



Universiteit
Leiden
The Netherlands

Tuning the tuner: the art of automatic performance optimization

Willemsen, F.Q.

Citation

Willemsen, F. Q. (2026, April 14). *Tuning the tuner: the art of automatic performance optimization*. Retrieved from <https://hdl.handle.net/1887/4301430>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4301430>

Note: To cite this publication please use the final published version (if applicable).

Bibliography

- [1] M. Abramowitz and I. Stegun. “Bessel Functions of Fractional Order”. In: *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied mathematics series. Dover Publications, 1965, pp. 435–478.
- [2] Z. Alomari, O. E. Halimi, K. Sivaprasad, and C. Pandit. *Comparative studies of six programming languages*. 2015.
- [3] J. Ansel, S. Kamil, K. Veeramachaneni, J. Ragan-Kelley, J. Bosboom, U.-M. O’Reilly, et al. “Opentuner: An extensible framework for program autotuning”. In: *Proceedings of the 23rd international conference on Parallel architectures and compilation*. 2014.
- [4] A. H. Ashouri, W. Killian, J. Cavazos, G. Palermo, and C. Silvano. “A Survey on Compiler Autotuning using Machine Learning”. In: *ACM Computing Surveys* 51.5 (Sept. 30, 2019), pp. 1–42.
- [5] M. Ateeq, H. Habib, A. Umer, and M. U. Rehman. “C++ or python? Which one to begin with: a learner’s perspective”. In: *2014 international conference on teaching and learning in computing and engineering*. IEEEExplore, 2014, pp. 64–69.
- [6] D. Bailey. “Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers”. In: *Supercomputing Review* 4 (June 9, 1991).
- [7] D. H. Bailey. “Misleading performance claims in parallel computations”. In: *Proceedings of the 46th Annual Design Automation Conference*. DAC ’09. New York, NY, USA: Association for Computing Machinery, July 26, 2009, pp. 528–533.
- [8] H. Bal, D. Epema, C. d. Laat, R. v. Nieuwpoort, J. Romein, F. Seinsträ, et al. “A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term”. In: *Computer* 49.5 (2016), pp. 54–63.
- [9] P. Balaprakash, J. Dongarra, T. Gamblin, M. Hall, J. K. Hollingsworth, B. Norris, et al. “Autotuning in High-Performance Computing Applications”. In: *Proceedings of the IEEE* 106.11 (Nov. 2018), pp. 2068–2083.
- [10] P. Balaprakash, S. M. Wild, and P. D. Hovland. “Can search algorithms save large-scale automatic performance tuning?” In: *Procedia Computer Science* 4 (2011), pp. 2136–2145.

