Workshop on Real World Experiment Design and Active Learning at ICML 2020

Max-value Entropy Search for Multi-Objective Bayesian Optimization

Syrine Belakaria SYRINE.BELAKARIA@WSU.EDU
Washington State University

Aryan Deshwal ARYAN.DESHWAL@WSU.EDU
Washington State University

Janardhan Rao Doppa JANA.DOPPA@WSU.EDU
Washington State University

Abstract

We consider the problem of multi-objective (MO) blackbox optimization using expensive function evaluations, where the goal is to approximate the true pareto-set of solutions by minimizing the number of function evaluations. For example, in hardware design optimization, we need to find the designs that trade-off performance, energy, and area overhead using expensive computational simulations. In this paper, we propose a novel approach referred as *Max-value Entropy Search for Multi-objective Optimization (MESMO)* to solve this problem. MESMO employs an output-space entropy based acquisition function to efficiently select the sequence of inputs for evaluation to quickly uncover high-quality pareto-set solutions. We also provide theoretical analysis to characterize the efficacy of MESMO. Our experiments on several synthetic and real-world benchmark problems show that MESMO consistently outperforms the state-of-the-art algorithms.

1. Introduction

Many engineering and scientific applications involve making design choices to optimize multiple objectives. Some examples include tuning the knobs of a compiler to optimize performance and efficiency of a set of software programs; and designing new materials to optimize strength, elasticity, and durability. There are two common challenges in solving this kind of optimization problems: **1)** The objective functions are unknown and we need to perform expensive experiments to evaluate each candidate design choice. For example, performing computational simulations and physical lab experiments for compiler optimization and material design applications respectively. **2)** The objectives are conflicting in nature and all of them cannot be optimized simultaneously. Therefore, we need to find the *Pareto optimal* set of solutions. A solution is called Pareto optimal if it cannot be improved in any of the objectives without compromising some other objective. The overall goal is to approximate the optimal Pareto set by minimizing the number of function evaluations.

Bayesian Optimization (BO) Shahriari et al. (2016) is an effective framework to solve blackbox optimization problems with expensive function evaluations. The key idea behind BO is to build a cheap surrogate model (e.g., Gaussian Process Williams and Rasmussen (2006)) using the real experimental evaluations; and employ it to intelligently select the sequence of function evaluations using an acquisition function, e.g., expected improvement (EI). There is a large body of literature on single-objective BO algorithms Shahriari et al. (2016); Deshwal et al. (2020) and their applications including hyper-parameter tuning of machine learning methods Snoek et al. (2012); Kotthoff et al. (2017); Belakaria et al. (2020c).

However, there is relatively less work on the more challenging problem of BO for multiple objective functions Hernández-Lobato et al. (2016).

Prior work on multi-objective BO Belakaria et al. (2020a) is lacking in the following ways. Many algorithms reduce the problem to single-objective optimization by designing appropriate acquisition functions, e.g., expected improvement in Pareto hypervolume Knowles (2006); Emmerich and Klinkenberg (2008). Unfortunately, this choice is sub-optimal as it can potentially lead to aggressive exploitation behavior. Additionally, algorithms to optimize Pareto Hypervolume (PHV) based acquisition functions scale poorly as the number of objectives and dimensionality of input space grows. Other method relies on *input space entropy* based acquisition function Hernández-Lobato et al. (2016) to select the candidate inputs for evaluation. However, it is computationally expensive to approximate and optimize this acquisition function. More details about prior work can be found in Belakaria et al. (2019).

