Journal of Machine Learning Research 23 (2022) 1-9 Submitted 8/21; Revised 11/21; Published 2/22

SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization

Marius Lindauer[1] LINDAUER@TNT.UNI-HANNOVER.DE
Katharina Eggensperger[2] EGGENSPK@CS.UNI-FREIBURG.DE
Matthias Feurer[2] FEURERM@CS.UNI-FREIBURG.DE
André Biedenkapp[2] BIEDENKA@CS.UNI-FREIBURG.DE
Difan Deng[1] DENG@TNT.UNI-HANNOVER.DE
Carolin Benjamins[1] BENJAMINS@TNT.UNI-HANNOVER.DE
Tim Ruhkopf[1] RUHKOPF@TNT.UNI-HANNOVER.DE
René Sass[1] SASS@TNT.UNI-HANNOVER.DE
Frank Hutter[2,3] FH@CS.UNI-FREIBURG.DE
[1]*Leibniz University Hannover,* [2]*University of Freiburg,* [3]*Bosch Center for Artificial Intelligence*

Editor: Alexandre Gramfort

Abstract

Algorithm parameters, in particular hyperparameters of machine learning algorithms, can substantially impact their performance. To support users in determining well-performing hyperparameter configurations for their algorithms, datasets and applications at hand, `SMAC3` offers a robust and flexible framework for Bayesian Optimization, which can improve performance within a few evaluations. It offers several facades and pre-sets for typical use cases, such as optimizing hyperparameters, solving low dimensional continuous (artificial) global optimization problems and configuring algorithms to perform well across multiple problem instances. The `SMAC3` package is available under a permissive BSD-license at `https://github.com/automl/SMAC3`.

Keywords: Bayesian Optimization, Hyperparameter Optimization, Multi-Fidelity Optimization, Automated Machine Learning, Algorithm Configuration

1. Introduction

It is well known that setting hyperparameter configurations of machine learning algorithms correctly is crucial to achieve top performance on a given dataset (Bergstra and Bengio, 2012; Snoek et al., 2012). Besides simple random search (Bergstra and Bengio, 2012), evolutionary algorithms (Olson et al., 2016) and bandit approaches (Li et al., 2018), Bayesian Optimization (BO) (Mockus et al., 1978; Shahriari et al., 2016) is a commonly used approach for Hyperparameter Optimization (HPO) (Feurer and Hutter, 2019) because of its sample efficiency. Nevertheless, BO is fairly brittle to its own design choices (Lindauer et al., 2019b) and depending on the task at hand different BO approaches are required.

`SMAC3` is a flexible open-source BO package that (i) implements several BO approaches, (ii) provides different facades, hiding unnecessary complexity and allowing easy usages, and (iii) can thus be robustly applied to different HPO tasks. Its usage in successful AutoML tools, such as `auto-sklearn` (Feurer et al., 2015) and the recent version of `Auto-PyTorch` (Zimmer et al., 2021), and as part of the winning solution to the latest BBO challenge (Awad et al., 2020), demonstrates its value as a useful tool besides academic research.

2. Different Use Cases and Modes of `SMAC3`

`SMAC3` is designed to be robustly applicable to a wide range of different use-cases. To allow for maximal flexibility, `SMAC3` does not only implement a Python interface, but also a CLI to communicate with arbitrary processes and programming languages. Furthermore, `SMAC3` supports two kinds of parallelization techniques: (i) via DASK (Rocklin, 2015) and (ii) via running an arbitrary number of independent `SMAC3` instances exchanging information via the file system. Last but not least, its modular design allows to combine different modules seamlessly. For a use-case at hand, the user only has to define a scenario, including the configuration space $\mathbf{\Lambda}$ (Lindauer et al., 2019a), and choose a facade encoding different pre-sets of `SMAC3`, see Figure 1. These pre-sets are specifically designed to be efficient based on the characteristics of the given use-case, as described in the following.

