Bayesian Optimization for Min Max Optimization

Dorina Weichert dorina.weichert@iais.fraunhofer.de
Fraunhofer Center for Machine Learning
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS
Schloss Birlinghoven, 53757 Sankt Augustin, Germany

Alexander Kister alexander.kister@iais.fraunhofer.de
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS
Schloss Birlinghoven, 53757 Sankt Augustin, Germany

Abstract

A solution that is only reliable under favourable conditions is hardly a safe solution. Min Max Optimization is an approach that returns optima that are robust against worst case conditions. We propose algorithms that perform Min Max Optimization in a setting where the function that should be optimized is not known a priori and hence has to be learned by experiments. Therefore we extend the Bayesian Optimization setting, which is tailored to maximization problems, to Min Max Optimization problems. While related work extends the two acquisition functions Expected Improvement and Gaussian Process Upper Confidence Bound; we extend the two acquisition functions Entropy Search and Knowledge Gradient. These acquisition functions are able to gain knowledge about the optimum instead of just looking for points that are supposed to be optimal. In our evaluation we show that these acquisition functions allow for better solutions - converging faster to the optimum than the benchmark settings.

Keywords: Bayesian Optimization, Min Max Optimization, Worst Case Robustness

1. Introduction

Only in the lab, during development, engineers have control over the environment in which their artifact has to work. When the artifact is used, these environmental conditions are out of control. So to guarantee a certain level of performance, engineers have to find a solution that is robust against changes in the environment. For this optimization we formally assume the artifacts' performance is given by a function $f : \Theta \times Z \to \mathbb{R}$, where Θ is the space of controllable parameters and Z is the space of environment conditions, and use the Min Max Optimization given by:

$$\min_{\theta \in \Theta} \max_{\zeta \in Z} f(\theta, \zeta). \tag{1}$$

In this paper, we will study this problem with a continuous set of controllable parameters Θ and a discrete set of uncontrollable parameters Z.

In many practical situations the performance metric, $f(\theta, \zeta)$, is unknown and has to be treated as a black box. Bayesian Optimization is an established iterative framework for optimization of black box functions, that is - due to its sample efficiency - particularly useful when function evaluations are costly (Shahriari et al. (2016)). In each iteration, Bayesian Optimization suggests a new location $(\boldsymbol{\theta}_{n+1}, \boldsymbol{\zeta}_{n+1})$. After evaluating the black box function at this location, yielding the observation y, the existing data set of observations $\mathcal{D}_n$ is extended with this observation $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{((\theta_{n+1}, \zeta_{n+1}), y)\}$. Typically Bayesian Optimization uses a Gaussian Process (Rasmussen and Williams (2006)) as a surrogate model and an acquisition function for suggesting the next experiment to be performed (or, in other words, the next parameter to be evaluated), (Srinivas et al. (2010); Hennig and Schuler (2012); Frazier (2018); Shahriari et al. (2016)).

A distinguishing feature between acquisition functions are the properties a returned candidate is expected to have. The Gaussian Process Upper Confidence Bound (GP-UCB) (Srinivas et al. (2010)) and Expected Improvement (Jones et al. (1998)) return candidates that are potentially the optimum, while Entropy Search (Hennig and Schuler (2012)) and Knowledge Gradient (Frazier (2018); Frazier et al. (2009)) return candidates that increase the information about the optimum. So, as the information can increase also by by the exclusion of candidates, these latter acquisition functions explicitly encourage to evaluate non optimal points.

Related Work The existing approaches for extending the Bayesian Optimization framework to the Min Max problem usually pick an acquisition function that was designed to work for maximization problems and alter it such that it also works for the Min Max problem. The literature is dominated by approaches that extend GP-UCB (e.g., Sessa et al. (2020); Bogunovic et al. (2018); Wabersich and Toussaint (2015)) or the Expected Improvement (ur Rehman and Langelaar (2015),Marzat et al. (2016)). They use two acquisition functions: one to find the candidate for the controllable parameter and one to find the uncontrollable parameter that is a good candidate for the worst case match for the controllable parameter candidate. The GP-UCB approaches of Sessa et al. (2020) and Bogunovic et al. (2018) tackle problems that cover the Min Max problem as a special case.

