Journal of Machine Learning Research 25 (2024) 1-11 Submitted 4/23; Published 5/24

OpenBox: A Python Toolkit for Generalized Black-box Optimization

Huaijun Jiang[1*] JIANGHUAIJUN@PKU.EDU.CN
Yu Shen[1*] SHENYU@PKU.EDU.CN
Yang Li[2*] THOMASYNGLI@TENCENT.COM
Beicheng Xu[1] BEICHENGXU@STU.PKU.EDU.CN
Sixian Du[1] DUSIXIAN@STU.PKU.EDU.CN
Wentao Zhang[1] WENTAO.ZHANG@PKU.EDU.CN
Ce Zhang[3] CE.ZHANG@ETHZ.CH
Bin Cui[1] BIN.CUI@PKU.EDU.CN

[1] *Key Lab of High Confidence Software Technologies (MOE), School of CS, Peking University, China*
[2] *Department of Data Platform, TEG, Tencent Inc., China*
[3] *Department of Computer Science, ETH Zürich, Switzerland*
[*] *Equal contribution.*

Editor: Zeyi Wen

Abstract

Black-box optimization (BBO) has a broad range of applications, including automatic machine learning, experimental design, and database knob tuning. However, users still face challenges when applying BBO methods to their problems at hand with existing software packages in terms of applicability, performance, and efficiency. This paper presents `OpenBox`, an open-source BBO toolkit with improved usability. It implements user-friendly interfaces and visualization for users to define and manage their tasks. The modular design behind `OpenBox` facilitates its flexible deployment in existing systems. Experimental results demonstrate the effectiveness and efficiency of `OpenBox` over existing systems. The source code of `OpenBox` is available at `https://github.com/PKU-DAIR/open-box`.

Keywords: Python, Black-box Optimization, Bayesian Optimization, Hyper-parameter Optimization

1. Introduction

Black-box optimization (BBO) (Muñoz et al., 2015) deals with optimizing an objective function under a limited budget for function evaluation. However, since the evaluation of objective function is usually expensive, the goal of BBO is to find a configuration that approaches the optimal configuration as soon as possible. Recently, generalized BBO has attracted great attraction in various areas (Foster et al., 2019; Sun et al., 2022b,a; Meng et al., 2023; Huang et al., 2023), which requires more functionalities than traditional BBO tasks. Specifically, traditional BBO refers to BBO with integer or float input parameters and with only a single objective, while there can be various input types (e.g., ordinal and categorical), multiple objectives, and constraints in generalized BBO. Though many software and platforms have been developed for traditional BBO (Hutter et al., 2011; Bergstra et al., 2011; Golovin et al., 2017; Akiba et al., 2019; Balandat et al., 2020; Liu et al., 2022; Lindauer et al., 2022), so far, there is no platform specifically designed to target generalized

BBO. Existing platforms suffer from the following limitations when applied to generalized BBO scenarios: 1) Restricted application scope. Many existing BBO platforms cannot support multiple objectives and constraints, and they cannot support categorical parameters. 2) Unstable performance across problems. Many platforms only implement one or very few BBO algorithms. According to the "no free lunch" theorem (Ho and Pepyne, 2001), this would inevitably lead to unstable performance when applied to various problems. 3) Limited scalability and efficiency. Most platforms execute optimization in a sequential manner, which is inefficient and unscalable.

This paper proposes `OpenBox` to address the above limitations simultaneously. `OpenBox` has the following crucial features: 1) **Generality.** `OpenBox` hosts most state-of-the-art optimization algorithms and is capable of handling BBO tasks with different requirements. 2) **Ease of use.** `OpenBox` provides user-friendly interfaces, visualization, and automatic decisions on algorithms. Users can conveniently define their tasks via either wrapped services or ask-and-tell interfaces. 3) **State-of-the-art performance.** Extensive experiments show that `OpenBox` outperforms the existing systems on a wide range of BBO tasks.

