Getting the Best Bang For Your Buck: Choosing What to Evaluate for Faster Bayesian Optimization

Md Shahriar Iqbal[1] **Jianhai Su**[1] **Lars Kotthoff**[2] **Pooyan Jamshidi**[1]

[1]University of South Carolina
[2]University of Wyoming

Abstract Machine learning system design frequently necessitates balancing multiple objectives, such as prediction error and energy consumption, for deep neural networks (DNNs). Typically, no single design performs well across all objectives; thus, finding Pareto-optimal designs is of interest. Measuring different objectives frequently incurs different costs; for example, measuring the prediction error of DNNs is significantly more expensive than measuring the energy consumption of a pre-trained DNN because it requires re-training the DNN. Current state-of-the-art methods do not account for this difference in objective evaluation cost, potentially wasting costly evaluations of objective functions for little information gain. To address this issue, we propose a novel cost-aware decoupled approach that weights the improvement of the hypervolume of the Pareto region by the measurement cost of each objective. To evaluate our approach, we perform experiments on several machine learning systems deployed on energy constraints environments.

1 Introduction

Many engineering and scientific applications require design decisions to be made to optimize multiple objectives $f_1(x), ..., f_n(x)$ over some bounded domain $\mathcal{X} \subset R^d$, where d is the dimensionality of the design space. For example, tuning DNN training and model design hyperparameters, as well as hardware and architectural design options to optimize objectives such as accuracy and energy in a DNN system. Solving these optimization problems is challenging mainly due to three reasons. (I) It is difficult to conduct efficient explorations of the enormous design space $\mathcal{X}$ that is formed by the combinatorial explosion of design options from different components of the DNN system. (II) The objective functions are unknown, and we must conduct costly experiments to evaluate each candidate design. (III) The objectives are inherently conflicting, and they cannot all be optimized at the same time. As a result, we must find the set of designs that is Pareto optimal.

In this work, we provide a novel solution for a classical problem—finding Pareto-optimal design (in exponentially large design space) given a fixed limited budget. The overall goal is to minimize the number of function evaluations to approximate the optimal Pareto set. Multi-objective Bayesian Optimization (MOBO) is an effective framework to solve black-box optimization problems with expensive function evaluation. A common strategy is to estimate each function f using a probabilistic model $\mathcal{M}$, such as a Gaussian process (GP) [12, 17, 16]. These strategies use the uncertainty captured by the probabilistic model to generate an acquisition function (a faster and cheaper proxy of the unknown objective function f), the maximum of which provides an effective heuristic for identifying a promising location on which to evaluate the objectives at each iteration t to identify Pareto optimal designs $\mathcal{X}^*$.

Existing Gap. Existing MOBO approaches are classified into the following categories based on their cost distribution assumptions (cost aware [1] vs decoupled [2]): (I) Coupled Unaware (e.g., PAL [23]),

[1] Incorporates the costs of evaluating objectives for choosing objectives for evaluating a design. Note that cost-aware approaches exist, in that they use the cost of evaluating designs (across all objectives) as constraints to decide whether to select a design during the iterative optimization. vs unaware) and evaluation strategy (coupled

[2] Only a subset of objectives is evaluated for the selected design at each iteration.

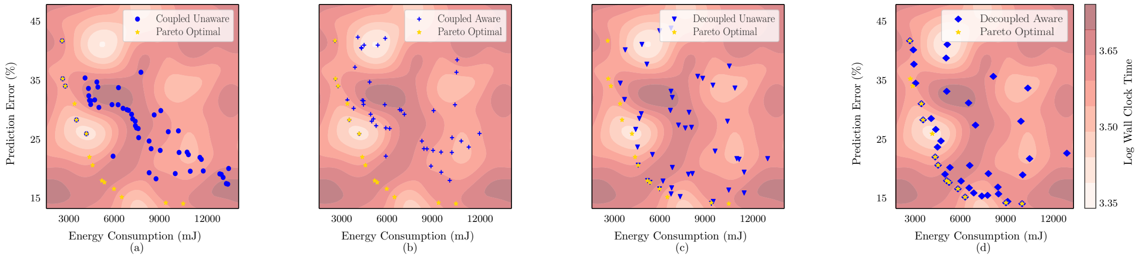

Figure 1: (a) Coupled unaware approaches wastes resources by evaluating designs with higher evaluation costs (b) Coupled aware approaches can suffer due to poor performance if the Pareto optimal designs are from regions of high evaluation cost (c) Decoupled aware approaches invests a lot of resources in evaluating designs with lower quality (high prediction error and high energy consumption) and do not perform well across both objectives (d) Decoupled aware approaches find designs with better quality for the objective evaluation cost when compared to other approaches.

