An Empirical Analysis of Entropy Search in Batch Bayesian Optimisation

A Comprehensive Study of Function Shape, Batch Size, Noise Level, and Dimensionality Impact on Information-Theoretic Methods

Petre-Alexandru Hautelman [1]

Supervisors: Matthijs Spaan[1], **Joery de Vries**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

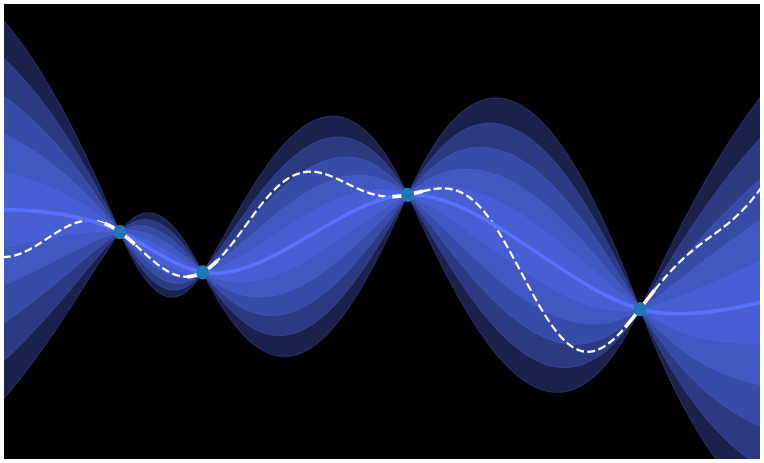

Figure 1: Representation of the Gaussian Process model of a black box function. Source https://github.com/Duane321/mutual_information

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Petre-Alexandru Hautelman
Final project course: CSE3000 Research Project
Thesis committee: Matthijs Spaan, Joery de Vries, Christoph Lofi

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Bayesian optimisation is a rapidly growing area of research that aims to identify the optimum of the black-box function, as it strategically directs the optimisation process towards promising regions. This paper provides an overview of the theoretical background used by the Entropy Search algorithms under study, mainly Predictive Entropy Search, Max-Value Entropy Search, and Joint Entropy Search. Furthermore, we empirically analyse the performance and sensitivity of the algorithms in different environment settings. In particular, we discuss the impact of function shape, batch size, noise level, and the number of input dimensions on the final simple regret metric. The results show the weak spots of the information-theoretic methods. However, the algorithms perform better for batch optimisation, demonstrating the advantage when considering the information on the maximum function value.

1 Introduction

Black-box optimisation is a strategy that does not use derivative information to find the optimal point of a function of many dimensions that is often expensive to evaluate (in terms of time or resources). A common solution to this challenge is to use Bayesian optimisation (BO). This technique involves building a probabilistic model of the black-box function, followed by applying an acquisition function. The acquisition function helps to determine which point (or group of points) to sample next in such a way as to find the location of the optimal point of the function, $x^* = \arg\max_{x \in \mathcal{X}} f(x)$, where $\mathcal{X}$ is the function domain. One metric that quantifies the success of BO is simple regret $\tau_T = f(x^*) - f(x_T)$. Here, x_T is the best estimate, after T timestamps, of the location of the optimal point.

Traditionally, the points for the evaluation of the functions are chosen sequentially, which can lead to a slow optimisation process, especially if the evaluation of the functions is time-consuming. On the contrary, batch point selection, also known as parallel optimisation, allows the simultaneous selection of multiple points. This concept is particularly relevant to our study as it can significantly speed up the optimisation process with respect to wall-clock time. Furthermore, it is an attractive choice for real-world applications where the function can be evaluated simultaneously in different systems, such as optimising machine learning model hyperparameters, drug development, or robotics [2]. Although there have been recent efforts to extend classical acquisition functions, such as Probability of Improvement, Expected Improvement, or Upper Confidence Bound, to the parallel setting, they often suffer from a myopic behaviour, where the algorithm does not consider the long-term effects of the selected batch of points. [16; 7]

Entropy Search (ES) is a class of algorithms that describes a selection policy that is non-myopic by design. ES is based on mathematical concepts introduced by Claude Shannon: information gain and information entropy [15]. With each new selection, ES aims to increase the information gained about the position of the global objective maximise. The existing literature has contributed significantly to expanding the ES approach by developing different efficient approximations and providing robust implementations for the ES algorithm, such as Predictive Entropy Search (PES) [8], Parallel Predictive Entropy Search (PPES) [14], Max-value Entropy Search (MES) [21], and Joint Entropy Search (JES) [18; 9].

Although there have been recent advances in ES algorithms, it is important to address existing knowledge gaps. One such issue is the need for a systematic study on the performance of (P)PES, MES, and JES in parallel optimisation across various environment settings, such as different batch sizes, function dimensions, and types of objective functions. Additionally, understanding how these algorithms work in different situations can improve their performance in real-world scenarios, leading to better results and increased efficiency.

The central research question of the paper is: *How do Entropy Search algorithms perform under various environment specifications, and what are the factors influencing their performance?*' with a particular focus on the value of the regret function and compute time, as well as a comprehensive comparison in the efficiency of the three algorithms.

This study underscores the superiority of entropy search-based acquisition functions in batch Bayesian optimisation compared to the greedy function qEI, particularly when dealing with unimodal functions. Despite their robustness, the adverse effects of noise in high-dimensional spaces require effective countermeasures. These findings map the capabilities and challenges of these methods, deepening our understanding of their operation.

2 Background Information

In this section, we elaborate on the mathematical underpinnings of our research and provide an overview of the theoretical background used by Bayesian optimisation and the methods employed in the field. We also provide an outline of the ES algorithms under study, mainly PES, MES, and JES, and discuss the specifics of their application in the batch specification.