So far, `OpenBox` has helped researchers solve various realistic BBO problems like database tuning (Kanellis et al., 2022; Zhang et al., 2022c) and traffic simulation (Liang et al., 2022). It has also performed as the core part of the open-source graph learning system SGL (Zhang et al., 2022a) and database tuning system DBTune (Zhang et al., 2022b). Besides the academic usages, `OpenBox` won the first place in CIKM 2021 AnalytiCup Track 2 (Jiang et al., 2021), and parts of the toolkit have been successfully deployed in corporations like Tencent (Li et al., 2023) and ByteDance (Shen et al., 2023).

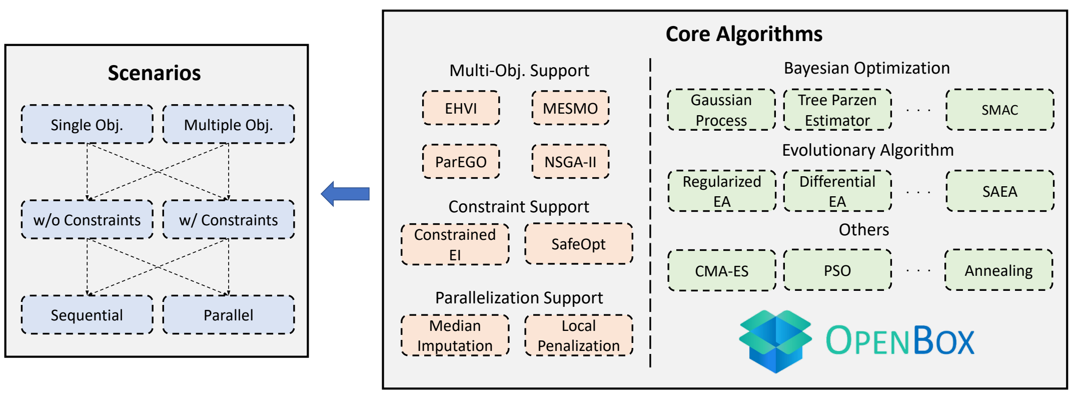


Figure 1: Simplified overview of supported scenarios and built-in algorithms in `OpenBox`.

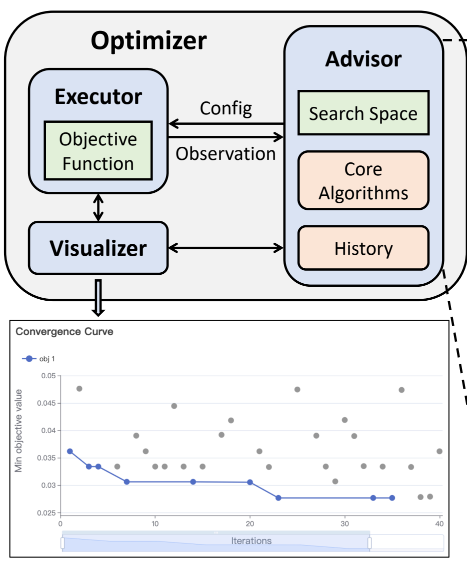
2. Functionality Overview

`OpenBox` implements a wide range of state-of-the-art algorithms for various BBO problems (Figure 1). Concretely, `OpenBox` is capable of handling BBO problems with different numbers of objectives (Belakaria and Deshwal, 2019; Belakaria et al., 2020) and constraints (Sui et al., 2015), and the optimization procedure can either be sequential or parallel (González et al., 2016). To ensure stable performance in different scenarios, `OpenBox` implements variants of Bayesian optimization (Hutter et al., 2011; Bergstra et al., 2011; Snoek et al., 2012; Hou and Behdinan, 2022), evolutionary algorithms (Storn and Price, 1997; Deb et al., 2002; Jin, 2011), and other competitive methods like CMA-ES (Hansen et al., 2003).