In this paper, we propose a novel and principled approach referred as **M***ax-value* **E***ntropy* **S***earch for* **M***ulti-objective* **O***ptimization* (MESMO) to overcome the drawbacks of prior work. MESMO employs an *output space entropy* based acquisition function to select the candidate inputs for evaluation. The key idea is to evaluate the input that maximizes the information gain about the optimal Pareto front in each iteration. Output space entropy search has many advantages over algorithms based on input space entropy search: a) allows much tighter approximation; b) significantly cheaper to compute; and c) naturally lends itself to robust optimization. Indeed, our experiments demonstrate these advantages of MESMO. Our work is inspired by the recent success of single-objective BO algorithms based on the idea of optimizing output-space information gain Wang and Jegelka (2017); Hoffman and Ghahramani (2015), which are shown to be most efficient and robust among a family of information-theoretic acquisition functions Hennig and Schuler (2012); Hernández-Lobato et al. (2014). Specifically, we extend the max-value entropy search approach Wang and Jegelka (2017) to the challenging multi-objective setting.

2. Background and Problem Setup

Bayesian Optimization (BO) Framework. BO is a very efficient framework to solve global optimization problems using *black-box evaluations of expensive objective functions.* Let $\mathfrak{X} \subseteq \Re^d$ be an input space. In single-objective BO formulation, we are given an unknown real-valued objective function $f : \mathfrak{X} \mapsto \Re$, which can evaluate each input $\vec{x} \in \mathfrak{X}$ to produce an evaluation $y = f(\vec{x})$. Each evaluation $f(\vec{x})$ is expensive in terms of the consumed resources. The main goal is to find an input $\vec{x^*} \in \mathfrak{X}$ that approximately optimizes f by performing a limited number of function evaluations. BO algorithms learn a cheap surrogate model from training data obtained from past function evaluations. They intelligently select the next input for evaluation by trading-off exploration and exploitation to quickly direct the search towards optimal inputs. The three key elements of BO framework are:

1) Statistical Model of the true function $f(x)$. *Gaussian Process (GP)* Williams and Rasmussen (2006) is the most commonly used model. A GP over a space $\mathfrak{X}$ is a random process from $\mathfrak{X}$ to $\Re$. It is characterized by a mean function $\mu : \mathfrak{X} \mapsto \Re$ and a covariance or kernel function $\kappa : \mathfrak{X} \times \mathfrak{X} \mapsto \Re$. If a function f is sampled from GP(μ, κ), then $f(x)$ is distributed normally $\mathcal{N}(\mu(x), \kappa(x, x))$ for a finite set of inputs from $x \in \mathcal{X}$.

2) Acquisition Function (α) to score the utility of evaluating a candidate input $\vec{x} \in \mathfrak{X}$ based on the statistical model. Some popular acquisition functions in the single-objective literature include expected improvement (EI), upper confidence bound (UCB), predictive entropy search (PES) Hernández-Lobato et al. (2014), and max-value entropy search (MES) Wang and Jegelka (2017).

3) Optimization Procedure to select the best scoring candidate input according to α depending on statistical model. DIRECT Jones et al. (1993) is a very popular approach for acquisition function optimization.

Multi-Objective Optimization (MOO) Problem. Without loss of generality, our goal is to minimize real-valued objective functions $f_1(\vec{x}), f_2(\vec{x}), \cdots, f_K(\vec{x})$, with $K \geq 2$, over continuous space $\mathfrak{X} \subseteq \Re^d$. Each evaluation of an input $\vec{x} \in \mathfrak{X}$ produces a vector of objective values $\vec{y} = (y^1, y^2, \cdots, y^K)$ where $y^i = f_i(x)$ for all $i \in \{1, 2, \cdots, K\}$. We say that a point $\vec{x}$ *Pareto-dominates* another point $\vec{x'}$ if $f_i(\vec{x}) \leq f_i(\vec{x'})$ $\forall i$ and there exists some $j \in \{1, 2, \cdots, K\}$ such that $f_j(\vec{x}) < f_j(\vec{x'})$. The optimal solution of MOO problem is a set of points $\mathcal{X}^* \subset \mathfrak{X}$ such that no point $\vec{x'} \in \mathfrak{X} \setminus \mathcal{X}^*$ Pareto-dominates a point $\vec{x} \in \mathcal{X}^*$. The solution set $\mathcal{X}^*$ is called the optimal *Pareto set* and the corresponding set of function values $\mathcal{Y}^*$ is called the optimal *Pareto front*. Our goal is to approximate $\mathcal{X}^*$ by minimizing the number of function evaluations.