2.1 Bayesian Optimisation and Acquisition Functions

The surrogate model, typically a Gaussian process (GP), is at the heart of Bayesian optimisation [13]. Given a dataset $D_n = \{(X_n, Y_n)\} = \{(x_i, y_i)_{i=1}^n\}$, where $y_i = f(x_i) + \epsilon_i$ are the noisy scalar evaluations of the black box function $f : \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^d$, and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ i.i.d., a GP constructs a probabilistic model of the true objective function, assuming that the observed data are generated by an unknown function that is drawn from the GP prior. The conditional probability of a new function value, given the observed data, is then defined as $p(y|D_n) = \mathcal{N}(y|D_n, \beta)$, where β is a set of hyperparameters. As such, we can derive the posterior function model $\tilde{f} : \mathcal{X} \to \mathcal{N}(\mu(\mathcal{X}), \sigma^2(\mathcal{X}))$, which out-

puts the mean μ and the confidence σ in the predicted value for each point in our domain. The GP utilises kernel functions, such as the squared exponential kernel or the Matern kernel, to model the correlations between input points. The kernel contains hyperparameters that significantly affect the properties of the resulting function. For example, the length scale influences the smoothness of the function, and the signal variance controls the overall variability of the function values. These hyperparameters are typically inferred from the data by maximising the logarithmic marginal likelihood, a measure of the probability of the observed data given the posterior prediction of the GP, $\mathcal{L} = \log p(Y_n|X_n, \beta) = \log \int p(Y_n|\tilde{f})p(\tilde{f}|X_n, \beta)d\tilde{f}$.

In Bayesian optimisation, acquisition functions are instrumental in selecting subsequent query points. They are heuristics that leverage the probabilistic model built by the GP to measure the improvement gained by evaluating a (set of) points. Classic examples of acquisition functions are Probability of Improvement (PI) (1), Expected Improvement (EI) (2), and Upper Confidence Bound (UCB) (3) [20]. These functions aim to achieve an optimal balance between exploration and exploitation: exploration involves probing areas with high uncertainty, while exploitation involves selecting points that yield high expected objective value. Consequently, acquisition functions are indispensable tools for locating the optimum of the black-box function, as they strategically direct the Bayesian optimisation process towards promising regions.

$$\alpha_{PI}(x) = P(f(x) \geq f^*) = \quad (1)$$
$$= \Phi\left(\frac{\mu(x) - f^*}{\sigma(x)}\right)$$

$$\alpha_{EI}(x) = \mathbb{E}[\max(0, f^* - f(x))] = \quad (2)$$
$$= (f^* - \mu(x))\Phi\left(\frac{f^* - \mu(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f^* - \mu(x)}{\sigma(x)}\right)$$

$$\alpha_{UCB}(x) = \mu(x) + \beta\sigma(x), \quad (3)$$
where $\beta > 0$ is a parameter that controls the exploitation-exploration trade-off

Figure 2: Acquistion functions for Probability of Improvement (PI), Expected Improvement (EI), and Upper Confidence Bound (UCB). Here, f^* denotes the optimum value obtained so far, $\Phi(\cdot)$ is the normal cumulative distribution function (CDF), and $\phi(\cdot)$ the probability density function (PDF).

Although traditional acquisition functions such as PI, EI, and UCB have been shown to be effective in sequential point selection, they face specific issues when used in batch selection. Extending these functions directly to batch selection complicates the process and can result in less effective results. One solution proposed to address these difficulties is to use the Markov chain Monte Carlo (MCMC) method to approximate the estimation of the acquisition function to sequentially populate the selection set [16]:

$$\alpha_{PI-MCMC}(x|\{x_{q'}\}_{q'=1}^{q}) =$$
$$= \int_{\mathcal{X}^q} [\alpha_{PI}(x|D_n \cup \{x_{q'}, y_{q'}\}_{q'=1}^{q})$$
$$p(\{y_{q'}\}_{q'=1}^{q}|D_n, \{x_{q'}\}_{q'=1}^{q})dy_1..dy_q]$$

is the *expected* gain in evaluating x after evaluating $\{x_{q'}, y_{q'}\}_{q'=1}^{q}$. However, this approach leads to a problem known as the "double greedy" selection process [14]. This occurs because acquisition functions are inherently greedy when choosing a single candidate that is expected to give the most utility when considering the simple next-step regret. Furthermore, the MCMC approach greedily populates the batch set S_t one by one using the output of the acquisition function. This parallel optimisation algorithm often results in a set of similar or identical points, reducing the effectiveness of the evaluation. Therefore, while MCMC may help overcome some issues of batch selection, it is not the most optimal solution and introduces its challenges.

2.2 Entropy Search and Information-Theoretic Acquisition Functions

Information-Theoretic acquisition functions aim to maximise the gain in information about the global maximiser from the next observation. By focusing on information gain instead of immediate improvement, it encourages the selection of a diverse set of points, effectively mitigating the "double-greedy" issue in batch selection.

The authors of [9; 11] propose an efficient, albeit suboptimal, batch selection procedure for information-theoretic acquisition functions that does not introduce any conditional terms between the selected sampling points. This extension is achieved simply by summing the acquisition values of each point within the batch:

$$\alpha(\{x_q\}_{q=1}^{Q}) = \sum_{q=1}^{Q} \alpha(x_q)$$

where the previously selected points are 'fantasized' to be part of the training data. Remarkably, the summed quantity provides an upper-bound approximation of the true acquisition function, thus enhancing the versatility of information-theoretic methods in Bayesian optimisation. As such, we will only consider the case of a single input when referring to the acquisition functions of information-theoretic algorithms. Nevertheless, it is important to remember that the function can easily be extended to a batch setting.

One popular approach is Entropy Search (ES), which calculates the expected gain in information about the location of the global maximiser x^* when querying a new point x. The information gain is measured in terms of the reduction in the differential entropy $H[\cdot]$ of the distribution over the global maximizer:

$$\alpha_{ES}(x) = H[p(x^*|D_n)] -$$
$$\mathbb{E}_{p(y|D_n,x)}[H\left(p(x^*|D_n \cup \{x, y\})\right)] \quad (4)$$

While this formula is conceptually straightforward, it is computationally intractable due to non-analytic entropy calculations and the requirement to calculate the probability distribution of the maximum given an extended dataset.