While it may be challenging for users to choose the proper algorithm for different scenarios, `OpenBox` also provides automatic decisions on algorithms and settings according to the characteristics of the incoming task, including the search space, number of objectives, and number of constraints. The decisions are based on experimental analysis (Eggensperger et al., 2013) or practical experience. For example, if there are over ten parameters in the input space, or the number of trials exceeds 300, we choose Probabilistic Random Forest (PRF) (Hutter et al., 2011) instead of Gaussian Process (GP) as the surrogate to avoid incompatibility or high computational complexity in Bayesian optimization.

3. System Design

In this section, we will introduce the system design of `OpenBox`. As shown in the upper left corner of Figure 2, the core of the optimization framework is `Optimizer`, which takes an objective function and a search space as inputs, executes the optimization process, and outputs the final results. `Optimizer` consists of three main components: `Advisor`, `Executor`, and `Visualizer`. `Advisor` implements the optimization algorithms described in Section 2 and suggests new configurations based on the provided search space and stored optimization history. `Executor` then runs the objective function on the recommended configurations and returns the results, i.e., observations. To monitor the optimization process, `Visualizer` provides comprehensive visualization APIs for users, including the convergence curve, parameter importance analysis based on SHAP (Lundberg and Lee, 2017), etc. For multi-objective problems, Pareto front and Hypervolume charts are also available. An example of the convergence curve is provided on the bottom left of Figure 2, where the best-observed configurations during optimization are connected by a blue line.



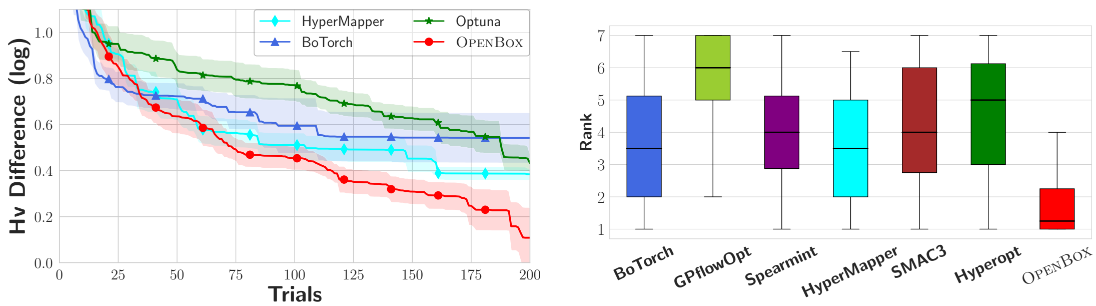

```python
from openbox import space, Advisor, Observation

# Build advisor
search_space = space.Space(...)
advisor = Advisor(search_space)

# Run optimization
for i in range(max_iter):
    # Ask
    config = advisor.get_suggestion()
    # Evaluate
    error_rate = train_and_eval_model(config)
    # Build observation
    obs = Observation(objectives=[error_rate])
    # Tell
    advisor.update_observation(obs)

# Obtain and visualize history
history = advisor.get_history()
history.visualize_html()
```

Figure 2: System overview of `OpenBox`, including the system architecture (upper left), an example of the ask-and-tell interface using Advisor (right), and an example of visualization interfaces (bottom left).



(a) CONSTR (2 objectives, 2 constraints)

(b) LightGBM on 24 OpenML datasets

Figure 3: Performance comparison on constrained multi-objective benchmark CONSTR (left) and LightGBM tuning task (right).

For ease of use, `OpenBox` also provides the ask-and-tell interface using `Advisor`. The example code is shown on the right side of Figure 2. Users first build an `Advisor` based on the search space (Line 4-5). Next, users run optimization by interacting with the `Advisor` (Line 8-16). Concretely, users get a configuration suggestion from `Advisor` (Line 10) and evaluate the performance of the configuration by themselves (Line 12). Users then build an `Observation` that contains the evaluation result (Line 14) and update the `Observation` to `Advisor` (Line 16). Finally, an HTML page for visualization is generated based on the optimization history obtained from `Advisor` (Line 19-20). This ask-and-tell interface is compatible with various types of optimization algorithms and enables users to make flexible modifications to the optimization process.