Our approach deals with the adaption of two acquisition functions: Entropy Search and Knowledge Gradient. They were already adapted to the case when robustness is not required with respect to the worst case but with respect to the mean: Fröhlich et al. (2020) adapted the Entropy Search while Toscano-Palmerin and Frazier (2018) adapted the Knowledge Gradient. While, for the Knowledge Gradient, the necessary adaptions are similar for both kinds of robustness, they differ strongly when it comes to Entropy Search.

In contrast to the GP-UCB and Expected Improvement approaches, the focus of Entropy Search and Knowledge Gradient on information improvement allows us to use only one acquisition function. Our hypothesis is that the empirically observed good performance of Entropy Search and Knowledge Gradient for the maximization problem are inherited by our adaptions to the Min Max Optimization.

Contributions In short, our contributions are: (i) the adaption of the Entropy Search and Knowledge Gradient acquisition functions for the Min Max problem, (ii) the demonstration of their efficiency in comparison to Thompson Sampling (Thompson (1933)) and the algorithm of Wabersich and Toussaint (2015) (on synthetic problems), (iii) the discussion of the results, showing the advantageous behaviour of our adaptions and (iv) an outline of our future work.

2. Bayesian Optimization for Min Max Optimization

We tailor the two acquisition functions, Entropy Search (Hennig and Schuler (2012)) and Knowledge Gradient (Frazier (2018); Frazier et al. (2009)), so that they are applicable for searching the Min Max point.

Entropy Search In Entropy Search, we seek candidates that improve the knowledge about the location of the optimum. Given the set of observations $\mathcal{D}_n$, the Entropy Search acquisition function is defined by: $\alpha_{\text{ES}}((\boldsymbol{\theta}, \boldsymbol{\zeta}); \mathcal{D}_n) := \mathbb{E}\left[\eta_n^\star - \eta_{n+1}^\star \middle| (\boldsymbol{\theta}_{n+1}, \boldsymbol{\zeta}_{n+1}) = (\boldsymbol{\theta}, \boldsymbol{\zeta})\right]$, where $\eta_n^\star$ is the entropy of a distribution p_{opt} derived from the current Gaussian Process surrogate model that represents our knowledge about the location of the optimum and $\eta_{n+1}^\star = \eta_{n+1}^\star(y)$ is the same quantity derived from the Gaussian Process surrogate model that was updated with the fictive observation y at location $(\boldsymbol{\theta}, \boldsymbol{\zeta})$. The expectation is taken with respect to the measure for the fictive observation y induced by the (not updated) current surrogate model. The distribution p_{opt} is given by $p_{\text{opt}}((\boldsymbol{\theta}, \boldsymbol{\zeta})) = p((\boldsymbol{\theta}, \boldsymbol{\zeta}) = (\boldsymbol{\theta}^\star, \boldsymbol{\zeta}^\star))$, where p is the measure of the Gaussian Process surrogate model and $(\boldsymbol{\theta}^\star, \boldsymbol{\zeta}^\star)$ is the searched optimum. In our context, this is the Min Max point and in the original Entropy Search setting it is the global maximum.

In the original maximization setting there is no closed form expression for the distribution and instead two approximations are used: first, the uncountable search space is replaced by a finite set of representative points. Second, Expectation Propagation (EP) (Minka and Lafferty (2013)) is used for estimating the distribution over the optimum within these representative points. We follow the first approximation and introduce the representative controllable parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_N$. While following the second approximation we encountered the obstacle that in the Min Max setting the distribution does not have a nice multiplicative decomposition into terms that depend on maximally two locations of the Gaussian Process (as it is needed for EP). Instead, we express: $p_{\text{opt}}((\boldsymbol{\theta}, \boldsymbol{\zeta})) = p_{\text{Min Max}}\left((\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star)\right)$ as