2.3 Predictive Entropy Search

Predictive Entropy Search (PES) is an evolution of the ES acquisition function, which is moving towards a more analytically tractable formulation. While ES seeks to directly minimise the entropy of the global optimum's distribution upon a potential observation, PES is designed to compute the change in the entropy of the predictive distribution at the global optimum's location. This adjustment arises from the utilization of mutual information in the derived equation 4:

$$\alpha_{PES}(x) = \\ = H[p(y|D_n, x)] - \mathbb{E}_{p(x^*|D_n,S_t)}[H\left(p(y|D_n, x, x^*)\right)]$$

The PES formulation gives rise to two easier-to-calculate terms given the statistical model derived from the GP. As such, the first term, which describes the entropy of the predictive distribution, can be calculated analytically using the model inference variance $H[p(y|D_n, x)] = 0.5\log(2\pi e(\sigma(x) + \sigma_\epsilon^2))$, where σ_ϵ is the variance of observation noise.

However, the second term poses more challenges due to its reliance on calculating the entropy over the GP's posterior, given the expected location of the global maxima: $p(x*|D) = p(f(x^*) = \max_{x\in\mathcal{X}} f(x)|D_n)$. PES resorts to a series of approximations that effectively reduce this second term to a Gaussian model to manage its complexity.

The first step makes use of Brochner's theorem to build an estimate of the GP kernel that is based on its Fourier dual [4]. This allows efficient sampling of a function path $f_s \sim p(\tilde{f}|D_n)$ from the GP model. Let $x_s^* = \arg\max_{x\in\mathcal{X}} f_s(x)$ be the global maximum argument of the sampled function. Then, expectation propagation is used to construct a Gaussian posterior distribution that is conditioned on the new maximum location, as well as the belief that $f_s(x_s^*)$ is the optimum function value [10]. As such, the expected entropy can be calculated using the predictive variance that results from the approximation, $\sigma(x|x_s^*)$.

In the original description of the algorithm given in [8], the authors marginalise over a set of M GP hyperparameters to obtain a Bayesian treatment of the global maxima location expectation, as well as to account for uncertainty in the model parameters. Let β denote a vector of hyperparameters that includes any kernel parameters, as well as the noise variance σ^2, then, the Monte Carlo method of slice sampling is used to draw M samples $\{\beta^{(i)}\}_{i=1}^M$ from $p(\beta|D_n)$ [19].

The final PES acquisition function becomes:

$$\alpha_{PES}(x) = \\ = \frac{1}{2M}\sum_{i=1}^{M}\left(\log(\sigma^{(i)}(x) + \sigma_\epsilon^{2(i)}) - \log(\sigma^{(i)}(x|x^{*(i)}) + \sigma_\epsilon^{2(i)}\right)$$

The PES approximation is feasible since most of the terms do not depend on the input and can be precomputed. Furthermore, it is suitable for gradient search optimisation, enabling for such methods to identify the point x that maximises α_{PES}.

2.4 Max-value Entropy Search

Max-Value Entropy Search (MES) is a robust and efficient improvement in ES, particularly with regard to the implementation of PES [21]. The MES algorithm works by considering information regarding the maximum function *value*, $y^* = f(x^*)$, rather than approximating the distribution of the optimum position. The former variable resides in a one-dimensional space, thus greatly simplifying the calculations compared to the multidimensional scenario of PES.

This simplification also provides the basis for the MES acquisition function:

$$\alpha_{MES}(x) = \\ = H[p(y|D_n, x)] - \mathbb{E}_{p(y^*|D_n)}[H(p(y|x, D_n, y^*))]$$

The first term is shared with the PES formula and consists of the entropy of the predicted Gaussian distribution. The second term is the entropy of a truncated Gaussian distribution, taken over the expected distribution of the maximum function value, based on the assumption that y^* is the global maximum of the function.

At first glance, it may seem that an information increase about the function's maximum value does not also help in finding its position. However, the formula favours points that reduce the tail probability of the maximum function value $p(y^*|D_n)$. By strategically selecting points expected to significantly reduce uncertainty about the highest value, the algorithm successfully identifies the location of the optimum more efficiently than its counterparts.

MES begins with a fitted GP model following a similar implementation path as PES. To approximate the second term of the acquisition function, it becomes necessary to sample the maximum values of the functions of K from this model, $Y^* = \{(y_k^*)\}_{k=1}^K$, where $y^* = max_{x\in\mathcal{X}}\tilde{f}(x)$ is the maximum value of a function sampled $\tilde{f}\ GP(\mu, \sigma|D_n)$. The authors of [21] describe two methodologies that can be employed for this process: the Gumbel or Monte Carlo (MC) techniques, each providing distinct advantages. Subsequently, the tractable implementation marginalises these samples to compute the acquisition function.

$$\alpha_{MES}(x) \approx \frac{1}{K}\sum_{y^*\in Y^*}[\frac{\gamma_{y^*}(x)\phi(\gamma_{y^*}(x))}{2\Phi(\gamma_{y^*(x)})} - \log(\Phi(\gamma_{y^*(x))})]$$

where ϕ is the probability density function and Φ the normal cumulative density function, and $\gamma_{y^*} = \frac{y^*-\mu(x)}{\sigma(x)}$. Notably, when using only one max sample, MES becomes equivalent to the Probability of Improvement (PI) strategy. [21; 22]

MES proves to be a more robust approach to PES that is easier to implement and faster. In practise, MES often outperforms PES.

However, despite these benefits, MES has its limitations. Later studies show that it can overestimate the information

gain of points in noisy settings, as it assumes a noiseless environment. This drawback requires careful consideration when applying MES in real-world scenarios [12].