ϵ-PAL [22], PESMO [9], MESMO [2], MESMOC [3]) (II) Coupled Aware (e.g., CA-MOBO [1], CArBO [14] [3]), and (III) Decoupled Unaware (e.g., PESMO-DEC [9]). However, none of these approaches are particularly useful for budget-constrained applications when the difference between the objective evaluation costs is sufficiently high. For example, measuring the prediction error of DNNs is orders of magnitude more expensive than measuring the energy consumption of a pre-trained DNN, as it requires re-training the DNN while optimizing the prediction error and energy consumption of a DNN system. Because the methods are unaware of the non-uniform objective evaluation costs, they waste resources evaluating the selected designs across all objectives, even if there is little or no gain through a specific objective.

Motivation. To address these limitations, we propose a decoupled cost-aware MOBO approach that takes into account the non-uniformity of objective evaluation costs and evaluates expensive objectives only if the information gained from the evaluation is worthwhile. To motivate our work, we performed a sandbox experiment to optimize the prediction error and energy consumption of an image recognition DNN system SqueezeNet [10] with CIFAR-10 dataset deployed on a resource-constrained Nvidia JETSON TX2 device for inference on 5,000 test images. Additionally, we use 8 Nvidia Tesla K80 GPUs deployed on Google cloud for training using 45,000 training images. We tuned the following hardware, architectural and DNN design options: CPU Frequency, GPU Frequency, Swappiness, Memory Growth, Filter Size, Number of Filters, and Number of epochs to compare our decoupled aware approach with coupled unaware: PAL, coupled aware: CA-MOBO, and decoupled unaware: PESMO-DEC approaches.

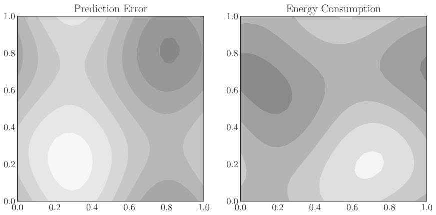

Figure 2: Contour curves for prediction error (left) and energy consumption (right) of SqueezeNet by varying CPU Frequency and Number of Filters while keeping the other design options fixed. Decoupled unaware approaches perform poorly when objectives with different costs have the same complexity (both non-linear).

Figure 1(a) indicates that the designs selected by coupled unaware approaches for evaluation have higher costs given the quality (lower prediction error and energy consumption indicate better quality) of the designs in comparison with decoupled aware approaches. Coupled aware approaches evaluate a high number of cheap designs

[3] CArBO is a single objective optimization technique

by avoiding the expensive regions in the search space. However, these methods can produce sub-optimal results when the Pareto optimal designs are located in the expensive regions of the search space, as shown in Figure 1(b). We also observe that the exploration-exploitation trade-off of these approaches is not balanced considering the cost. Compared to their coupled counterparts, decoupled unaware approaches traverse the search space more uniformly across designs from regions of different evaluation costs. However, they also waste resources by evaluating a higher number of low-quality designs across the more expensive objective e.g., prediction error as shown in Figure 1(c) when compared to decoupled aware approaches. This happens because decoupled unaware approaches are not notably effective when the complexity of the objective functions are the same, as shown in Figure 2 and the difference between their evaluation cost is significantly high. On the other hand, from Figure 1(d) we discover that our decoupled aware technique addresses each of the above limitations by a more balanced exploration across the search space considering the evaluation cost. We also observe that our approach does not waste resources in evaluating many designs in regions with a higher cost if the quality of the design is low.

Our approach. We extend on the concepts of the cutting-edge MOBO techniques PAL [23] and PESMO-DEC [9] by defining a function for *objective evaluation cost* using their computational time. Our acquisition function incorporates the uncertainty of the GP prediction as well as the objective evaluation cost to balance the exploration and exploitation, which iteratively improves the quality of the Pareto optimal search space, also known as the *Pareto region.* This acquisition function selects the next objective along with the next design for evaluation. As a result, we can make a trade-off between the additional information obtained from an evaluation and the cost of obtaining it, preventing us from performing costly evaluations for little potential gain. Our intuition is that by avoiding evaluation of the more costly objective without the necessity of evaluating it, we can traverse the objective space and find the Pareto optimal designs with increased efficiency. We demonstrate the promise of our approach via experimental evaluations on a variety of DNNs (see Table 1).