$$\int \prod_{\substack{\boldsymbol{\zeta} \in Z \\ \boldsymbol{\zeta}^\star \neq \boldsymbol{\zeta}}} H\left[f\left(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star\right) - f\left(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}\right)\right] \prod_{i \in \{1,\dots,N\} \setminus i^\star} \left(1 - \prod_{\substack{\boldsymbol{\zeta} \in Z \\ \boldsymbol{\zeta}^\star \neq \boldsymbol{\zeta}}} H\left[f\left(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star\right) - f\left(\boldsymbol{\theta}_i, \boldsymbol{\zeta}\right)\right]\right) p\left(f\right) df, \tag{2}$$

where H is the heavy side function and p is the measure of the Gaussian Process. To derive this expression we used that stating the point $(\boldsymbol{\theta}^\star, \boldsymbol{\zeta}^\star)$ is the Min Max point of function f is equivalent to the following two statements:

- For the worst case optimal controllable parameter $\boldsymbol{\theta}_{i^\star}$, the function value $f(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star)$ is the worst case among all uncontrollable parameter; in short: $f(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star) \geq f(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta})$ for all $\boldsymbol{\zeta} \in Z$,

- For all other controllable parameter settings $\boldsymbol{\theta}_i$ for $i \in \{1, ..., N\} \setminus i^\star$, the function value $f(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star)$ is exceeded for at least one uncontrollable parameter setting; in short $f(\boldsymbol{\theta}_{i^\star}, \boldsymbol{\zeta}^\star) \leq f(\boldsymbol{\theta}_i, \boldsymbol{\zeta})$ for at least one $\boldsymbol{\zeta} \in Z$.

Our workaround for the sum that emerges after multiplying out the second product in formula 2 is to condition on the argmax function g, this is the function that maps each controllable parameter setting $\boldsymbol{\theta}$ to the worst case uncontrollable parameter setting. We consider the conditional probability of $\boldsymbol{\theta}_{i^\star}$ being the minimizer of the worst case function: $P(\boldsymbol{\theta}_{i^\star}$ is optimal $|$ g is the argmax function). Up to a normalization constant this conditional probability is given by

$$\int \prod_{i=1}^{N} \prod_{\substack{\boldsymbol{\zeta}\in Z \\ \boldsymbol{\zeta}^\star \neq \boldsymbol{\zeta}}} H\left[f\left(\boldsymbol{\theta}_i, g\left(\boldsymbol{\theta}_i\right)\right) - f\left(\boldsymbol{\theta}_i, \boldsymbol{\zeta}\right)\right] \prod_{i\in\{1,\ldots,N\}\setminus i^\star} H\left[f\left(\boldsymbol{\theta}_i, g\left(\boldsymbol{\theta}_i\right)\right) - f\left(\boldsymbol{\theta}_{i^*}, g\left(\boldsymbol{\theta}_{i^*}\right)\right)\right] p\left(f\right) df, \tag{3}$$

where the first product (over the index i) under the integral is the indicator for the function g to actually be the argmax function and the second product is the indicator for $\boldsymbol{\theta}_{i^\star}$ to be the minimizer of the worst case function $f(\boldsymbol{\theta}_i, g(\boldsymbol{\theta}_i))$. To estimate (the unconditional) probability of $\boldsymbol{\theta}_{i^\star}$ being the minimizer of the worst case function, we sample argmax functions $g_1, g_2, ..., g_M$ from the Gaussian Process and take the mean of the resulting conditional probabilities.

As the integrand in formula 3 is a product of terms that depend on maximally two locations of a Gaussian Process, we can reuse large parts of existing Entropy Search implementations: our implementation is based on the GPyOpt package (The GPyOpt authors (2016)).