2.5 Joint Entropy Search

Joint Entropy Search (JES) presents a more accurate algorithm in the domain of information-theoretic acquisition functions, combining ideas from PES and MES [18; 9]. JES constructs a joint probability distribution over both the input and output space by incorporating information about the distribution of the maximum position and the maximum function value. As such, JES offers a maximally informed heuristic for determining the next sampling point.

$$\alpha_{JES}(x) =$$
$$= H[p(y|D_n, x)] - \mathbb{E}_{p(x^*, y^*|D_n)}[H(p(y|D_n \cup (x^*, y^*), x, y^*))]$$

JES follows a clear implementation path. First, it generates a set of optimal pairs $\{(x_l^*, y_l^*)\}_{l=1}^L$, where each y_l^* is the maximum point of a function sampled from the GP model. Given the current information about the function, this helps to estimate both the distribution of maximum values and their position. The algorithm marginalises these two distributions by then computing a new conditional posterior distribution of L GP models, each including an optimal pair in its training data $D_n \cup \{x_l^*, y_l^*\}$. However, it is necessary to truncate the new probability that imposes $f(x) <= y^* i$ to maintain the belief that the value is maximal.

$$\alpha_{JES}(x) \approx$$
$$\approx \log(\sigma(x) + \sigma_\epsilon^2) - \frac{1}{L} \sum_{l=1}^{L} \log(\sigma_\epsilon^2 + \sigma_{f|y_l^*}(x; D_n \cup (x_l^*, y_l^*)))$$

The entropy of the estimated term is calculated using the variance $\sigma_{f|y_l^*}(x; D_n \cup (x_l^*, y_l^*)) = \sigma_T(y^*; \mu_l(x), \sigma_l(x))$ where $\sigma_T(\alpha; \mu, \sigma)$ is the variance of the Gaussian distribution truncated at α, and $\mu_l(x)$ and $\sigma_l(x)$ are the mean and covariance of the GP conditioned on the optimal pair (x_l^*, y_l^*).

JES not only discerns the probable sites for the optimal value, but also estimates the likely minimum and maximum bounds of that optimal value, thereby providing valuable information for subsequent enquiries.

3 Methodology

This research investigates the performance of information-theoretic acquisition functions in batch Bayesian optimisation (BO). Specifically, it focuses on three critical aspects: the type of objective functions, the input data dimensions, and the noise levels in the evaluations. Additionally, the study investigates the interaction of these variables with batch size, thereby uncovering performance nuances under different settings.

Batch BO has received significant attention, especially in areas where the parallel evaluation of black-box functions offers a significant advantage in reduced evaluation time. Notable examples include A/B testing and hyperparameter tuning of algorithms. The premise that larger batch sizes could accelerate the optimisation process with respect to the wall clock time is theoretically appealing. However, it often results in compromised performance compared to sequential selection of the same number of points, as quantified by the immediate regret metric (5). This limitation is attributed to the inherent bias in the batch selection procedure, where the points are selected based on the model generated by GP which can be inaccurate. To assess the performance of the algorithms in different parallel settings, the study considers batches of sizes 2, 5, 10, and 25.

This study recognises that the types of problems with which BO deals are inherently complex. They often come with limited training data, no gradient information, and potentially noisy evaluations. GPs are robust tools for modelling these problems, as they can approximate any arbitrary function. However, different types of objective functions could compromise their inference efficacy. Therefore, this study includes a diverse set of objective functions to explore this aspect. The chosen objective functions include Griewank, Zakharov, Easom, Ackley, the sum of different powers, and Schwefel. These functions span various functional landscapes, from unimodal to complex multimodal, representing a broad spectrum of real-world challenges. For instance, Easom poses the problem of a flat outer region with a maximum that occupies a small proportion of the search space. Meanwhile, Griewank presents a complex multimodal landscape with numerous local maxima. Schwefel, in contrast, requires a balance between exploration and exploitation, thereby necessitating a balanced approach from the acquisition function.

Another factor that is present in real-world applications is the presence of noise in the evaluation process. Noise, quantified as a percentage of the range of values of the objective function, can obscure the actual value of the objective function and present significant challenges to the optimisation algorithm. Noise levels ranging from a noiseless setting to 5%, 10%, 20%, and 40% are considered to assess the robustness of the acquisition functions in various noise conditions.

The input data dimension is another important aspect investigated in this study. The 'curse of dimensionality' is a common concern in high-dimensional problems, where an increase in the input dimensions can drastically increase the complexity of the problem. Therefore, this study considers different input dimensions of 2, 5, 10, 25, and 50 to evaluate the effectiveness of acquisition functions and ascertain the feasibility of these methods in high-dimensional settings.

Given the extensive computational requirements of the algorithms, the study adopts a fractional factorial design approach [3]. This method allows selecting a subset of the complete experimental setup while maintaining a comprehensive understanding of the performance implications of the chosen parameters. A complete description of how the technique was applied can be found in A.

The algorithms evaluated in this study include qPredictive-Entropy-Search, qMax-Value-Entropy-Search, and qJoint-Entropy-Search, as they are implemented in the BOTorch Python framework [1]. A simple random agent is also included for baseline comparison, along with the popular qExpected-Improvement method, a classical approach to balance computational efficiency and performance. The testing environment was implemented us-

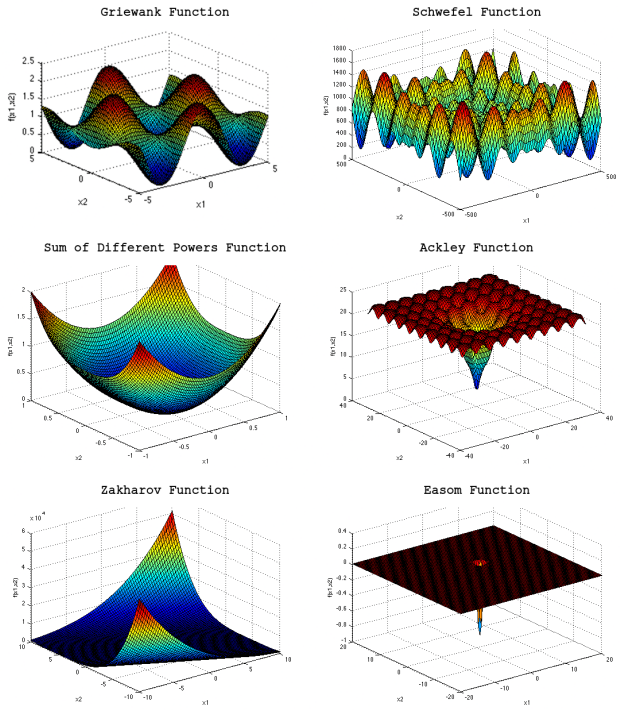

Figure 3: A 2-dimensional representation of the functions used for evaluating the acquisition functions, each presenting its challenges. Notably, Griewank, Schwefel, and Ackley are more difficult to optimize multimodal functions; Sum of Different Powers, Zakharov, and Easom are unimodal. Easom stands out for its small global minimum area relative to its search space. Source [17]

ing the jit_env and BBOx codebases [6; 5]. The agent implementations, testing environment, obtained results, and methods used to analyse the data can be found, in their entirety, at https://github.com/ahautelman/entropy-seach-batch-global-optimiation-performance. Finally, the tests were performed and averaged over five iterations to ensure the reliability and impartiality of the results.