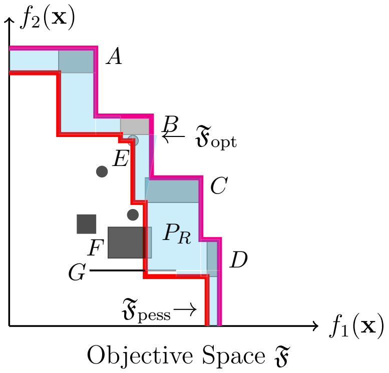
$f_2(\mathbf{x})$
A
B
$\leftarrow \mathfrak{F}_{\text{opt}}$
E
C
P_R
F
G
D
$\mathfrak{F}_{\text{pess}} \rightarrow$
$f_1(\mathbf{x})$
Objective Space $\mathfrak{F}$

Figure 3: Example showing pruning of non-dominated points to construct $\mathcal{F}_{opt}$, $\mathcal{F}_{pess}$. The shaded blue region P_R enclosed by $\mathcal{F}_{opt}$, $\mathcal{F}_{pess}$ indicates the Pareto region.

2 Methodology

In this section, we explain the technical details of our approach to identify the optimal Pareto front $\hat{\mathcal{F}}^*$ by evaluating a small subset of the design space $\mathcal{X}$ that uses a cost-aware acquisition function to incorporate the evaluation costs of each objective in the standard Bayesian optimization framework. Given the same budget, the cost-awareness of the acquisition function enables us to sample the search space more efficiently compared to other state-of-the-art approaches. Our approach is an active learning algorithm that not only selects a sequence of designs $(\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ in the design space $\mathcal{X}$ but also objectives $(f_{1,i}, ..., f_{T,i})$, where $1 \leq i \leq n$ for evaluation to predict a Pareto front $\hat{\mathcal{F}}^*$. We evaluate a design $\boldsymbol{x}$ across an objective f_i if the information gain is large enough compared to the objective evaluation cost $\theta_{t,i}$. This allows us to avoid expensive measurements for little or no information gain, and to only evaluate across an objective when the information gain is worthy compared to the evaluation cost. Our objective evaluation cost function is defined as $\theta_{t,i} = \log\left(t_{wc,i}\right)$, where $t_{wc,i}$ is the wall-clock time required to evaluate an objective f_i at iteration t.

We initially select a set of designs $\mathcal{X}_m$ from the design space $\mathcal{X}$ using Monte-Carlo sampling technique [19]. We then model each objective function f_i with a separate surrogate model $\mathcal{M}_i$. The objective values of a design $\boldsymbol{x}$ that has not been evaluated across any objective are estimated

Domain	Architecture	Dataset	Compiler	Num. Layers	Num. Params	Train Size	Test Size
Image	ResNet [8]	CIFAR-10 [13]	Keras	50	25M	45K	5K
	SqueezeNet [10]	CIFAR-10 [13]	Keras	3	1.2M	45K	5K
NLP	BERT [5]	SQuAD 2.0 [18]	PyTorch	12	110M	56K	5K
Speech	DeepSpeech [7]	Common Voice [15]	PyTorch	9	68M	300 (hrs)	2 (hrs)

Table 1: The DNN architectures and datasets used in the experimental evaluation.

by $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{x}) = (\mu_1(\boldsymbol{x}), \dots, \mu_n(\boldsymbol{x}))$, and the associated uncertainty is estimated by $\boldsymbol{\sigma}(\boldsymbol{x}) = (\sigma_1(\boldsymbol{x}), \dots, \sigma_n(\boldsymbol{x}))$. At this point, we use the $\boldsymbol{\mu}_t(\boldsymbol{x})$ and $\boldsymbol{\sigma}_t(\boldsymbol{x})$ values to determine the uncertainty region $R_t(\boldsymbol{x})$ for each design $\boldsymbol{x} \in \mathcal{X}_m$. We define the uncertainty region associated with a prediction of the surrogate model as $R_t(\boldsymbol{x}) = \{\boldsymbol{y} : \boldsymbol{\mu}_t(\boldsymbol{x}) - \sqrt{\beta_t}\boldsymbol{\sigma}_t(\boldsymbol{x}) \leq \boldsymbol{y} \leq \boldsymbol{\mu}_t(\boldsymbol{x}) + \sqrt{\beta_t}\boldsymbol{\sigma}_t(\boldsymbol{x})\}$, where β_t is a scaling parameter that controls the exploration-exploitation trade-off. Similar to PAL [23], we use $\beta_t = 2/9 \log(n|\mathcal{X}_m|\pi^2 t^2/6\delta)$ for $\delta \in (0, 1)$. The dimension of $R_t(\boldsymbol{x})$ depends on the number of objectives n. Later, we exploit the information about the uncertainty regions to determine the non-dominated designs set $\mathcal{U}$ [21]. We then use the optimistic (maximum of $R_t(\boldsymbol{x})$) and pessimistic (minimum of $R_t(\boldsymbol{x})$) values of the non-dominated designs in $\mathcal{U}$ to build the optimistic Pareto front $\mathcal{F}_{opt}$, pessimistic Pareto front $\mathcal{F}_{pess}$, and Pareto region P_R as shown in Figure 3.