3. MESMO Algorithm for Multi-Objective Optimization

In this section, we explain the technical details of our proposed MESMO algorithm. We first mathematically describe the output space entropy based acquisition function and provide an algorithmic approach to efficiently compute it.

Surrogate models. Gaussian processes (GPs) are shown to be effective surrogate models in prior work on single and multi-objective BO Hernández-Lobato et al. (2014); Wang et al. (2016); Wang and Jegelka (2017); Srinivas et al. (2009); Hernández-Lobato et al. (2016). Similar to prior work Hernández-Lobato et al. (2016), we model the objective functions $f_1, f_2, \cdots, f_K$ using K independent GP models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K$ with zero mean and i.i.d. observation noise. Let $\mathcal{D} = \{(\vec{x}_i, \vec{y}_i)\}_{i=1}^{t-1}$ be the training data from past $t-1$ function evaluations, where $\vec{x}_i \in \mathfrak{X}$ is an input and $\vec{y}_i = \{y_i^1, y_i^2, \cdots, y_i^K\}$ is the output vector resulting from evaluating functions $f_1, f_2, \cdots, f_K$ at $\vec{x}_i$. We learn surrogate models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K$ from $\mathcal{D}$.

Output space entropy based acquisition function. Input space entropy based methods like PESMO Hernández-Lobato et al. (2016) selects the next candidate input $\vec{x}_t$ (for ease of notation, we drop the subscript in below discussion) by maximizing the information gain about the optimal Pareto set $\mathcal{X}^*$. The acquisition function based on input space entropy is given as follows:

$$\begin{aligned}
\alpha(\vec{x}) &= I(\{\vec{x}, \vec{y}\}, \mathcal{X}^* \mid D) && (1)\\
&= H(\mathcal{X}^* \mid D) - \mathbb{E}_y[H(\mathcal{X}^* \mid D \cup \{\vec{x}, \vec{y}\})] && (2)\\
&= H(\vec{y} \mid D, \vec{x}) - \mathbb{E}_{\mathcal{X}^*}[H(\vec{y} \mid D, \vec{x}, \mathcal{X}^*)] && (3)
\end{aligned}$$

Information gain is defined as the expected reduction in entropy $H(.)$ of the posterior distribution $P(\mathcal{X}^* \mid D)$ over the optimal Pareto set $\mathcal{X}^*$ as given in Equations 2 and 3 (resulting from symmetric property of information gain). This mathematical formulation relies on a very expensive and high-dimensional ($m \cdot d$ dimensions) distribution $P(\mathcal{X}^* \mid D)$, where m is size of the optimal Pareto set $\mathcal{X}^*$. Furthermore, optimizing the second term in r.h.s poses significant challenges: a) requires a series of approximations Hernández-Lobato et al. (2016) which can be potentially sub-optimal; and b) optimization, even after approximations, is expensive c) performance is strongly dependent on the number of Monte-Carlo samples.

To overcome the above challenges of computing input space entropy based acquisition function, we take an alternative route and propose to maximize the information gain about the optimal **Pareto front** $\mathcal{Y}^*$. This is equivalent to expected reduction in entropy over the Pareto front $\mathcal{Y}^*$, which relies on a computationally cheap and low-dimensional ($m \cdot K$ dimensions, which is significantly less than $m \cdot d$ as $K \ll d$ in practice) distribution $P(\mathcal{Y}^* \mid D)$. Our acquisition function that maximizes the information gain between the next candidate input for evaluation $\vec{x}$ and Pareto front $\mathcal{Y}^*$ is given as:

The full derivation of our acquisition function can be found in Belakaria et al. (2019). We get the final form of our acquisition function as shown below:

$$\alpha(\vec{x}) \simeq \frac{1}{S}\sum_{s=1}^{S}\sum_{j=1}^{K}\left[\frac{\gamma_s^j(\vec{x})\phi(\gamma_s^j(\vec{x}))}{2\Phi(\gamma_s^j(\vec{x}))} - \ln\Phi(\gamma_s^j(\vec{x}))\right] \tag{4}$$

where $\gamma_s^j(x) = \frac{y_s^{j*} - \mu_j(\vec{x})}{\sigma_j(\vec{x})}$, $y_s^{j*} = \max\{z_1^j, \cdots z_m^j\}$, and ϕ and Φ are the p.d.f and c.d.f of a standard normal distribution respectively. A complete description of the MESMO algorithm is given in Algorithm 1. The blue colored steps correspond to computation of our output space entropy based acquisition function via sampling.

Algorithm 1 MESMO Algorithm

Input: input space $\mathfrak{X}$; K blackbox objective functions $f_1(x), f_2(x), \cdots, f_K(x)$; and maximum no. of iterations T_{max}

Initialize Gaussian process models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K$ by evaluating at N_0 initial points
for each iteration $t = N_0 + 1$ to T_{max} **do**
    Select $\vec{x}_t \leftarrow \arg max_{\vec{x} \in \mathfrak{X}} \ \alpha_t(\vec{x})$, where $\alpha_t(.)$ is computed as:
        for each sample $s \in 1, \cdots, S$:
            Sample $\tilde{f}_i \sim \mathcal{M}_i, \quad \forall i \in \{1, \cdots, K\}$
            $\mathcal{Y}_s^* \leftarrow$ Pareto front of *cheap* multi-objective optimization over $(\tilde{f}_1, \cdots, \tilde{f}_K)$
        Compute $\alpha_t(.)$ based on the S samples of $\mathcal{Y}_s^*$ as given in Equation 4
    Evaluate $\vec{x}_t$: $\vec{y}_t \leftarrow (f_1(\vec{x}_t), \cdots, f_K(\vec{x}_t))$
    Aggregate data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\vec{x}_t, \vec{y}_t)\}$
    Update models $\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K$
    $t \leftarrow t + 1$
end for
return Pareto front of $f_1(x), f_2(x), \cdots, f_K(x)$ based on $\mathcal{D}$

4. Experiments and Results

In this section, we describe our experimental setup, present results of MESMO on diverse real-world experiments, and compare MESMO with existing methods.

4.1 Experimental Setup

Multi-objective BO algorithms. We compare MESMO with existing methods described in the related work: ParEGO Knowles (2006), PESMO Hernández-Lobato et al. (2016), SMSego Ponweiser et al. (2008), EHI Emmerich and Klinkenberg (2008), and SUR Picheny (2015). We employ the code for these methods from the BO library Spearmint[1]. For methods requiring PHV computation, we employ the *PyGMO* library[2]. According to PyGMO documentation, the algorithm from Nowak et al. (2014) is employed for PHV computation. We did not include PAL Zuluaga et al. (2013) as it is known to have similar performance as SMSego Hernández-Lobato et al. (2016) and works only for finite discrete input space.

Statistical models. We use a GP based statistical model with squared exponential (SE) kernel in all our experiments. The hyper-parameters are estimated after every 5 function evaluations. We initialize the GP models for all functions by sampling initial points at random from a Sobol grid. This initialization procedure is same as the one in-built in the Spearmint library.

Real-world benchmarks. We employed four real-world benchmarks with data available at Zuluaga et al. (2013); Shah and Ghahramani (2016).

1) Hyper-parameter tuning of neural networks. In this benchmark, our goal is to find a neural network with high accuracy and low prediction time. We optimize a dense neural network over the MNIST dataset LeCun et al. (1998). Hyper-parameters include the number of hidden layers, the number of neurons per layer, the dropout probability, the learning rate, and the regularization weight penalties l_1 and l_2. We employ 10K instances for validation and 50K instances for training. We train the network for 100 epochs for evaluating each candidate hyper-parameter values on validation set. We apply a logarithm function to error rates due to their very small values.