Bibliography

- [11] N. Bannur, H. Maheshwari, S. Jain, S. Shetty, S. Merugu, and A. Raval. “Adaptive COVID-19 Forecasting via Bayesian Optimization”. In: *8th ACM IKDD CODS and 26th COMAD*. CODS COMAD 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 432.
- [12] R. Bardenet, M. Brendel, B. Kégl, and M. Sebag. “Collaborative hyperparameter tuning”. In: *International conference on machine learning*. PMLR, 2013, pp. 199–207.
- [13] R. E. Barlow and H. D. Brunk. “The Isotonic Regression Problem and its Dual”. In: *Journal of the American Statistical Association* 67.337 (Mar. 1, 1972), pp. 140–147.
- [14] C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. “Satisfiability Modulo Theories. Frontiers in artificial intelligence and applications, vol. 185, ch. 26”. In: *Handbook of Satisfiability*. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press, 2008, pp. 825–885.
- [15] T. Bartz-Beielstein, C. Doerr, D. v. d. Berg, J. Bossek, S. Chandrasekaran, T. Eftimov, et al. “Benchmarking in Optimization: Best Practice and Open Issues”. In: *arXiv:2007.03488 [cs, math, stat]* (Dec. 16, 2020).
- [16] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. “Cython: The Best of Both Worlds”. In: *Computing in Science & Engineering* 13.2 (Mar. 2011), pp. 31–39.
- [17] V. Beiranvand, W. Hare, and Y. Lucet. “Best practices for comparing optimization algorithms”. In: *Optimization and Engineering* 18.4 (Dec. 1, 2017), pp. 815–848.
- [18] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. “Algorithms for Hyper-Parameter Optimization”. In: *Advances in Neural Information Processing Systems*. NIPS’11. Red Hook, NY, USA: Curran Associates Inc., 2011, pp. 2546–2554.
- [19] J. Bergstra, N. Pinto, and D. Cox. “Machine learning for predictive auto-tuning with boosted regression trees”. In: *2012 Innovative Parallel Computing (InPar)*. 2012 Innovative Parallel Computing (InPar). May 2012, pp. 1–9.
- [20] A. Biere, M. Heule, H. van Maaren, and T. Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*. NLD: IOS Press, Jan. 2009. 980 pp.
- [21] S. C. Billups, S. P. Dirkse, and M. C. Ferris. “A Comparison of Large Scale Mixed Complementarity Problem Solvers”. In: *Computational Issues in High Performance Software for Nonlinear Optimization*. Ed. by A. Murli and G. Toraldo. Boston, MA: Springer US, 1997, pp. 3–25.
- [22] N. Bjørner, L. de Moura, L. Nachmanson, and C. M. Wintersteiger. “Programming Z3”. In: *Engineering Trustworthy Software Systems: 4th International School, SETSS 2018, Chongqing, China, April 7–12, 2018, Tutorial Lectures*. Ed. by J. P. Bowen, Z. Liu, and Z. Zhang. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 148–201.
- [23] S. C. Brailsford, C. N. Potts, and B. M. Smith. “Constraint satisfaction problems: Algorithms and applications”. In: *European Journal of Operational Research* 119.3 (Dec. 16, 1999), pp. 557–581.

- [24] E. Brochu, V. Cora, and N. Freitas. “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning”. In: *CoRR* abs/1012.2599 (2010).
- [25] J. Brownlee. *A note on research methodology and benchmarking optimization algorithms*. Technical Report 70125. Victoria, Australia: Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, 2007.
- [26] W. van Cappellen, T. Oosterloo, M. Verheijen, E. Adams, B. Adebahr, R. Braun, et al. “Aperitif: Phased array feeds for the westerbork synthesis radio telescope-system overview and performance characteristics”. In: *Astronomy & Astrophysics* 658 (2022), A146.
- [27] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, et al. “Rodinia: A benchmark suite for heterogeneous computing”. In: *2009 IEEE International Symposium on Workload Characterization (IISWC)*. 2009 IEEE International Symposium on Workload Characterization (IISWC). Oct. 2009, pp. 44–54.
- [28] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, et al. *Evaluating large language models trained on code*. 2021.
- [29] R. S. Chen and J. K. Hollingsworth. “ANGEL: A Hierarchical Approach to Multi-Objective Online Auto-Tuning”. In: *Proceedings of the 5th International Workshop on Runtime and Operating Systems for Supercomputers - ROSS '15*. the 5th International Workshop. Portland, OR, USA: ACM Press, 2015, pp. 1–8.
- [30] M. Cianfriglia, F. Vella, C. Nugteren, A. Lokhmotov, and G. Fursin. *A model-driven approach for a new generation of adaptive libraries*. 2018.
- [31] C. A. Coello Coello and E. M. Montes. “Constraint-handling in genetic algorithms through the use of dominance-based tournament selection”. In: *Advanced Engineering Informatics* 16.3 (2002), pp. 193–203.
- [32] C. A. Coello Coello. “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art”. In: *Computer Methods in Applied Mechanics and Engineering* 191.11 (2002), pp. 1245–1287.
- [33] C. A. Coello Coello. “Use of a self-adaptive penalty approach for engineering optimization problems”. In: *Computers in Industry* 41.2 (2000), pp. 113–127.
- [34] H. Crowder, R. S. Dembo, and J. M. Mulvey. “On Reporting Computational Experiments with Mathematical Software”. In: *ACM Transactions on Mathematical Software* 5.2 (June 1979), pp. 193–203.