We now employ our cost-aware acquisition function, which makes use of an information gain I based on objective space entropy. Being cost-aware, our proposed acquisition function $\alpha_{t,i}(\boldsymbol{x})$ considers the evaluation cost $\theta_{t,i}$ across each objective f_i:

$$\alpha_{t,i}(\boldsymbol{x}) = \frac{I(\{\boldsymbol{x}, f_{t,i}(\boldsymbol{x})\}, \hat{\mathcal{F}}^*|\mathcal{X}_m^*)}{\theta_{t,i}} = \frac{V(P_R|\hat{\mathcal{F}}^*) - V\left(P_R|\hat{\mathcal{F}}^*_{R_{t,i}(\boldsymbol{x})=\boldsymbol{\mu}_{t,i}(\boldsymbol{x})}\right)}{\theta_{t,i}} = \frac{\Delta V_{t,i}}{\theta_{t,i}} \tag{1}$$

Here, $\alpha_{t,i}(\boldsymbol{x})$ computes the amount of information that can be gained per cost for a design $\boldsymbol{x}$ to be evaluated for an objective f_i. In Equation 1, we compute the gain of information as the change of volume of the Pareto region if the Pareto front $\hat{\mathcal{F}}^* = \mathcal{F}_{opt} \cup \mathcal{F}_{pess}$ is updated by setting the uncertainty values $R_{t,i}(\boldsymbol{x})$ of $\boldsymbol{x}$ to its mean $\mu_{t,i}(\boldsymbol{x})$ for the corresponding designs in $\mathcal{X}_m^*$. Our acquisition function computes the change of volume $\Delta V_{t,i}$ of the Pareto region P_R across each objective f_i to judiciously determine the gain of information that would be achieved if design $\boldsymbol{x}$ is evaluated for f_i. We select a design $\boldsymbol{x}_t$ and an objective $f_{t,i}$ using $\boldsymbol{x}_t, f_{t,i} = \text{argmax}_{\boldsymbol{x} \in \mathcal{X}_m^* \text{ for each } f_i} \alpha_{t,i}(\boldsymbol{x})$ to identify the most promising design for an objective function that gains the most information given the cost of evaluating it. Finally, we update the surrogate model $\mathcal{M}_i$ corresponding to the chosen objective function f_i by incorporating the newly-evaluated design and objective value. We stop when the maximum budget θ_{max} is exhausted and return the Pareto front obtained.

3 Experimental Setup and Results

In this section, we evaluate the effectiveness of our approach to optimize energy consumption and prediction error of DNNs in comparison to four state-of-the-art baselines such as PAL, PESMO, CA-MOBO, and PESMO-DEC. We use four DNN architectures from three different problem domains; Image, NLP, and Speech. For each architecture, we select the most common dataset and compiler typically used in practice, as shown in Table 1. We run each optimization pipeline 5 times using different initial evaluations, where the initial evaluations in one run are the same for all methods. We chose a number of architectures, hardware, and DNN design options based on similar hardware configuration guides/tutorials and other related work [6]. To reduce the effect of noise, we repeat energy measurements for each design 10 times and take the median; however, because prediction error measurements are stable, we do not repeat them [11]. We employ a distributed setup where the training of a DNN is done remotely on virtual machine instances with 8 NVIDIA Tesla K80 GPU deployed on the Google cloud and the measurements and optimization algorithms run locally on a

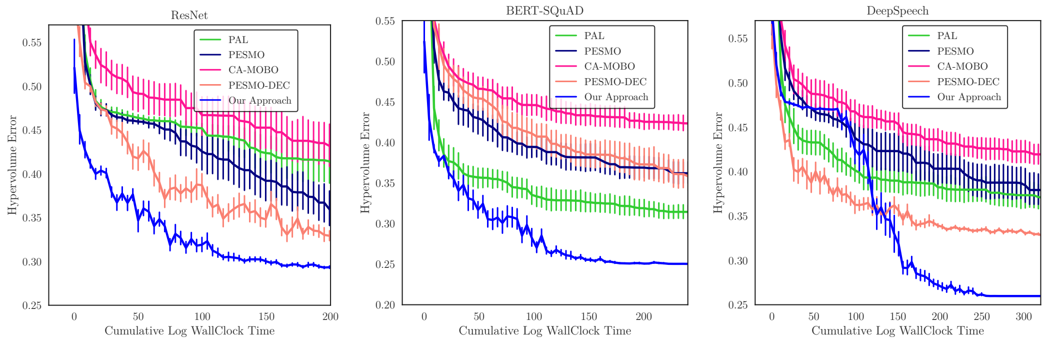

Figure 4: Comparison of hypervolume error obtained by our approach when compared to other MOBO baselines for DNNs for image recognition, NLP, and speech recognition applications.