Knowledge Gradient The Knowledge Gradient acquisition function α_{KG} (Frazier et al. (2009); Frazier (2018)) reflects how strongly the optimum of the mean of the surrogate model is influenced by a function evaluation at a given location. The acquisition function is defined analogously to the one for Entropy Search. The entropies, $\eta_n^\star$ and $\eta_{n+1}^\star(y)$, are replaced by the means, $\mu_n^\star$ and $\mu_{n+1}^\star(y)$, of the corresponding Gaussian Processes: $\alpha_{\mathrm{KG}}((\boldsymbol{\theta}, \boldsymbol{\zeta}); \mathcal{D}_n) := \mathbb{E}\left[\mu_n^\star - \mu_{n+1}^\star \middle| (\boldsymbol{\theta}_{n+1}, \boldsymbol{\zeta}_{n+1}) = (\boldsymbol{\theta}, \boldsymbol{\zeta})\right]$,

For maximization, an implementation is described in Frazier (2018) Algorithm 2. Adopting this approach, that uses Monte Carlo estimates for the expectation and grid search for maximising the acquisition function, is straight forward. The more scalable approach, that uses stochastic gradient descent for finding the maximum of the acquisition function, as implemented in Algorithm 3 and 4 in Frazier (2018), is left for future work.

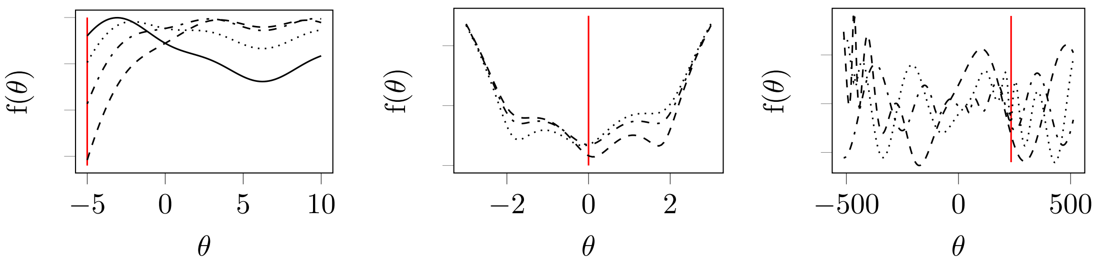
3. Experiments

To test our approaches, we adapt three synthetic two-dimensional problems, namely the branin, the six-hump camel and the eggholder function[1]. Our test cases, visualized in Figure 1, are representative for three problems: the robust optimum is at the boundary, at a not differentiable and at a differentiable location of the worst case function. We benchmark our methods[2] against Thompson Sampling (Thompson (1933)) and the GP-UCB approach of Wabersich and Toussaint (2015).

Further details about our experimental setup and the benchmark implementations can be found in appendix A and appendix B.

We run 100 trials with randomized initializations of (in total) 5 points and measure the mean absolute residuals (the mean absolute difference of the current function value at the estimated Min Max location and the function value at the real Min Max location), see

1. https://www.sfu.ca/∼ssurjano/optimization.html
2. https://github.com/fraunhofer-iais/MinMaxOpt



(a) branin⋆ (b) six-hump camel⋆ (c) eggholder⋆

Figure 1: The three different test problems are adaptions of the functions branin, six-hump camel and eggholder. The vertical line indicates the Min Max location. In branin⋆, the Min Max point is at the boundary and the argmax function is nearly constant. In six-hump camel⋆ the Min Max point is at a non differentiable point of the argmax functions. In eggholder⋆, the optimum lies in a local optimum of one of the functions and the argmax changes frequently.