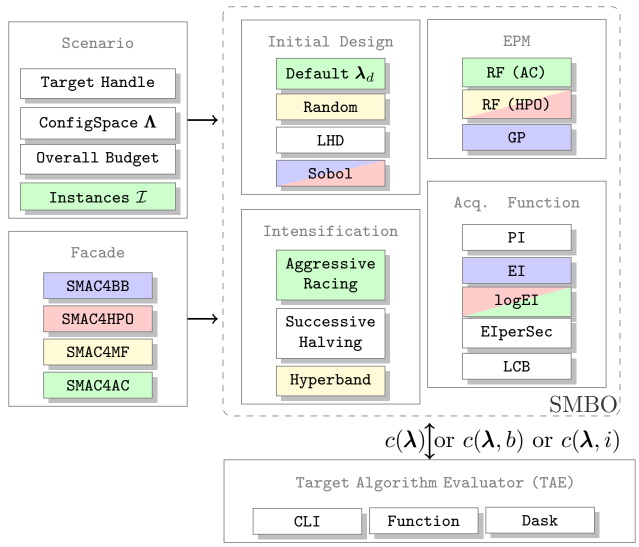

Figure 1: Simplified overview of components in `SMAC3`. We color-code the different pre-set options activated by different facades.

2.1 `SMAC4BB`: SMAC for Low-dimensional and Continuous Black-Box Functions

The most general view on HPO is one of black-box optimization, where an unknown cost function c is minimized with respect to its input hyperparameters $\boldsymbol{\lambda} \in \mathbf{\Lambda}$; one can equivalently frame this as minimizing the loss $\mathcal{L}$ on validation data $\mathcal{D}_{\text{val}}$ of a model trained on training data $\mathcal{D}_{\text{train}}$ with hyperparameters $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \mathbf{\Lambda}} c(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\lambda} \in \mathbf{\Lambda}} \mathcal{L}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}; \boldsymbol{\lambda}). \tag{1}$$

BO with Gaussian Processes (GP) is the traditional choice for HPO on continuous spaces with few dimensions. `SMAC3` builds on top of existing GP implementations and offers several acquisition functions, including LCB (Srinivas et al., 2010), TS (Thompson, 1933), PI and EI (Jones et al., 1998) and variants, e.g., EI per second (Snoek et al., 2012) for evaluations with different runtimes and logEI (Hutter et al., 2010) for heavy-tailed cost distributions. The pre-set of `SMAC4BB` follows commonly used components and comprises a Sobol sequence as initial design, a GP with 5/2-Matérn Kernel and EI as acquisition function, similar to Snoek et al. (2012).

2.2 SMAC4HPO: SMAC for CASH and Structured Hyperparameter Optimization

SMAC3 can also tackle the combined algorithm selection and hyperparameter optimization problem (CASH) (Thornton et al., 2013), and searches for a well-performing A_i from a *set of algorithms* $\mathbf{A}$ and its hyperparameters $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_i$:

$$(A^*, \boldsymbol{\lambda}^*) \in \operatorname*{arg\,min}_{A_i \in \mathbf{A}, \boldsymbol{\lambda} \in \boldsymbol{\Lambda}_i} c(A_i, \boldsymbol{\lambda}) = \operatorname*{arg\,min}_{A_i \in \mathbf{A}, \boldsymbol{\lambda} \in \boldsymbol{\Lambda}_i} \mathcal{L}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}; A_i(\boldsymbol{\lambda})). \tag{2}$$

The hyperparameter sub-space $\boldsymbol{\Lambda}_i$ for an algorithm A_i is only active if A_i was chosen. Therefore, there is a conditional hierarchy structure between the top-level hyperparameter choosing A_i and the subspace $\boldsymbol{\Lambda}_i$. SMAC3 models these spaces by using a random forest (Breimann, 2001), with SMAC having been the first BO approach to use this type of model (Hutter et al., 2011). SMAC3 supports multiple levels of conditionalities, e.g., one top-level hyperparameter choosing a classification algorithm and another sub-level hyperparameter an optimizer (for training NNs) enabling a second sub-level of hyperparameters. The pre-set of SMAC4HPO is based on the tuning by Lindauer et al. (2019b) and combines a Sobol sequence as initial design, a RF as surrogate model and logEI as acquisition function.

2.3 SMAC4MF: SMAC for Expensive Tasks and Automated Deep Learning

On some HPO tasks, e.g., optimizing hyperparameters and architectures for deep learning, training many models might be too expensive. Multi-fidelity optimization is a common approach, for cases where we can observe cheaper approximations of the true cost:

$$\boldsymbol{\lambda}^* \in \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} c(\boldsymbol{\lambda}, b_{max}) = \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \mathcal{L}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}; \boldsymbol{\lambda}, b_{max}). \tag{3}$$

Here, a configuration is evaluated with a budget $b \leq b_{max}$ (e.g., number of epochs, dataset size or channels of CNN) to obtain a cheap proxy of the true cost function at b_{max}. SMAC3 follows the principle of BOHB (Falkner et al., 2018) in combining Hyperband (Li et al., 2018) and BO, where the surrogate model is fitted on the highest budget-level with sufficient observations. However, its RF models tend to yield much better performance than BOHB's TPE models, see Figure 2 for illustrative examples. With the exception of the RF, the pre-set of SMAC4MF is similar to Falkner et al. (2018) and consists of a random initial design and Hyperband as intensification method.