4. Benchmark

To demonstrate the generality and efficiency of `OpenBox`, we conduct experiments on the constrained multi-objective benchmark CONSTR and the LightGBM (Ke et al., 2017) tuning task on 24 OpenML datasets (Feurer et al., 2021). We report the Hypervolume difference from the optimum in the CONSTR benchmark and the performance rank of the best-achieved accuracy in the LightGBM tuning task. In Figure 3(a), we observe that `OpenBox` outperforms the other baselines in terms of convergence speed and stability. In Figure 3(b), we observe that `OpenBox` outperforms the other competitive systems, achieves a median rank of 1.25, and ranks first in 12 out of 24 datasets.

5. Conclusion

This paper presents `OpenBox`, an open-source system for solving generalized BBO tasks. `OpenBox` hosts a wide range of state-of-the-art optimization algorithms and provides user-friendly interfaces along with comprehensive visualization functions. Evaluations showcase the excellent performance of `OpenBox` over existing systems. The recently released version 0.8.3 has been tested on Linux, macOS, and Windows and can be installed easily via PyPI by '`pip install openbox`'. The source code of `OpenBox` is now available at `https://github.com/PKU-DAIR/open-box`. More detailed examples, APIs, and advanced usages can be found in our documentation[1].

1. `https://open-box.readthedocs.io/`

Acknowledgments

We thank all contributors to this project. This work is supported by the National Natural Science Foundation of China (No. U23B2048 and U22B2037), Beijing Municipal Science and Technology Project (No. Z231100010323002), and High-performance Computing Platform of Peking University. Bin Cui is the corresponding author.

References

Leonel Aguilar Melgar, David Dao, Shaoduo Gan, Nezihe M Gürel, Nora Hollenstein, Jiawei Jiang, Bojan Karlaš, Thomas Lemmin, Tian Li, Yang Li, et al. Ease. ml: A lifecycle management system for machine learning. In *11th Annual Conference on Innovative Data Systems Research (CIDR 2021)(virtual)*. CIDR, 2021.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Omid Azizi, Aqeel Mahesri, Benjamin C Lee, Sanjay J Patel, and Mark Horowitz. Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis. *ACM SIGARCH Computer Architecture News*, 38(3):26–36, 2010.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *NeurIPS*, 2020.

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018.

Syrine Belakaria and Aryan Deshwal. Max-value entropy search for multi-objective bayesian optimization. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10044–10052, 2020.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *arXiv preprint arXiv:1812.06855*, 2018.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Katharina Eggensperger, Matthias Feurer, Frank Hutter, James Bergstra, Jasper Snoek, Holger Hoos, Kevin Leyton-Brown, et al. Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*, volume 10, 2013.

Katharina Eggensperger, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Efficient benchmarking of hyperparameter optimizers via surrogates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

David Eriksson and Matthias Poloczek. Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. PMLR, 2021.

Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018.

Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for bayesian optimization using ranking-weighted gaussian process ensembles. In *AutoML Workshop at ICML*, volume 7, 2018.

Matthias Feurer, Jan N van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. Openml-python: an extensible python api for openml. *Journal of Machine Learning Research*, 22(100):1–5, 2021.

Adam Foster, Martin Jankowiak, Eli Bingham, Paul Horsfall, Yee Whye Teh, Tom Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. *arXiv preprint arXiv:1903.05480*, 2019.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

Pieter Gijsbers, Erin LeDell, Janek Thomas, Sébastien Poirier, Bernd Bischl, and Joaquin Vanschoren. An open source automl benchmark. *arXiv preprint arXiv:1907.00909*, 2019.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and David Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1487–1495, 2017.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2): 577–586, 2020.

Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

José Miguel Hern, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, Zoubin Ghahramani, et al. A general framework for constrained bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(160):1–53, 2016.

José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. PMLR, 2015.

Yu-Chi Ho and David L Pepyne. Simple explanation of the no free lunch theorem of optimization. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 5, pages 4409–4414. IEEE, 2001.

Chun Kit Jeffery Hou and Kamran Behdinan. Dimensionality reduction in surrogate modeling: A review of combined methods. *Data Science and Engineering*, 7(4):402–427, 2022.

Shiyue Huang, Yanzhao Qin, Xinyi Zhang, Yaofeng Tu, Zhongliang Li, and Bin Cui. Survey on performance optimization for database systems. *Science China Information Sciences*, 66(2):121102, 2023.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pages 754–762. PMLR, 2014.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

Mahdi Imani, Seyede Fatemeh Ghoreishi, Douglas Allaire, and Ulisses M Braga-Neto. Mfbo-ssm: Multi-fidelity bayesian optimization for fast inference in state-space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7858–7865, 2019.

Huaijun Jiang, Yu Shen, and Yang Li. Automated hyperparameter optimization challenge at cikm 2021 analyticcup. *arXiv preprint arXiv:2111.00513*, 2021.

Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

Konstantinos Kanellis, Cong Ding, Brian Kroth, Andreas Müller, Carlo Curino, and Shivaram Venkataraman. Llamatune: sample-efficient dbms configuration tuning. *Proceedings of the VLDB Endowment*, 15(11):2953–2965, 2022.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Nicolas Knudde, Joachim van der Herten, Tom Dhaene, and Ivo Couckuyt. GPflowOpt: A Bayesian Optimization Library using TensorFlow. *arXiv preprint – arXiv:1711.03845*, 2017.

Patrick Koch, Oleg Golovidov, Steven Gardner, Brett Wujek, Joshua Griffin, and Yan Xu. Autotune: A derivative-free optimization framework for hyperparameter tuning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 443–452, 2018.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

Yang Li, Jiawei Jiang, Jinyang Gao, Yingxia Shao, Ce Zhang, and Bin Cui. Efficient automatic cash via rising bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4763–4771, 2020.

Yang Li, Yu Shen, Jiawei Jiang, Jinyang Gao, Ce Zhang, and Bin Cui. Mfes-hb: Efficient hyperband with multi-fidelity quality measurements. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8491–8500, 2021a.

Yang Li, Yu Shen, Wentao Zhang, Yuanwei Chen, Huaijun Jiang, Mingchao Liu, Jiawei Jiang, Jinyang Gao, Wentao Wu, Zhi Yang, et al. Openbox: A generalized black-box optimization service. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3209–3219, 2021b.

Yang Li, Huaijun Jiang, Yu Shen, Yide Fang, Xiaofeng Yang, Danqing Huang, Xinyi Zhang, Wentao Zhang, Ce Zhang, Peng Chen, and Bin Cui. Towards general and efficient online tuning for spark. *Proc. VLDB Endow.*, 16(12):3570–3583, 2023.

Chumeng Liang, Zherui Huang, Yicheng Liu, Zhanyu Liu, Guanjie Zheng, Hanyuan Shi, Yuhao Du, Fuliang Li, and Zhenhui Li. Cblab: Scalable traffic simulation with enriched data supporting. *arXiv preprint arXiv:2210.00896*, 2022.

Edo Liberty, Zohar Karnin, Bing Xiang, Laurence Rouesnel, Baris Coskun, Ramesh Nallapati, Julio Delgado, Amir Sadoughi, Yury Astashonok, Piali Das, et al. Elastic machine learning algorithms in amazon sagemaker. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 731–737, 2020.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

Sijia Liu, Parikshit Ram, Deepak Vijaykeerthy, Djallel Bouneffouf, Gregory Bramble, Horst Samulowitz, Dakuo Wang, Andrew Conn, and Alexander Gray. An admm based framework for automl pipeline configuration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4892–4899, 2020.