Figure 2. For all test cases, Entropy Search, Knowledge Gradient and Thompson Sampling show superior performance over the benchmark of Wabersich and Toussaint (2015). The benchmark's unfavourable behaviour is caused by the deteriorations due to maximization in the inner loop that could not be counterbalanced by the improvement due to minimization in the outer loop. This is particularly noticeable for the branin⋆ and the camel⋆ problem, as the oscillation is especially large in the first iterations, when the minimizing loop lacks a sufficient amount of data. For eggholder⋆, the performance in the first iterations is comparable to the other acquisition functions, but the algorithm of Wabersich and Toussaint (2015) tends to get stuck in a local minimum. Knowledge Gradient shows a fast convergence in the first iterations, but sticks to local optima as well, as can be seen for the eggholder⋆ case. This is due to a lack of exploration, as the Knowledge Gradient only concentrates on the difference of the mean function values. On the contrary, Thompson Sampling, that performs well on branin⋆ and camel⋆, explores too heavily on the eggholder⋆ problem. This is due to the short (and fixed) lengthscales of the underlying Gaussian Process, as their is a high variance in the samples drawn from it, when the amount of training data is low. Entropy Search is comparable to Thompson Sampling in the smooth problems (branin⋆ and camel⋆). But it shows its superior performance on the eggholder⋆ problem, as it does not explore as aggressively as Thompson Sampling, but exploits the problem structure when estimating the probability of being the Min Max location. Further analyses (e.g., standard deviations, influence of algorithm parameters in Entropy search) are provided in appendix C.

4. Conclusion and Future Work

We extended two existing acquisition functions - Entropy Search and Knowledge Gradient - such that they are applicable for solving the Min Max problem. We compared and benchmarked the algorithms on three representative problems. We could show the comparable or advantageous performance of our approaches against two existing benchmarks, namely Thompson Sampling (Thompson (1933)) and Wabersich and Toussaint (2015). The

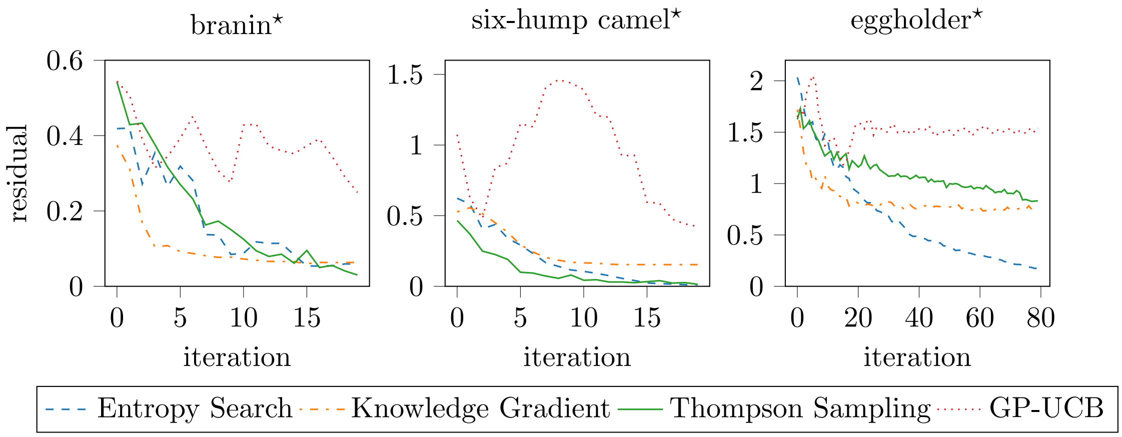


Figure 2: Mean residuals of the different algorithms on the three test problems.

Knowledge Gradient acquisition function shows a very fast convergence during the first iterations but might get stuck in local minima. Entropy search converges slower, but it comes close to the optimum even for the eggholder problem in an acceptable number of iterations. Regarding our hypothesis in the beginning, we showed that the empirically shown good performance of the acquisition functions Knowledge Gradient and Entropy Search for maximization is transferable to Min Max Optimization.

Future Work There are two directions for future work: enhancements to the adaption of the Entropy Search acquisition function and to the experimental sections. For the adoption of the Entropy Search acquisition function, Gessner et al. (2019) enables us to directly approximate Equation 2 with the Expectation Propagation algorithm, instead of enforcing a product representation as we are doing currently by conditioning on the argmax function.

Furthermore, Max-value Entropy Search (Wang and Jegelka (2017)) allows to apply that the Min Max value is the maximal value that is, for each controllable parameter setting, exceeded (including matched) for at least one setting of the uncontrollable parameter. This alternative approach has the potential to be more accurate and easier to compute as it was seen where the goal was maximization (Wang and Jegelka (2017)), where it was robustness with respect to the mean (Fröhlich et al. (2020)), and where it was Pareto optimality (Belakaria et al. (2019)).