2) SW-LLVM compiler settings optimization. SW-LLVM is a data set with 1024 compiler settings Siegmund et al. (2012) determined by d=10 binary inputs. The goal of this experiment is to find a setting of the LLVM compiler that optimizes the memory footprint and performance on a given set of software programs. Evaluating these objectives is very costly and testing all the compiler settings takes days.

3) SNW sorting network optimization. The data set SNW was first introduced by Zuluaga et al. (2012). The goal is to optimize the area and throughput for the synthesis of a field-programmable gate array (FPGA) platform. The input space consists of 206 different hardware design implementations of a sorting network. Each design is defined by $d = 4$ input variables.

4) Network-on-chip (NOC) optimization. The design space of NoC dataset Almer et al. (2011) consists of 259 implementations of a tree-based network-on-chip. Each configuration is defined by $d = 4$ variables: width, complexity, FIFO, and multiplier. We optimize energy and runtime of application-specific integrated circuits (ASICs) on the Coremark benchmark workload Almer et al. (2011).

In a recent work Belakaria et al. (2020b), MESMO is adapted to design and optimize electrified aviation power systems guided by expensive simulations.

1. https://github.com/HIPS/Spearmint/tree/PESM
2. https://esa.github.io/pygmo/

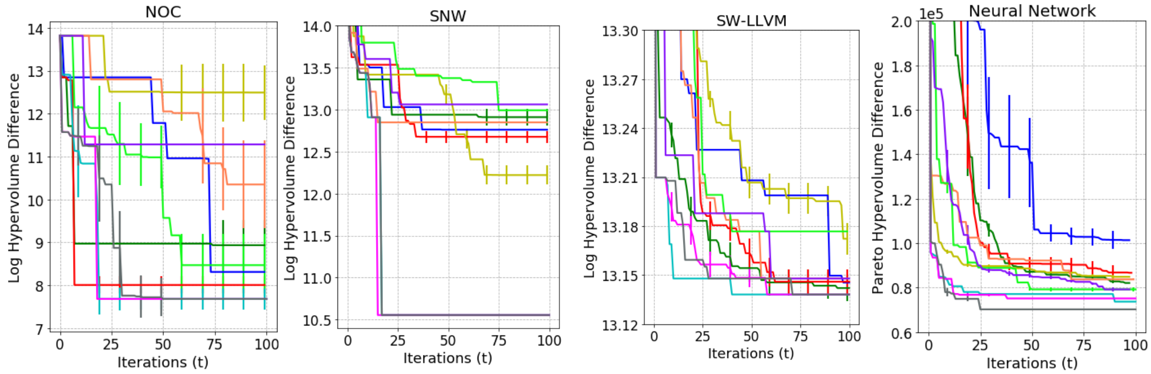

Figure 1: Results of different multi-objective BO algorithms including MESMO on real-world benchmarks. The log of the hypervolume difference is shown with different number of function evaluations. The mean and variance of 10 different runs are plotted. The title of each figure refers to the name of real-world benchmark. (Figures better seen in color.)

Evaluation metric. We employ a common metric used in practice: The Pareto hypervolume (PHV) Zitzler (1999).

4.2 Results and Discussion

We run all experiments 10 times. The mean and variance of the PHV metric across different runs are reported as a function of the number of iterations.

