean over the 50 runs while the shaded area represents the 95% confidence intervals.
Table 15: Best solutions found for multi-point acquisition functions with five initial training points per input dimension.

| Type            | Method | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|-----------------|--------|--------|-------------|----------|----------|--------------------|--------------------|--------------|--------|
| Sequential Monte Carlo | PI     | 1.00   | 1.00        | 1.00     | 0.98     | 0.98               | 0.99               | 0.74         | 0.24   |
|                 | EI     | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 0.99               | 0.69         | 0.41   |
|                 | UCB (variable) | 0.98 | 1.00        | 0.99     | 0.98     | 0.98               | 0.99               | 0.74         | 0.97   |
|                 | UCB (β=5) | 1.00 | 1.00        | 1.00     | 0.98     | 0.99               | 1.00               | 0.83         | 0.98   |
|                 | UCB (β=1) | 1.00 | 1.00        | 1.00     | 0.98     | 0.99               | 1.00               | 0.87         | 0.97   |
| Joint Monte Carlo | PI     | 1.00   | 1.00        | 1.00     | 0.97     | 0.98               | 0.98               | 0.73         | 0.12   |
|                 | EI     | 1.00   | 1.00        | 1.00     | 0.98     | 0.98               | 0.99               | 0.67         | 0.08   |
|                 | UCB (variable) | 0.96 | 0.99        | 0.99     | 0.97     | 0.97               | 0.98               | 0.65         | 0.95   |
|                 | UCB (β=5) | 1.00 | 1.00        | 1.00     | 0.98     | 0.99               | 1.00               | 0.84         | 0.98   |
|                 | UCB (β=1) | 1.00 | 1.00        | 1.00     | 0.99     | 0.99               | 1.00               | 0.83         | 0.97   |
| Analytical      | CL min | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 0.99               | 0.66         | 0.50   |
|                 | CL max | 1.00   | 1.00        | 1.00     | 0.98     | 0.99               | 0.99               | 0.82         | 0.74   |
|                 | BUCB (variable) | 0.98 | 1.00        | 0.98     | 0.97     | 0.97               | 0.97               | 0.69         | 0.91   |
|                 | BUCB (β=5) | 0.99 | 1.00        | 0.99     | 0.98     | 0.99               | 1.00               | 0.85         | 0.98   |
|                 | BUCB (β=1) | 1.00 | 1.00        | 1.00     | 0.97     | 0.98               | 0.99               | 0.91         | 0.99   |
Table 16: Averaged area under the curve with standard error for multi-point acquisition functions with five initial training points per input dimension.

| Type                     | Method       | Sphere | Dixon-Price | Griewank | Hartmann | Hartmann low noise | Hartmann high noise | Michalewicz | Ackley |
|--------------------------|--------------|--------|-------------|----------|----------|-------------------|---------------------|--------------|--------|
| Joint Monte Carlo        | PI           | 0.99   | (± 0.01)    | 1.00     | (± 0.00) | 0.99              | 0.92                | 0.93         | 0.94   |
|                          | (β=5)        | 0.99   | (± 0.00)    | 1.00     | (± 0.00) | 0.99              | 0.95                | 0.96         | 0.97   |
|                          | (β=1)        | 1.00   | (± 0.00)    | 1.00     | (± 0.00) | 0.99              | 0.96                | 0.96         | 0.96   |
| Sequential Monte Carlo   | PI           | 0.98   | (± 0.01)    | 1.00     | (± 0.00) | 0.99              | 0.86                | 0.89         | 0.89   |
|                          | (β=5)        | 0.99   | (± 0.00)    | 1.00     | (± 0.00) | 0.99              | 0.95                | 0.96         | 0.96   |
|                          | (β=1)        | 1.00   | (± 0.00)    | 1.00     | (± 0.00) | 0.96              | 0.97                | 0.97         | 0.77   |
| CL min                   |              | 0.99   | (± 0.00)    | 1.00     | (± 0.00) | 0.99              | 0.94                | 0.95         | 0.89   |
|                          | (± 0.00)     | 1.00   | (± 0.00)    | 1.00     | (± 0.00) | 0.95              | 0.96                | 0.96         | 0.74   |
|                          | (β=5)        | 0.98   | (± 0.01)    | 1.00     | (± 0.00) | 0.98              | 0.85                | 0.84         | 0.85   |
|                          | (β=1)        | 0.99   | (± 0.01)    | 1.00     | (± 0.00) | 0.99              | 0.88                | 0.89         | 0.90   |
| CL max                   |              | 1.00   | (± 0.00)    | 1.00     | (± 0.00) | 0.95              | 0.96                | 0.96         | 0.74   |
|                          | (± 0.00)     | 1.00   | (± 0.00)    | 1.00     | (± 0.00) | 0.95              | 0.96                | 0.96         | 0.67   |
|                          | (β=5)        | 0.98   | (± 0.01)    | 0.98     | (± 0.00) | 0.88              | 0.84                | 0.85         | 0.61   |
|                          | (β=1)        | 0.99   | (± 0.01)    | 0.99     | (± 0.00) | 0.88              | 0.89                | 0.90         | 0.74   |
| BUCB                     |              | 0.98   | (± 0.01)    | 1.00     | (± 0.00) | 0.99              | 0.94                | 0.94         | 0.69   |
| (variable)               | (β=5)        | 0.98   | (± 0.01)    | 1.00     | (± 0.00) | 0.99              | 0.94                | 0.94         | 0.64   |
|                          | (β=1)        | 0.99   | (± 0.01)    | 0.99     | (± 0.00) | 0.99              | 0.94                | 0.94         | 0.84   |

31