Bibliography

- [35] P. Czarnul and P. Rościszewski. “Auto-tuning methodology for configuration and application parameters of hybrid CPU + GPU parallel systems based on expert knowledge”. In: *2019 International Conference on High Performance Computing & Simulation (HPCS)*. 2019 International Conference on High Performance Computing & Simulation (HPCS). July 2019, pp. 551–558.
- [36] V. Dalibard, M. Schaarschmidt, and E. Yoneki. “BOAT: Building Auto-Tuners with Structured Bayesian Optimization”. In: *Proceedings of the 26th International Conference on World Wide Web. WWW '17*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, pp. 479–488.
- [37] U. Dastgeer, L. Li, and C. Kessler. “Adaptive Implementation Selection in the SkePU Skeleton Programming Library”. In: *Advanced Parallel Processing Technologies*. Ed. by C. Wu and A. Cohen. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 170–183.
- [38] R. I. Davis. “On the Evaluation of Schedulability Tests for Real-Time Scheduling Algorithms”. In: *7th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*. 2016, p. 9.
- [39] E. Daxberger, A. Makarova, M. Turchetta, and A. Krause. “Mixed-Variable Bayesian Optimization”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)* (2020), pp. 2633–2639.
- [40] L. De Moura and N. Bjørner. “Z3: An efficient SMT solver”. In: *International conference on tools and algorithms for the construction and analysis of systems*. Springer, 2008, pp. 337–340.
- [41] K. Deb. “An efficient constraint handling method for genetic algorithms”. In: *Computer Methods in Applied Mechanics and Engineering* 186.2 (2000), pp. 311–338.
- [42] C. Doerr, H. Wang, F. Ye, S. van Rijn, and T. Bäck. “IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics”. In: *arXiv e-prints:1810.05281* (Oct. 2018).
- [43] E. D. Dolan and J. J. Moré. “Benchmarking Optimization Software with Performance Profiles”. In: *Mathematical Programming* 91.2 (Jan. 1, 2002), pp. 201–213.
- [44] T. Dong, V. Dobrev, T. Kolev, R. Rieben, S. Tomov, and J. Dongarra. “A step towards energy efficient computing: Redesigning a hydrodynamic application on CPU-GPU”. In: *2014 IEEE 28th international parallel and distributed processing symposium*. IEEE, 2014, pp. 972–981.
- [45] J. Dorn and M. Girsch. “Genetic operators based on constraint repair”. In: *Proceedings of the ECAI*. Vol. 94. Citeseer, 1994.
- [46] A. Duarte, J. Sánchez-Oro, N. Mladenović, and R. Todosijević. “Variable neighborhood descent”. In: *Handbook of heuristics*. Ed. by R. Martí, P. M. Pardalos, and M. G. C. Resende. Cham: Springer International Publishing, 2018, pp. 341–367.
- [47] T. L. Falch and A. C. Elster. “Machine learning based auto-tuning for enhanced OpenCL performance portability”. In: *2015 IEEE international parallel and distributed processing symposium workshop*. Hyderabad, India: IEEE, May 2015, pp. 1231–1240.