We plan to perform tests on higher dimensional test settings and to benchmark against other state-of-the-art approaches such as (Sessa et al. (2020)). Additionally, as it is easy to extend our approaches to the case where the space of uncontrollable parameters Z is not finite, we will perform experiments to test its performance for this setting.

Acknowledgments

This work was jointly developed by the Fraunhofer Center for Machine Learning within the Fraunhofer Cluster for Cognitive Internet Technologies and the Fraunhofer Lighthouse projects ML4P and SWAP.

	branin*	six-hump camel*	eggholder*
description	negated branin	six-hump camel†	eggholder
fixed slices	{0, 4, 8, 12}	{-0.9, 0, 1}	{-512, 0, 185}
min-max location	(-5, 12)	(0, 0), (0, 1)	(234.647671, 185)
(σ_n, σ_v, l)	(0.001, 1, (0.2, 0.4))	(0.001, 0.5, (0.2, 0.2))	(0.001, 1, (0.09, 0.09))

Table 1: The test problems are adaptions of the traditional branin, six-hump camel and eggholder function. To match our Min Max problem, we treat the first dimension as controllable parameters $\boldsymbol{\theta}$ and the second as uncontrollable parameters $\boldsymbol{\zeta}$. We fix a set of parameters in the second dimension, producing multiple slices. Additionally, we make small adaptions to construct interesting problems for Min Max Optimization. The parameters (σ_n, σ_v, l) are the hyperparameters we used for the Gaussian Process. †: restricted to $[-3, 3] \times [-2, 2]$ and transformed by $\log(f(\boldsymbol{\theta}, \boldsymbol{\zeta}) + 2)$.

Appendix A. Test Setup

During the optimization, we use a Gaussian Process with an automatic relevance determination squared-exponential covariance function and zero mean function with fixed hyperparameters (signal variance σ_v^2, lengthscales l and noise variance σ_n^2), to avoid disturbances of the analysis due to wrongly estimated hyperparameters.

Furthermore, we applied the Gaussian Process model to scaled versions of the functions: the input locations of the functions are fit to the bounding box $[0, 1]^2$ and the output values of the functions are normalized to zero mean and variance 1. A summary of the test setups is provided in Table 1.

We developed our code with the use of the python packages GPyOpt (The GPyOpt authors (2016)) and BoTorch (Balandat et al. (2019)).