2.4 SMAC4AC: SMAC for Algorithm Configuration

A more general view on the problem of HPO is called algorithm configuration (AC) (Hutter et al., 2009; Ansótegui et al., 2009; López-Ibáñez et al., 2016), where the goal is to determine a well-performing robust configuration across a set of problem instances i from a finite set $\mathcal{I}$:

$$\boldsymbol{\lambda}^* \in \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} c(\boldsymbol{\lambda}) = \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \sum_{i \in \mathcal{I}} c'(\boldsymbol{\lambda}, i) \tag{4}$$

The origin of SMAC lies in AC (Hutter et al., 2011). It inherits the ideas of aggressive racing (Hutter et al., 2009) which evaluates less promising candidate configurations on only a few instances, and to collect sufficient empirical evidence on many instances if the candidate

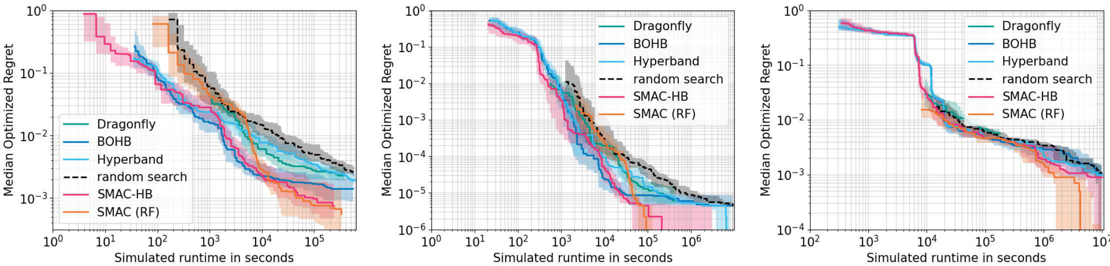

Figure 2: Comparison on Net_{Letter} (6D), NBHPO_{Naval} (9D) and Nas1Shot1_2 (9D). Since these are all tabular or surrogate benchmarks from HPOBench, runtime is simulated by table look-ups or surrogate predictions.

could become the next incumbent configuration. Furthermore, `SMAC3` supports imputation of right-censored observations and a logEI to model heavy-tailed cost distributions; it also uses a customized hyperparameter configuration of the RF surrogate model for AC. SMAC successfully optimized configuration spaces with more than 300 hyperparameters for SAT solvers (Hutter et al., 2017), yielding speedups of up to several orders of magnitude. The pre-set of `SMAC4AC` follows Hutter et al. (2011) and combines a single default configuration as initial design, a RF as surrogate model (with different hyperparameter settings compared to `SMAC4HPO`), logEI as acquisition function and aggressive racing as intensification mechanism.

3. Brief Empirical Comparison

To provide an impression on the sequential performance of `SMAC3`, we compared it against random search (Bergstra and Bengio, 2012)—although we consider it a weak baseline (Turner et al., 2020)—Hyperband (Li et al., 2018), Dragonfly (Kandasamy et al., 2020) and BOHB (Falkner et al., 2018) on the surrogate benchmark for HPO on DNNs on the letter dataset (Falkner et al., 2018), a joint HPO+NAS benchmark (Klein and Hutter, 2019) on the Naval-Propulsion dataset and a pure NAS benchmark (Zela et al., 2020).[1] As shown in Figure 2, `SMAC3`'s multi-fidelity approach (see Sec. 2.3) performs as well as Hyperband in the beginning, performs best in the middle, until `SMAC3`'s pure BO with RFs catches up in the end. For the whole time, `SMAC3` consistently outperforms Dragonfly and in the later phases, also BOHB. For a larger empirical study, incl. `SMAC3`, we refer to Eggensperger et al. (2021).