The performance of the algorithms was evaluated using two metrics: simple and cumulative regret (5). Simple regret compares the optimal solution with the best solution discovered. Cumulative regret offers an accumulated disparity between the optimal and chosen points over iterations, measuring the overall quality of decisions. High cumulative regret values can also suggest that the algorithm performs significant exploration; however, as long as the simple regret remains minimal, the algorithm's performance is considered adequate.

$$\tau_t^{\text{cumulative}} = \tau_{t-1}^{\text{cumulative}} + \sum_{S_t}[f(x^*) - f(x)] \tag{5}$$

4 Results

In this section, the study empirically analyses the performance and sensitivity of information-theoretic algorithms in different environment settings. In each case, we discuss the impact of each individual parameter on the final simple regret metric: function shape, batch size (q), noise level, and the number of input dimensions (D); then, remarks are offered as to the interaction between them. Finally, we analyse the runtime behaviour of the algorithms and make concluding remarks about their performance.

It is important to note that the benchmarks for the qPES function were not fully conducted due to the significant computational resources, particularly in terms of RAM and/or GPU, required to execute the algorithm. The intensity of these resource demands posed considerable challenges during the execution of the test cases. The limited memory capacity of the testing machines resulted in frequent failures and compromised the integrity of the obtained results. Consequently, qPES could not be included in most of the benchmarks due to these practical limitations.

Griewank (q=2, D=2, varying noise) [Figure 4] The Griewank benchmark did not pose a difficulty to the performance of any of the algorithms. Even at a high noise level of 40%, the acquisition functions could rapidly converge towards the global maximum within a few iterations. It is noteworthy that despite the inherent complexity of the Griewank function, which is characterised by many widespread local minima, the function exhibits a general unimodal appearance when disregarding the oscillations of the function's values. Consequently, all acquisition functions successfully identified reasonable candidates for this low-dimensional benchmark.

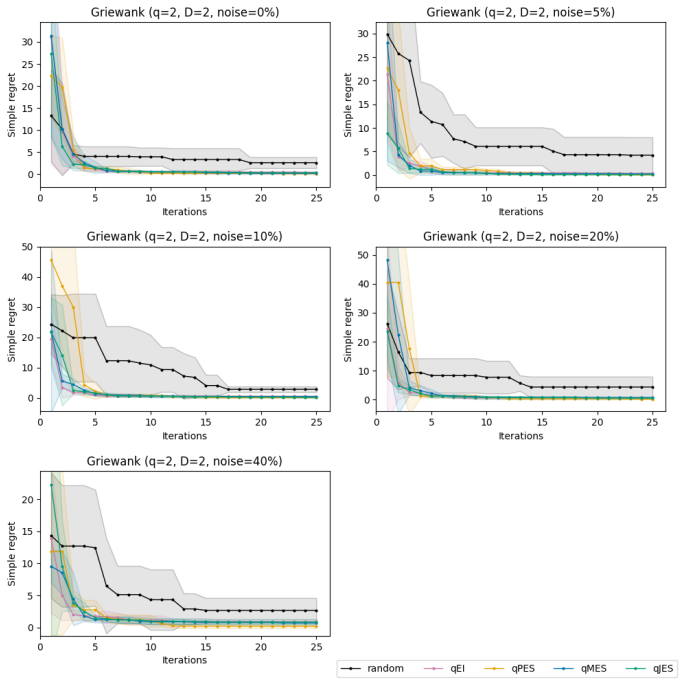

Figure 4: Griewank run

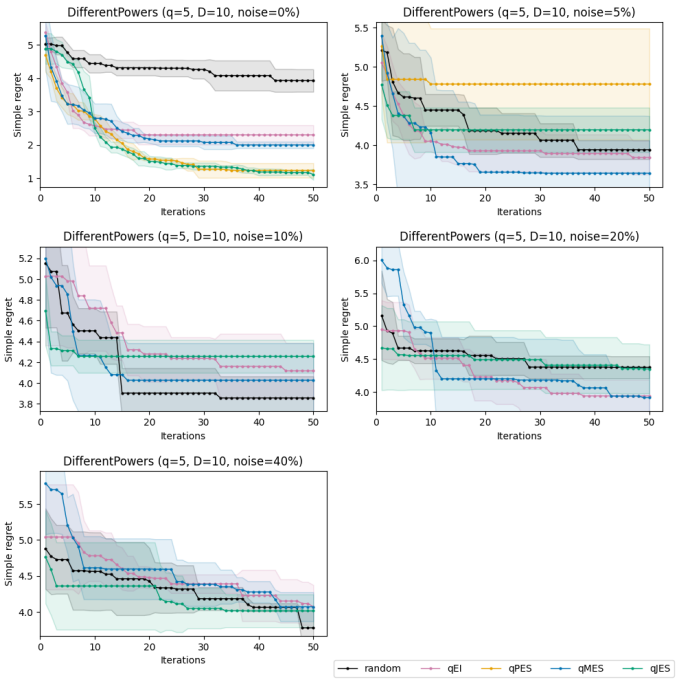

Figure 5: Sum of Different Powers run

Agent	Noise - Simple regret correlation
qEI	0.939
qPES	0.3715
qMES	0.8729
qJES	0.8633

Table 1: Correlation between the four noise levels (0, 5%, 10%, 20%, 40%) and simple regret, after 20 points evaluation, for benchmark Griewank (q=2, D=2)