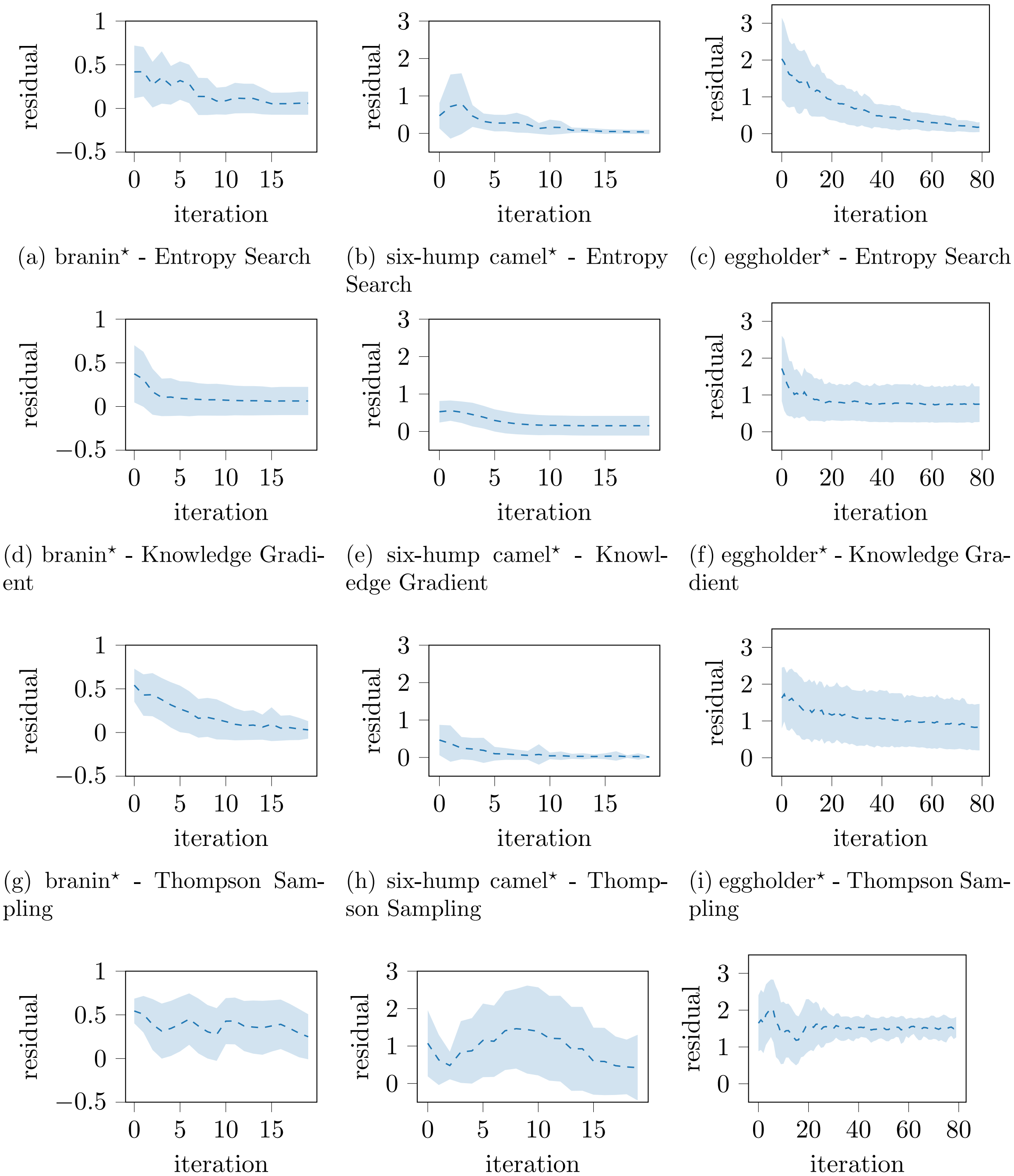
Appendix B. Benchmarks

For Thompson Sampling, a sample of a Gaussian Process is drawn in every iteration and its optimum location (here: the Min Max) used for the next evaluation. As we use a discretization to find the Min Max location of the sample (due to discontinuous partial first derivatives of the worst case function we cannot use traditional gradient based optimizers), we also use this discretization for sampling from the Gaussian Process, avoiding expensive operations like spectral sampling (Lázaro-Gredilla et al. (2010)).

The algorithm of Wabersich and Toussaint (2015) stays in the nested setting of the Min Max problem, resulting in an outer optimization loop for minimizing and an inner for maximimizing given the current candidate for the minimum. Here, GP-UCB with a tailored exploration-exploitation-tradeoff parameter β for the current optimization state, favouring exploration at the beginning of the optimization and exploitation at its end, is used.

Appendix C. Further results

For the standard deviations of the acquisition functions on the test cases, see Figure 3. The overall large size of standard deviations is due to the low number of samples for the



(a) branin⋆ - Entropy Search

(b) six-hump camel⋆ - Entropy Search

(c) eggholder⋆ - Entropy Search

(d) branin⋆ - Knowledge Gradient

(e) six-hump camel⋆ - Knowledge Gradient

(f) eggholder⋆ - Knowledge Gradient

(g) branin⋆ - Thompson Sampling

(h) six-hump camel⋆ - Thompson Sampling

(i) eggholder⋆ - Thompson Sampling

(j) branin⋆ - Benchmark

(k) six-hump camel⋆ - Benchmark

(l) eggholder⋆ - Benchmark

Figure 3: Standard deviations (one standard deviation interval is coloured) and means of residuals of the acquisition functions on the test problems. The benchmark of Wabersich and Toussaint (2015) is simply called ‘"Benchmark"’ here.