MESMO vs. State-of-the-art. We evaluate the performance of MESMO and PESMO with different number of Monte-Carlo samples for acquisition function optimization. Figure 1 shows the results of all multi-objective BO algorithms including MESMO for the four real-world experiments on PHV. Additional results with We present synthetic benchmarks and R_2 metric can be found in Belakaria et al. (2019). We make the following empirical observations: 1) MESMO consistently performs better than all baselines and also converges much faster. For blackbox optimization problems with expensive function evaluations, faster convergence has practical benefits as it allows the end-user or decision-maker to stop early. 2) Rate of convergence of MESMO slighly varies with different number of Monte-Carlo samples. However, in all cases, MESMO performs better than baseline methods. 3) The convergence rate of PESMO is dramatically affected by the number of Monte-Carlo samples: 100 samples lead to better results than 10 and 1. In contrast, *MESMO maintains a better performance consistently even with a single sample!*. The results strongly demonstrate that MESMO is much more robust to the number of Monte-Carlo samples than PESMO. 4) Performance of ParEGO is very inconsistent. In some cases, it is comparable to MESMO, but performs poorly on many other cases. This is expected due to random scalarization.

Comparison of acquisition function optimization time. We compare the runtime of acquisition function optimization for different multi-objective BO algorithms. The results and discussion of this comparison can be found in Belakaria et al. (2019). The acquisition function optimization time of MESMO is significantly smaller than PESMO for the same number of Monte-Carlo samples and comparable to ParEGO, which relies on scalarization to reduce to acquisition function optimization in single-objective BO.

References

Oscar Almer, Nigel Topham, and Björn Franke. A learning-based approach to the automated design of mpsoc networks. In *International Conference on Architecture of Computing Systems*, pages 243–258. Springer, 2011.

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

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 10044–10052, 2020a.

Syrine Belakaria, Derek Jackson, Yue Cao, Janardhan Rao Doppa, and Xiaonan Lu. Machine learning enabled fast multi-objective optimization for electrified aviation power system design. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020b.

Syrine Belakaria, Zhiyuan Zhou, Aryan Deshwal, Janardhan Rao Doppa, Partha Pande, and Deuk Heo. Design of multi-output switched-capacitor voltage regulator via machine learning. In *Proceedings of the Twenty-Third IEEE/ACM Design Automation and Test in Europe Conference (DATE)*, 2020c.

Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Alan Fern. Optimizing discrete spaces via expensive evaluations: A learning to search framework. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 3773–3780, 2020.

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Technical Report, Leiden University*, 34, 2008.

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

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1492–1501, 2016.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pages 918–926, 2014.

Matthew W Hoffman and Zoubin Ghahramani. Output-space predictive entropy search for flexible global optimization. In *NIPS workshop on Bayesian Optimization*, 2015.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1): 157–181, 1993.

Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Lars Kotthoff, Chris Thornton, Holger H Hoos, Frank Hutter, and Kevin Leyton-Brown. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *Journal of Machine Learning Research (JMLR)*, 18(1):826–830, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Krzysztof Nowak, Marcus Märtens, and Dario Izzo. Empirical performance of the approximation of the least hypervolume contributor. In *International Conference on Parallel Problem Solving From Nature*, pages 662–671. Springer, 2014.

Victor Picheny. Multi-objective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer, 2008.

Amar Shah and Zoubin Ghahramani. Pareto frontier learning with expensive correlated objectives. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1919–1927, 2016. URL https://github.com/amarshah/correl_bayesopt/tree/master/objectives/misc.

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.

Norbert Siegmund, Sergiy S Kolesnikov, Christian Kästner, Sven Apel, Don Batory, Marko Rosenmüller, and Gunter Saake. Predicting performance via automated feature-interaction detection. In *Proceedings of the 34th International Conference on Software Engineering (ICSE)*, pages 167–177, 2012.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

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*, 2009.

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

Zi Wang, Bolei Zhou, and Stefanie Jegelka. Optimization as estimation with gaussian processes in bandit settings. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1022–1031, 2016.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press, 2006.

Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Ithaca: Shaker, 1999.

Marcela Zuluaga, Peter Milder, and Markus Püschel. Computer generation of streaming sorting networks. In *Proceedings of Design Automation Conference (DAC)*, pages 1241–1249, 2012.

Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 462–470, 2013. URL http://www.spiral.net/software/pal.html.