resource-constrained Jetson TX2 device. Our experiments took a total of 1440 hours of wall-clock time to complete. Code and data are provided at `https://github.com/softsys4ai/FlexiBO`.

We evaluate the quality of the obtained Pareto fronts using the hypervolume error [4, 20] and the cumulative log wall-clock time as the objective evaluation cost required to obtain it. As the ground truth Pareto fronts are unknown, we approximate them by combining the Pareto fronts obtained by the different optimization methods considered in our experiments to compute the hypervolume error. From Figure 4, we observe that our approach consistently outperforms other methods in finding Pareto fronts with lower hypervolume error in each of the applications. For example, our approach achieves 4.8%, 7.6% and 8.2% lower hypervolume error in ResNet, BERT-SQuAD, and DeepSpeech, respectively, than the next best optimization method in that particular DNN system. These observations indicate that our approach is more effective than other baselines when the size of DNN increases. This is expected as the size (and training time) of the DNNs increases, the difference between objective evaluation costs also increases, and the penalty for wasting resources becomes higher given the budget constraints.

4 Discussions: Limitations and Impacts

In this work, we proposed a decoupled cost-aware acquisition function for Bayesian multi-objective optimization. Instead of evaluating all objective functions, we automatically choose the one that provides the highest benefit, weighted by the cost to perform the evaluation. We demonstrated the promise of our approach by conducting a comprehensive evaluation of three different DNN applications across a large design space on resource-constrained hardware platforms. While our evaluations are limited to optimizing DNN systems, we believe our approach can be generalized across domains. However, our method may not work as well for objectives with uniform evaluation costs, so we limit the scope to non-uniform objective evaluation costs.

Our method is especially useful for real-world machine learning systems, e.g., DNNs deployed in resource-constrained environments such as edge. Furthermore, when confronted with a performance bottleneck, this method can be useful in returning the system to a good operating region significantly faster than other baselines. Additionally, our method enriches the existing MOBO literature with a novel decoupled cost-aware technique. Finally, our approach can be used for better understanding of the joint optimization space of architecture, hardware, and DNN design options, as well as their interactions.

Acknowledgements

This work has been supported in part by NSF (Awards 2007202, and 2107463), Google, and Chameleon Cloud. We are grateful to all who provided feedback on this work, including Luigi Nardi, Mohammad Ali Javidian, Md Abir Hossen, and anonymous AutoML’22 reviewers.

References

[1] Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Cost-aware multi-objective bayesian optimisation. 2019.

[2] 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 7825–7835, 2019.

[3] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization with constraints. 2020.

[4] Yongtao Cao, Byran J Smucker, and Timothy J Robinson. On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, 160:60–74, 2015.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[6] Hassan Halawa, Hazem A Abdelhafez, Andrew Boktor, and Matei Ripeanu. Nvidia jetson platform characterization. In *European Conference on Parallel Processing*, pages 92–105. Springer, 2017.

[7] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] José Miguel Hernández-Lobato, Michael A Gelbart, Matthew W Hoffman, Ryan P Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. JMLR, 2015.

[10] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. 2016.

[11] Md Shahriar Iqbal, Lars Kotthoff, and Pooyan Jamshidi. Transfer Learning for Performance Modeling of Deep Neural Network Systems. In *USENIX Conference on Operational Machine Learning*, Santa Clara, CA, 2019. USENIX Association.

[12] 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.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Citeseer, 2009.

[14] Eric Hans Lee, Valerio Perrone, Cedric Archambeau, and Matthias Seeger. Cost-aware bayesian optimization. 2020.

[15] Mozilla, 2019. `https://commonvoice.mozilla.org/en/datasets`.

[16] Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6), 2015.

[17] 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.

[18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. 2016.

[19] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[20] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[21] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

[22] Marcela Zuluaga, Andreas Krause, and Markus Püschel. ε-pal: an active learning approach to the multi-objective optimization problem. *The Journal of Machine Learning Research*, 17(1):3619–3650, 2016.

[23] Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *International Conference on Machine Learning*, pages 462–470, 2013.