4. Related Work

With the initial success of BO for HPO (Hutter et al., 2011; Snoek et al., 2012; Bergstra et al., 2013), there were many follow up tools in recent years (Bakshy et al., 2018; Nardi et al., 2019; Kandasamy et al., 2020; Balandat et al., 2020; Li et al., 2021; Head et al., 2021). `SMAC3`'s advantage lies on the one hand in the efficient use of random forests as surrogate model for higher dimensional and complex spaces, and on the other hand, in its flexibility of combining different state-of-the-art BO and intensification strategies, such as aggressive

1. See `https://github.com/automl/HPOBench/` for details regarding the experimental setup.

racing and multi-fidelity approaches. In addition, evolutionary algorithms are also known as efficient black-box optimizers (Fortin et al., 2012; Olson et al., 2016; Loshchilov and Hutter, 2016; Rapin and Teytaud, 2018; Awad et al., 2021). Although there is the common belief that BO is particularly efficient for small budgets and evolutionary algorithms for cheap function evaluations (Feurer and Hutter, 2019), choosing the right optimizer for a given task is still an open problem. To address this, first systems have been introduced that schedule several optimizers sequentially to make use of their respective strengths (Awad et al., 2020; Lan et al., 2020; Turner et al., 2020).

5. Outlook

Although `SMAC3` performs robustly on many HPO tasks, it does not exploit their landscape structure. As a next step, we plan to integrate local BO approaches (Eriksson et al., 2019). Furthermore, `SMAC3` provides an easy-to-use `fmin`-API and facades, but choosing `SMAC3`'s own hyperparameters might still be challenging (Lindauer et al., 2019b). Therefore, we plan to add mechanisms to adaptively select `SMAC3`'s settings on the fly, e.g., via Bandits (Hoffman et al., 2011) or reinforcement learning (Biedenkapp et al., 2020).

Acknowledgements

Robert Bosch GmbH is acknowledged for financial support. This work has partly been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant no. 716721. The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG.

References

C. Ansótegui, M. Sellmann, and K. Tierney. A gender-based genetic algorithm for the automatic configuration of algorithms. In I. Gent, editor, *Proceedings of the Fifteenth International Conference on Principles and Practice of Constraint Programming (CP'09)*, volume 5732 of *Lecture Notes in Computer Science*, pages 142–157. Springer, 2009.

N. Awad, G. Shala, D. Deng, N. Mallik, M. Feurer, K. Eggensperger, A. Biedenkapp, D. Vermetten, H. Wang, C. Doerr, M. Lindauer, and F. Hutter. Squirrel: A switching hyperparameter optimizer description of the entry by AutoML.org & IOHprofiler to the NeurIPS 2020 BBO challenge. *arXiv:2012.08180 [cs.LG]*, 2020.

N. Awad, N. Mallik, and F. Hutter. DEHB: Evolutionary hyberband for scalable, robust and efficient hyperparameter optimization. In Z. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2147–2153. ijcai.org, 2021.

E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh. Ae: A domain-agnostic platform for adaptive experimentation. In *Proceedings of the international conference on Neural Information Processing Systems*, pages 1–8, 2018.

M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. Wilson, and E. Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*. Curran Associates, 2020.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

J. Bergstra, D. Yamins, and D. Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, page 20, 2013.

A. Biedenkapp, H. F. Bozkurt, T. Eimer, F. Hutter, and M. Lindauer. Dynamic Algorithm Configuration: Foundation of a New Meta-Algorithmic Framework. In J. Lang, G. De Giacomo, B. Dilkina, and M. Milano, editors, *Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20)*, pages 427–434, June 2020.

L. Breimann. Random forests. *Machine Learning Journal*, 45:5–32, 2001.

K. Eggensperger, P. Müller, N. Malik, M. Feurer, R. Sass, A. Klein, N. Awad, Marius Lindauer, and Frank Hutter. HPOBench: A collection of reproducible multi-fidelity benchmark problems for hpo. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

D. Eriksson, M. Pearce, J. Gardner, R. Turner, and M. Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 32nd International Conference on Advances in Neural Information Processing Systems (NeurIPS'19)*. Curran Associates, 2019.

S. Falkner, A. Klein, and F. Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80, pages 1437–1446. Proceedings of Machine Learning Research, 2018.

M. Feurer and F. Hutter. Hyperparameter optimization. In F. Hutter, L. Kotthoff, and J. Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, volume 5 of *The Springer Series on Challenges in Machine Learning*, chapter 1, pages 3–38. Springer, 2019. Available for free at http://automl.org/book.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Proceedings of the 28th International Conference on Advances* *in Neural Information Processing Systems (NeurIPS'15)*, pages 2962–2970. Curran Associates, 2015.