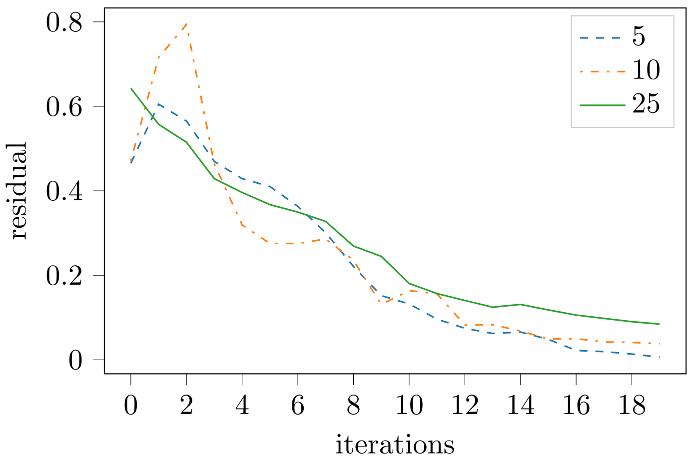


Figure 4: Residuals on the camel problem for different numbers of representative points and a fixed number of 10 argmax samples.

initialization. In the worst case, all 5 samples have the same coordinate for the second axis, resulting in a long exploration phase. The standard deviations of the information-based acquisition functions Entropy Search and Thompson sampling shrink with a higher number of iterations, as the algorithm gains further knowledge about the location of the optimum.

For the Entropy Search algorithm the number of representative points has a smoothing effect on the performance, see Figure 4. The higher the number of samples, the smoother the convergence plot; the speed of convergence is not effected (small differences are due to the high standard deviations of our experiments).

References

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: Programmable Bayesian Optimization in PyTorch. *arXiv preprint arXiv:1910.06403*, 2019.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7823–7833, 2019.

Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5765–5775, 2018.

Peter I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter I. Frazier, Warren B. Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

Lukas P. Fröhlich, Edgar D. Klenske, Julia Vinogradska, Christian Daniel, and Melanie N. Zeilinger. Noisy-input entropy search for efficient robust bayesian optimization. *arXiv preprint arXiv:2002.02820*, 2020.

Alexandra Gessner, Oindrila Kanjilal, and Philipp Hennig. Integrals over gaussians under linear domain constraints. *arXiv preprint arXiv:1910.09328*, 2019.

Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Miguel Lázaro-Gredilla, Joaquin Quiñonero Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

Julien Marzat, Eric Walter, and Hélène Piet-Lahanier. A new expected-improvement algorithm for continuous minimax optimization. *Journal of Global Optimization*, 64(4): 785–802, 2016.

Thomas P. Minka and John D. Lafferty. Expectation-propogation for the generative aspect model. *arXiv preprint arXiv:1301.0588*, 2013.

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

Pier Giuseppe Sessa, Ilija Bogunovic, Maryam Kamgarpour, and Andreas Krause. Mixed strategies for robust optimization of unknown objectives. *arXiv preprint arXiv:2002.12613*, 2020.

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(1):148–175, 2016.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

The GPyOpt authors. GPyOpt: a bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

William R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3):285–294, 1933.

Saul Toscano-Palmerin and Peter I. Frazier. Bayesian optimization with expensive integrands. *arXiv preprint arXiv:1803.08661*, 2018.

Samee ur Rehman and Matthijs Langelaar. Efficient global robust optimization of unconstrained problems affected by parametric uncertainties. *Structural and Multidisciplinary Optimization*, 52(2):319–336, 2015.

Kim P. Wabersich and Marc Toussaint. Automatic testing and minimax optimization of system parameters for best worst-case performance. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5533–5539, 2015.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning (ICML)*, pages 3627–3635, 2017.