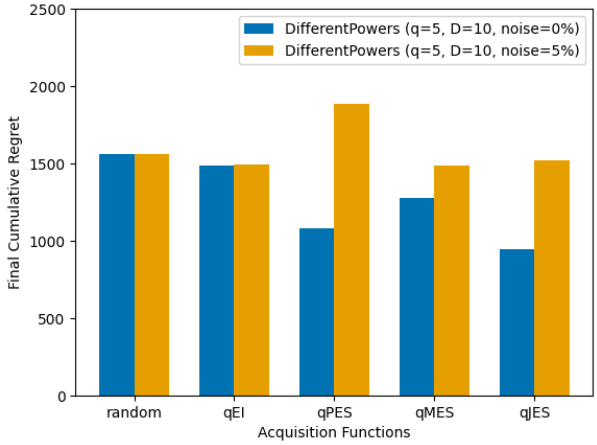

Figure 6: Performance of acquisition functions in high-dimensional, unimodal function as measured by cumulative regret after 50 iterations of 5 batch size. Information-theoretic methods have an overall better performance for noiseless batch optimisation. However, the inference of the optimal value decreases rapidly with the introduction of noise.

Sum of Different Powers (q=5, D=10, varying noise) [Figure 5] The sum of different powers benchmark is a reasonably simple unimodal problem with a steady slope towards the optimal position. This test investigates the effect of noise coupled with a high-dimensional space. Notably, in a noiseless setting, the information-theoretic algorithms demonstrated remarkable effectiveness in approaching the maximum function, even within the high-dimensional problem space. The results presented in Figure 6 highlight the

final cumulative regret, indicating the overall superior quality of the selected batch of points for information-theoretic methods for the run without noise. However, it should be noted that as the noise level increased, the functions encountered difficulties in accurately inferring the position of the optimal value, not even surpassing the performance of the random agent.

Easom (q=2, varying input dimension, noise=5%) [Figure 7] Unfortunately, the Easom benchmark proved difficult for BO, the acquisition functions struggled to find the small optimum area, even in a two-dimensional setting. In particular, MES and JES managed to do so more consistently than the other ES method, showing the advantage of considering the information of the maximum function value.

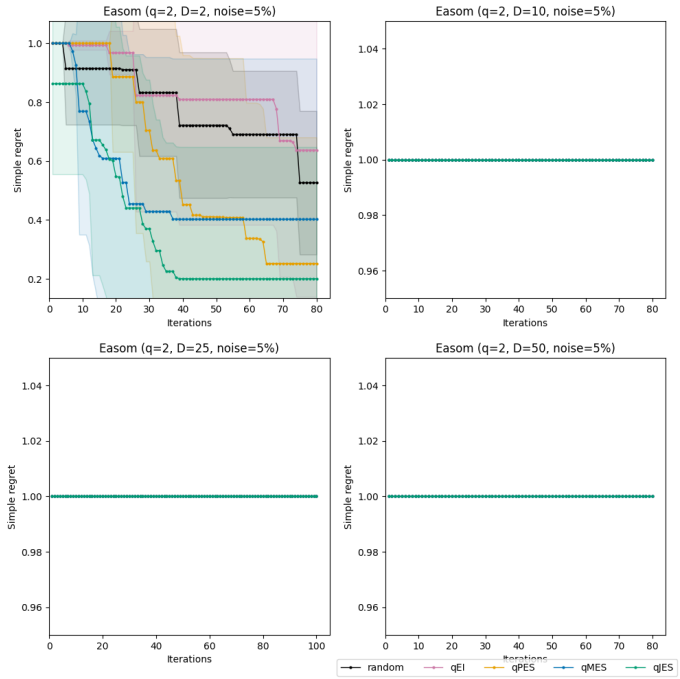

Figure 7: Easom run

Ackley (q=5, varying input dimension, noise=10%) [Figure 8] The evaluation of the Ackley benchmark is tricky due to the presence of numerous evenly distributed local minima. In low-dimensional scenarios, all acquisition functions can approximate the local optima with reasonable accuracy. However, as the problem's dimensionality increases, the acquisition functions' performance deteriorates. Among the acquired functions examined, qJES stands out because it exhibits greater resilience to the expansion of the input space. It consistently approaches a maximum even in 25 dimensions, showcasing its robustness in tackling higher-dimensional settings.

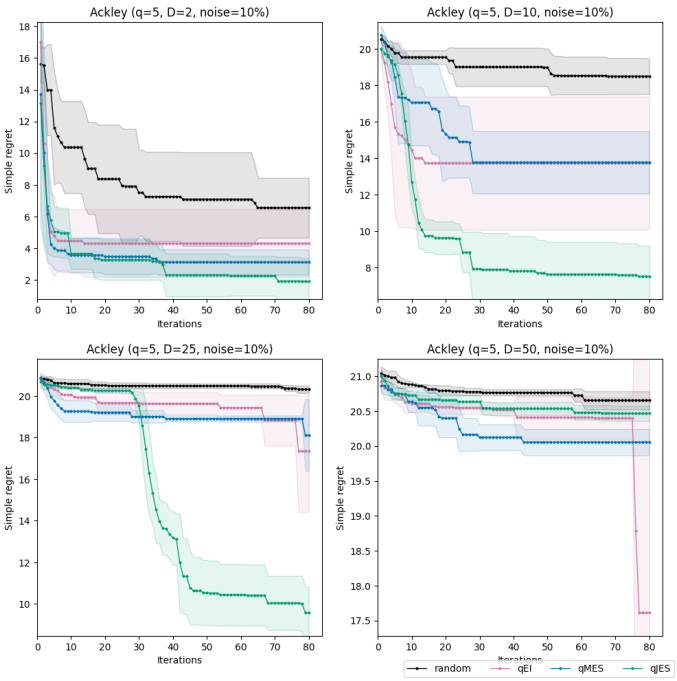

Figure 8: Ackley run

Zakharov (varying batch, D=2, noise=0%) [Figure 9] The low-dimensional, noise-free, unimodal function did not pose significant challenges for any acquisition functions examined. In particular, the batch selection strategy exhibited remarkable effectiveness, facilitating rapid convergence towards the minimum of the function. It is worth noting that a positive correlation emerged between the size of the batch and the magnitude of the simple regret 2. However, such an outcome was expected, and the observed regret increase proved insignificant compared to the output range of the Zakharov function.