-
- [48] J. Filipovič, J. Hozzová, A. Nezarat, J. Ol'ha, and F. Petrovič. "Using hardware performance counters to speed up autotuning convergence on GPUs". In: *Journal of Parallel and Distributed Computing* 160 (Feb. 2022), pp. 16–35.
 - [49] J. Filipovič, F. Petrovič, and S. Benkner. "Autotuning of OpenCL kernels with global optimizations". In: *Proceedings of the 1st workshop on autotuning and adaptivity approaches for energy efficient HPC systems*. 2017.
 - [50] M. Frigo and S. G. Johnson. "FFTW: An adaptive software architecture for the FFT". In: *Acoustics, speech and signal processing, 1998. Proceedings of the 1998 IEEE international conference on*. Vol. 3. IEEE, 1998, pp. 1381–1384.
 - [51] F. N. Fritsch and J. Butland. "A method for constructing local monotone piecewise cubic interpolants". In: *SIAM Journal on Scientific and Statistical Computing* 5.2 (1984), pp. 300–304.
 - [52] G. Fursin. "Collective knowledge: organizing research projects as a database of reusable components and portable workflows with common interfaces". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2197 (May 17, 2021), p. 20200211.
 - [53] G. Fursin, R. Miceli, A. Lokhmotov, M. Gerndt, M. Baboulin, A. D. Malony, et al. "Collective mind: Towards practical and collaborative auto-tuning". In: *Scientific Programming* 22.4 (Oct. 1, 2014), pp. 309–329.
 - [54] E. C. Garrido-Merchán and D. Hernández-Lobato. "Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes". In: *Neurocomputing* 380 (Mar. 2020), pp. 20–35.
 - [55] M. Gendreau and J.-Y. Potvin. "Tabu search". In: *Search methodologies: introductory tutorials in optimization and decision support techniques*. Springer, 2013, pp. 243–263.
 - [56] A. Georges, D. Buytaert, and L. Eeckhout. "Statistically rigorous java performance evaluation". In: *Proceedings of the 22nd annual ACM SIGPLAN conference on object-oriented programming systems, languages and applications*. OOPSLA '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 57–76.
 - [57] B. W. Goldman and W. F. Punch. "Parameter-less population pyramid". In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. GECCO '14. New York, NY, USA: Association for Computing Machinery, July 12, 2014, pp. 785–792.
 - [58] R. Goncalves, T. van Tilburg, K. Kyzirakos, F. Alvanaki, P. Koutsourakis, B. van Werkhoven, et al. "A spatial column-store to triangulate the netherlands on the fly." In: *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. Gis '16. Burlingame, California and New York, NY, USA: ACM, 2016, 80:1–80:4.
 - [59] N. Gould and J. Scott. "A Note on Performance Profiles for Benchmarking Software". In: *ACM Transactions on Mathematical Software* 43.2 (Aug. 16, 2016), 15:1–15:5.

Bibliography

- [60] J. Guerreiro, A. Ilic, N. Roma, and P. Tomás. “Multi-kernel auto-tuning on GPUs: Performance and energy-aware optimization”. In: *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, 2015, pp. 438–445.
- [61] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. “COCO: a platform for comparing continuous optimizers in a black-box setting”. In: *Optimization Methods and Software* 36.1 (Jan. 2, 2021), pp. 114–144.
- [62] A. Hartono, B. Norris, and P. Sadayappan. “Annotation-based empirical performance tuning using Orio”. In: *2009 IEEE international symposium on parallel distributed processing*. Rome, Italy, IEEE, 2009, pp. 1–11.
- [63] A. B. Hayes, L. Li, D. Chavarria-Miranda, S. L. Song, and E. Z. Zhang. “Orion: A framework for gpu occupancy tuning”. In: *Proceedings of the 17th International Middleware Conference*. 2016, pp. 1–13.
- [64] Q. He and L. Wang. “An effective co-evolutionary particle swarm optimization for constrained engineering design problems”. In: *Engineering applications of artificial intelligence* 20.1 (2007), pp. 89–99.
- [65] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. *SciKit-Optimize*. Sept. 2020.
- [66] S. Heldens, P. Hijma, B. V. Werkhoven, J. Maassen, A. S. Belloum, and R. V. Van Nieuwpoort. “The landscape of exascale research: a data-driven literature analysis”. In: *ACM Computing Surveys (CSUR)* 53.2 (2020), pp. 1–43.
- [67] S. Heldens and B. van Werkhoven. *Kernel Launcher: C++ Library for Optimal-Performance Portable CUDA Applications*. Mar. 22, 2023.
- [68] E. O. Hellsten, A. Souza, J. Lenfers, R. Lacouture, O. Hsu, A. Ejeh, et al. “BaCO: a fast and portable bayesian compiler optimization framework”. In: *Proceedings of the 28th ACM international conference on architectural support for programming languages and operating systems, volume 4*. Asplon ’23. New York, NY, USA: Association for Computing Machinery, Feb. 2024, pp. 19–42.
- [69] H. Heydarian, F. Schueder, M. T. Strauss, B. van Werkhoven, M. Fazel, K. A. Lidke, et al. “Template-free 2D particle fusion in localization microscopy”. In: *Nature methods* 15.10 (Oct. 2018), pp. 781–784.
- [70] P. Hijma, S. Heldens, A. Sclocco, B. Van Werkhoven, and H. E. Bal. “Optimization Techniques for GPU Programming”. In: *ACM Computing Surveys* 55.11 (Nov. 30, 2023), pp. 1–81.
- [71] T. Hoefler and R. Belli. “Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC ’15. New York, NY, USA: Association for Computing Machinery, Nov. 15, 2015, pp. 1–12.