Yu-Ren Liu, Yi-Qi Hu, Hong Qian, Chao Qian, and Yang Yu. Zoopt: a toolbox for derivative-free optimization. *Science China Information Sciences*, 65(10):207101, 2022.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Xiangfu Meng, Hongjin Huo, Xiaoyan Zhang, Wanchun Wang, and Jinxia Zhu. A survey of personalized news recommendation. *Data Science and Engineering*, 8(4):396–416, 2023.

Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

Mario A Muñoz, Yuan Sun, Michael Kirley, and Saman K Halgamuge. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, 317:224–245, 2015.

ChangYong Oh, Efstratios Gavves, and Max Welling. Bock: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3868–3877. PMLR, 2018.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Valerio Perrone, Michele Donini, Muhammad Bilal Zafar, Robin Schmucker, Krishnaram Kenthapadi, and Cédric Archambeau. Fair bayesian optimization. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 854–863, 2021.

David Salinas, Matthias Seeger, Aaron Klein, Valerio Perrone, Martin Wistuba, and Cedric Archambeau. Syne tune: A library for large scale hyperparameter tuning and reproducible research. In *International Conference on Automated Machine Learning*, pages 16–1. PMLR, 2022.

Rajat Sen, Kirthevasan Kandasamy, and Sanjay Shakkottai. Multi-fidelity black-box optimization with hierarchical partitions. In *International conference on machine learning*, pages 4538–4547. PMLR, 2018.

Yu Shen, Xinyuyang Ren, Yupeng Lu, Huaijun Jiang, Huanyong Xu, Di Peng, Yang Li, Wentao Zhang, and Bin Cui. Rover: An online spark sql tuning service via generalized transfer learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4800–4812, 2023.

Prabhu Teja Sivaprasad, Florian Mai, Thijs Vogels, Martin Jaggi, and François Fleuret. Optimizer benchmarking needs to account for hyperparameter tuning. In *International Conference on Machine Learning*, pages 9036–9045. PMLR, 2020.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Xingyou Song, Sagi Perel, Chansoo Lee, Greg Kochanski, and Daniel Golovin. Open source vizier: Distributed infrastructure and api for reliable and flexible blackbox optimization. In *International Conference on Automated Machine Learning*, pages 8–1. PMLR, 2022.

Rainer Storn and Kenneth Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341, 1997.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International conference on machine learning*, pages 997–1005. PMLR, 2015.

Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, 2022a.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022b.

Dana Van Aken, Andrew Pavlo, Geoffrey J Gordon, and Bohan Zhang. Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1009–1024, 2017.

Zeyi Wen, Jiashuai Shi, Qinbin Li, Bingsheng He, and Jian Chen. Thundersvm: A fast svm library on gpus and cpus. *The Journal of Machine Learning Research*, 19(1):797–801, 2018.

Wentao Zhang, Yu Shen, Zheyu Lin, Yang Li, Xiaosen Li, Wen Ouyang, Yangyu Tao, Zhi Yang, and Bin Cui. Pasca: A graph neural architecture search system under the scalable paradigm. In *Proceedings of the ACM Web Conference 2022*, pages 1817–1828, 2022a.

Xinyi Zhang, Zhuo Chang, Yang Li, Hong Wu, Jian Tan, Feifei Li, and Bin Cui. Facilitating database tuning with hyper-parameter optimization: a comprehensive experimental evaluation. *Proceedings of the VLDB Endowment*, 15(9):1808–1821, 2022b.

Xinyi Zhang, Hong Wu, Yang Li, Jian Tan, Feifei Li, and Bin Cui. Towards dynamic and safe configuration tuning for cloud databases. In *Proceedings of the 2022 International Conference on Management of Data*, pages 631–645, 2022c.