Agent	Batch - Simple regret correlation
qEI	0.723
qMES	0.936
qJES	0.929

Table 2: Correlation between the four batch sizes (2, 5, 10, 25) and simple regret, after 100 point evaluations, for benchmark Zakharov (D=2, noise=0%).

Schwefel (varying batch, D=10, noise=20%) [Figure 10] The performance evaluation of the acquisition functions using the Schwefel benchmark further emphasises the detrimental effects of a high noise level, particularly in conjunction with a high-dimensional search space. In this complex multimodal setting, the batch selection strategy proved ineffective. High noise levels significantly contaminated any potentially helpful information for the model, impairing its ability to identify optimal solutions accurately.

In our evaluations, the runtime of the information-theoretic methods was found to be considerably longer than that of

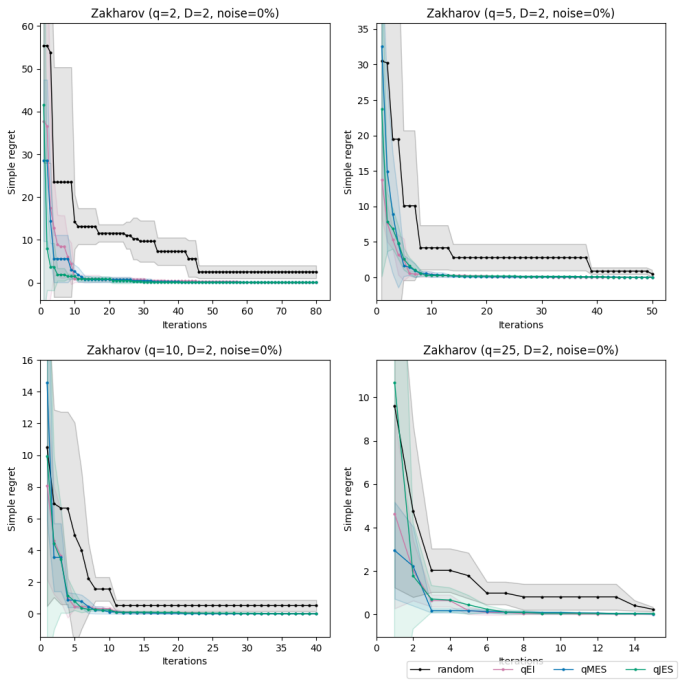

Figure 9: Zakharov run

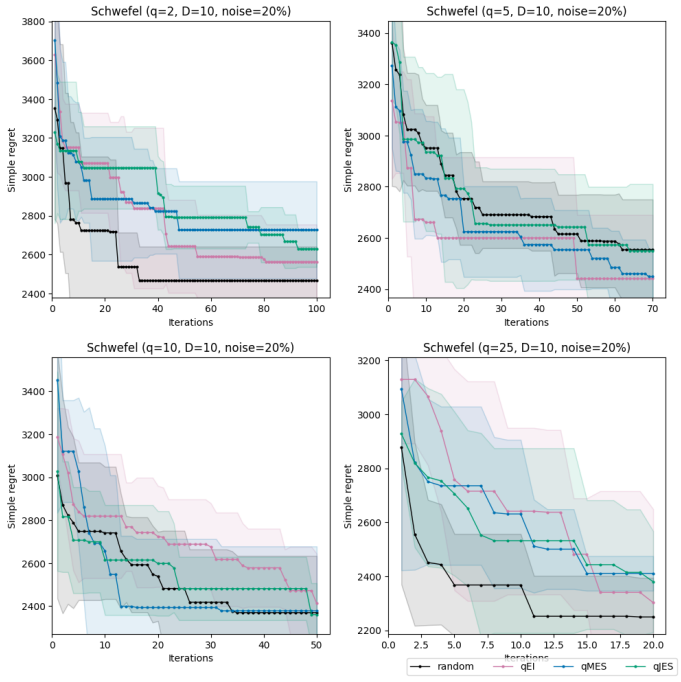

Figure 10: Schwefel run

Agent	Noise - Simple regret correlation
qEI	1.14
qPES	13.62
qMES	3.13
qJES	24.92

Table 3: Average runtime in seconds of the studied algorithms over all runs.

their classical counterparts, with qMES being the lone exception [3]. It is important to consider, however, that the results for the qPES agent, which was tested using GPU acceleration, are not entirely definitive. Generally speaking, PES tends to have a longer runtime compared to the other two acquisition functions [18].

This investigation of entropy search-based acquisition functions in batch Bayesian optimisation has highlighted their effectiveness and robustness across various input dimensions. In particular, qJES and possibly qPES (pending further testing) showed an increased resistance to greater input dimensions, as it results from the Ackley [8] and Sum of Different Powers settings [5]. The information-theoretic methods performed at least as well as, if not better than, qEI in most batch optimisation scenarios. The study has especially highlighted their proficiency in pinpointing optimum functions within batch settings, even under complex circumstances, with significant efficiency when optimising unimodal functions with discernible slopes. However, the results also highlight the detrimental impact of noise, especially within high-dimensional spaces, necessitating the deployment of noise reduction strategies or alternative methods in situations where noise interference is inevitable.

5 Responsible Research

This research was committed to integrity, transparency, and social impact. The primary objective was to ensure reproducibility and facilitate validation, advancement, and innovation by the wider academic community.

In particular, the open-source philosophy underpins the entire methodology. All the code used in the investigation, from data collection to analytical stages, is publicly accessible through the provided GitHub link. Second, we have ensured that all datasets used in our research are available in the public domain. By making these datasets publicly accessible, we allow for further exploration and the opportunity for peer feedback, thereby ensuring continuous improvement in the quality of research in the field.

Our methodological approach to implementing information-theoretic acquisition functions and data analysis is explained in-depth within the paper. This approach facilitated a clear understanding of the analysis steps and the foundations of our conclusions.

Research on the performance of information-theoretic acquisition functions for batch BO offers critical insights for researchers and industry practitioners. This knowledge will empower informed decision making when using information-theoretic algorithms for optimisation, potentially improving

operational efficiency and improving products and services, thus affecting societal progress.