-
- [72] M. Hoffman, E. Brochu, and N. de Freitas. “Portfolio allocation for Bayesian optimization”. In: *Proceedings of the twenty-seventh conference on uncertainty in artificial intelligence*. UAI’11. Barcelona, Spain: AUAI Press, 2011, pp. 327–336.
 - [73] J. N. Hooker. “Testing heuristics: We have it all wrong”. In: *Journal of Heuristics* 1.1 (Sept. 1, 1995), pp. 33–42.
 - [74] V. Horký, P. Libiř, A. Steinhauser, and P. Tůma. “DOs and DON’Ts of conducting performance measurements in Java”. In: *Proceedings of the 6th ACM/SPEC international conference on performance engineering*. ICPE ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 337–340.
 - [75] J. Hozzová, J. O. Tørring, B. van Werkhoven, D. Strelák, and R. Vuduc. “FAIR sharing of data in autotuning research (vision paper)”. In: *Companion of the 15th ACM/SPEC international conference on performance engineering*. ICPE ’24 companion. New York, NY, USA: Association for Computing Machinery, 2024, pp. 21–27.
 - [76] D. Jasrasaria and E. O. Pyzer-Knapp. “Dynamic control of explore/exploit trade-off in bayesian optimization”. In: *Intelligent computing*. Ed. by K. Arai, S. Kapoor, and R. Bhatia. Cham: Springer International Publishing, 2019, pp. 1–15.
 - [77] T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh. “Hyperparameter tuning for big data using Bayesian optimisation”. In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 2574–2579.
 - [78] R. Karlsson, L. Bliet, S. Verwer, and M. d. Weerdt. *Continuous surrogate-based optimization algorithms are well-suited for expensive discrete problems*. 2020.
 - [79] T. Kisuki, P. M. Knijnenburg, and M. F. O’Boyle. “Combined selection of tile sizes and unroll factors using iterative compilation”. In: *The Journal of Supercomputing* 24.1 (2003), pp. 43–67.
 - [80] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, et al. “Preliminary guidelines for empirical research in software engineering”. In: *IEEE Transactions on Software Engineering* 28.8 (Aug. 1, 2002), pp. 721–734.
 - [81] K. Koski, P. Manninen, and T. Nyrönen. “Fifty years of high-performance computing in finland”. In: *Impact of scientific computing on science and society*. Ed. by P. Neittaanmäki and M.-L. Rantalainen. Cham: Springer International Publishing, 2023, pp. 347–355.
 - [82] M. Kraus. *Bayesian Optimization for quicker hyperparameter tuning*. Mar. 2019.
 - [83] H. J. Kushner. “A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise”. In: *Journal of Basic Engineering* 86.1 (Mar. 1964), pp. 97–106.
 - [84] J. Lawson, M. Goli, D. McBain, D. Soutar, and L. Sugy. “Cross-platform performance portability using highly parametrized SYCL kernels”. In: *arXiv preprint arXiv:1904.05347* (2019).
 - [85] Q. Le and A. Smola. “Fastfood: Approximating Kernel Expansions in Loglinear Time”. In: *30th international conference on machine learning (ICML)*. International Conference on Machine Learning (ICML). Google Knowledge, 2013.