F. Fortin, F. De Rainville, M. Gardner, M. Parizeau, and C. Gagné. DEAP: evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, 2012.

T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. scikit-optimize/scikit-optimize, 2021. URL `https://doi.org/10.5281/zenodo.5565057`.

M. Hoffman, E. Brochu, and N. de Freitas. Portfolio allocation for bayesian optimization. In Cozman F and A. Pfeffer, editors, *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 327–336. AUAI Press, 2011.

F. Hutter, H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009.

F. Hutter, H. Hoos, K. Leyton-Brown, and K. Murphy. Time-bounded sequential parameter optimization. In C. Blum, editor, *Proceedings of the Fourth International Conference on Learning and Intelligent Optimization (LION'10)*, volume 6073 of *Lecture Notes in Computer Science*, pages 281–298. Springer, 2010.

F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In C. Coello, editor, *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization (LION'11)*, volume 6683 of *Lecture Notes in Computer Science*, pages 507–523. Springer, 2011.

F. Hutter, M. Lindauer, A. Balint, S. Bayless, H. Hoos, and K. Leyton-Brown. The configurable SAT solver challenge (CSSC). *Artificial Intelligence*, 243:1–25, 2017.

D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13:455–492, 1998.

K. Kandasamy, K. Vysyaraju, W. Neiswanger, B. Paria, C. Collins, J. Schneider, B. Poczos, and E. Xing. Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with Dragonfly. *Journal of Machine Learning Research*, 21(81): 1–27, 2020.

A. Klein and F. Hutter. Tabular benchmarks for joint architecture and hyperparameter optimization. *arXiv:1905.04970 [cs.LG]*, 2019.

G. Lan, J. Tomczak, D. Roijers, and A. Eiben. Time efficiency in optimization with a bayesian-evolutionary algorithm. *arXiv:2005.04166 [cs.NE]*, 2020.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

Y. Li, Y. Shen, W. Zhang, Y. Chen, H. Jiang, M. Liu, J. Jiang, J. Gao, W. Wu, Z. Yang, C. Zhang, and B Cui. Openbox: A generalized black-box optimization service. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2021*, 2021.

M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, J. Marben, P. Müller, and F. Hutter. BOAH: A Tool Suite for Multi-Fidelity Bayesian Optimization & Analysis of Hyperparameters. *arXiv:1908.06756 [cs.LG]*, 2019a.

M. Lindauer, M. Feurer, K. Eggensperger, A. Biedenkapp, and F. Hutter. Towards assessing the impact of bayesian optimization's own hyperparameters. In P. De Causmaecker, M. Lombardi, and Y. Zhang, editors, *IJCAI 2019 DSO Workshop*, 2019b.

M. López-Ibáñez, J. Dubois-Lacoste, L. Perez Caceres, M. Birattari, and T. Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

I. Loshchilov and F. Hutter. CMA-ES for hyperparameter optimization of deep neural networks. In *International Conference on Learning Representations Workshop track*, 2016. Published online: `iclr.cc`.

J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129), 1978.

L. Nardi, D. Koeplinger, and K. Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

R. Olson, N. Bartley, R. Urbanowicz, and J. Moore. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In T. Friedrich, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'16)*, pages 485–492. ACM, 2016.

J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`, 2018.

M. Rocklin. Dask: Parallel computation with blocked algorithms and task scheduling. In K. Huff and J. Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 130 – 136, 2015.

B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pages 2960–2968. Curran Associates, 2012.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pages 1015–1022. Omnipress, 2010.

W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

C. Thornton, F. Hutter, H. Hoos, and K. Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In I. Dhillon, Y. Koren, R. Ghani, T. Senator, P. Bradley, R. Parekh, J. He, R. Grossman, and R. Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*, pages 847–855. ACM Press, 2013.

R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In H. Escalante and K. Hofmann, editors, *NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 3–26. PMLR, 2020.

A. Zela, T. Elsken, T. Saikia, Y. Marrakchi, T. Brox, and F. Hutter. Understanding and robustifying differentiable architecture search. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*, 2020. Published online: `iclr.cc`.

L. Zimmer, M. Lindauer, and F. Hutter. Auto-Pytorch: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 3079–3090, 2021.