6 Conclusions and Future Work

Investigating entropy search-based acquisition functions within batch Bayesian optimisation has identified vital insights. This research confirms the superiority of these methods compared to the greedy acquisition function, qEI, in terms of efficacy, but also their robustness across various input dimensions. Significantly, their proficiency in identifying the optimum function within batch settings was consistently demonstrated, even under complex conditions.

The potency of entropy search methods is particularly evident when optimising unimodal functions, provided the existence of a discernible slope to navigate the input space effectively. In particular, the research sheds light on the harmful impact of noise on the optimisation process. Its influence is accentuated, particularly within high-dimensional spaces, underscoring the critical need for deploying noise reduction strategies or alternate methods in circumstances where noise interference is inevitable.

In summary, this research successfully maps out the virtues and constraints of entropy search methods within the framework of batch Bayesian optimisation. These invaluable insights significantly enhance our understanding of the dynamics shaping their performance.

In light of this revision, future studies can explore more complex scenarios. Settings with input-dependent noise, misspecified environments, or multiobjective optimisation are a promising direction for future research. Specifically, environments where rewards are delayed pose a prevalent real-world optimisation problem, making them critical areas for further exploration. Studying the performance of these algorithms under such challenging conditions will be instrumental in refining and enhancing their robustness and reliability, making them practical in a broader range of scenarios.

References

[1] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. French National Centre for Scientific Research, 12 2011.

[3] George E. P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters*. Wiley-Interscience, 5 2005.

[4] Salomon Brochner. *Lectures on Fourier Integrals. (AM-42)*. 12 1959.

[5] Joery de Vries. BBOx: Black-box optimization in jax, 2023.

[6] Joery A. de Vries. jit_env: A jax interface for reinforcement learning environments, 2023.

[7] Javier Carrascosa González, Michael A. Osborne, and Neil D. Lawrence. GLASSES: Relieving The Myopia Of Bayesian Optimisation. pages 790–799, 5 2016.

[8] José Hernández-Lobato, Matthew Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in Neural Information Processing Systems*, 1, 06 2014.

[9] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy search for maximally-informed bayesian optimization. 01 2023.

[10] Tom Minka and Rosalind W. Picard. A family of algorithms for approximate bayesian inference, 1 2001.

[11] Henry B. Moss, David M. Leslie, Javier T. Gonzalez, and Paul Rayson. GIBBON: General-purpose Information-Based Bayesian Optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 10 2021.

[12] Quoc Nguyen, Bryan Kian, Hsiang Low, and Patrick Jaillet. Rectified max-value entropy search for bayesian optimization, 2022.

[13] Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. 11 2005.

[14] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. 11 2015.

[15] Claude E. Shannon and Weaver Weaver. The mathematical theory of communication. *The Mathematical Gazette*, 34(310):312–313, 1950.

[16] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. volume 25, pages 2951–2959, 12 2012.

[17] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved June 25, 2023, from http://www.sfu.ca/~ssurjano.

[18] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 35, pages 9922–9938. Curran Associates, Inc., 2022.

[19] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14(1):1175–1179, 1 2013.

[20] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent Advances in Bayesian Optimization. *ACM Computing Surveys*, 1 2023.

[21] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. 03 2017.

[22] Zi Wang, Bolei Zhou, and Stefanie Jegelka. Optimization as Estimation with Gaussian Processes in Bandit Settings. pages 1022–1031, 5 2016.

A Fractional Factorial Design

A full factorial design is one in which all possible combinations of all levels of all factors are tested. However, this can be very resource intensive. To alleviate this, a fractional factorial design approach can be employed, enabling testing of only a select fraction of all conceivable combinations.

A prevalent form of fractional factorial design can be represented as 2^{n-k}, where n represents the total number of factors and k means the fraction of the entire factorial that is deliberately omitted. For the experimental design in this study, four factors are considered: function type, batch size, noise level, and number of input dimensions. Each of these factors is assigned two levels: low (-) and high (+) noise levels and small and large batch sizes. The complete factorial design matrix, which presents all permutations of these factors, is shown in 4.

Subsequently, a generator is selected as the guiding column to determine the combinations retained in the fractional factorial design. The choice of generator is often based on the factor assumed to have the least impact on the results. In this investigation, the function shape is chosen as the generator.

Finally, the XOR operation is performed between the chosen generator column and the remaining columns to finalise the combinations to retain in the fractional factorial design. These retained combinations represent the parameter settings that will be subjected to further experimentation in this study. The chosen combinations are indicated in red in the factorial design matrix.

Some combinations were excluded due to the constraints of the testing environment, which only presents six functions: we omitted (- + + +) and (+ + - +).

In addition, we systematically cycle through the values of a single parameter, which provides us with the opportunity to analyse its impact on the test set. As such, the complete testing environment can be found in 5

Function Type	Batch Size	Input Dimension	Noise Level
-	-	-	-
-	-	-	+
-	-	+	-
-	-	+	+
-	+	-	-
-	+	-	+
-	+	+	-
-	+	+	+
+	-	-	-
+	-	-	+
+	-	+	-
+	-	+	+
+	+	-	-
+	+	-	+
+	+	+	-
+	+	+	+

Table 4: Factorial design table

Function Type	Batch Size	Input Dimension	Noise Level
Different Powers	5	10	0
Different Powers	5	10	5
Different Powers	5	10	10
Different Powers	5	10	20
Different Powers	5	10	40
Easom	2	2	5
Easom	2	10	5
Easom	2	25	5
Easom	2	50	5
Zakharov	2	2	0
Zakharov	5	2	0
Zakharov	10	2	0
Zakharov	25	2	0
Griewank	2	2	0
Griewank	2	2	5
Griewank	2	2	10
Griewank	2	2	20
Griewank	2	2	40
Schwefel	2	10	20
Schwefel	5	10	20
Schwefel	10	10	20
Schwefel	25	10	20
Ackley	5	2	10
Ackley	5	10	10
Ackley	5	25	10
Ackley	5	50	10

Table 5: Testing setup