Bibliography

- [86] J.-Y. Le Boudec. *Performance Evaluation of Computer and Communication Systems*. Lausanne: EPFL Press, 2010.
- [87] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [88] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-Tzur, M. Hardt, et al. “A system for massively parallel hyperparameter tuning”. In: *Proceedings of machine learning and systems* 2 (2020), pp. 230–246.
- [89] X. Li and G. Agrawal. “Shrinking Sample Search Algorithm for Automatic Tuning of GPU Kernels”. In: *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC). Bengaluru, India: IEEE, Dec. 2021, pp. 262–271.
- [90] Y. Li, J. Dongarra, and S. Tomov. “A note on auto-tuning GEMM for GPUs”. In: *Computational science—ICCS 2009: 9th international conference baton rouge, la, USA, may 25–27, 2009 proceedings, part I* 9. Springer, 2009, pp. 884–892.
- [91] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, et al. “Competition-level code generation with AlphaCode”. In: *Science* 378.6624 (Dec. 2022), pp. 1092–1097.
- [92] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, et al. “A survey on evolutionary constrained multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 27.2 (2022), pp. 201–221.
- [93] R. Lim, B. Norris, and A. Malony. “Autotuning GPU kernels via static and predictive analysis”. In: *2017 46th international conference on parallel processing (ICPP)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017, pp. 523–532.
- [94] M. Liu, Yuan, Xialiang, and Tong Fei. “Evolution of heuristics: Towards efficient automatic algorithm design using large language model”. In: *International conference on machine learning (ICML)*. 2024.
- [95] Y. Liu, W. M. Sid-Lakhdar, O. Marques, X. Zhu, C. Meng, J. W. Demmel, et al. “GPTune: Multitask Learning for Autotuning Exascale Applications”. In: *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. PPoPP ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 234–246.
- [96] D. Lizotte. *Practical Bayesian Optimization*. University of Alberta, 2008.
- [97] G. LLC. *ortools: Google OR-Tools python libraries and modules*. Version 9.7.2996. Sept. 8, 2015.
- [98] P. Luong, S. Gupta, D. Nguyen, S. Rana, and S. Venkatesh. “Bayesian Optimization with Discrete Variables”. In: *Australasian Conference on Artificial Intelligence*. 2019.
- [99] M. Lurati, S. Heldens, A. Sclocco, and B. Van Werkhoven. “Bringing Auto-Tuning to HIP: Analysis of Tuning Impact and Difficulty on AMD and Nvidia GPUs”. In: *Euro-Par 2024: Parallel Processing*. Ed. by J. Carretero, S. Shende, J. Garcia-Blas, I. Brandic, K. Olcoz, and M. Schreiber. Vol. 14801. Cham: Springer Nature Switzerland, 2024, pp. 91–106.

-
- [100] R. Luss, S. Rosset, and M. Shahar. “Efficient Regularized Isotonic Regression with Application to Gene-Gene Interaction Search”. In: *The Annals of Applied Statistics* 6.1 (2012), pp. 253–283.
- [101] J. A. Martinez, E. M. Garzón, A. Plaza, and I. Garcia. “Automatic tuning of iterative computation on heterogeneous multiprocessors with ADITHE”. In: *The Journal of Supercomputing* 58 (2011), pp. 151–159.
- [102] P. Mattson, V. J. Reddi, C. Cheng, C. Coleman, G. Damos, D. Kanter, et al. “MLPerf: An industry standard benchmark suite for machine learning performance”. In: *IEEE Micro* 40.2 (Mar. 2020), pp. 8–16.
- [103] O. Mersmann, M. Preuss, and H. Trautmann. “Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis”. In: *Parallel Problem Solving from Nature, PPSN XI*. Ed. by R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 73–82.
- [104] Z. Michalewicz and G. Nazhiyath. “Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints”. In: *Proceedings of 1995 IEEE international conference on evolutionary computation*. Vol. 2. IEEE, 1995, pp. 647–651.
- [105] G. G. Mitchell, D. O’Donoghue, D. Barnes, and M. McCarville. “GeneRepair - a repair operator for genetic algorithms”. In: *Genetic and evolutionary computation - GECCO 2003*. 2003.
- [106] T. Miyazaki, I. Sato, and N. Shimizu. “Bayesian Optimization of HPC Systems for Energy Efficiency”. In: *High Performance Computing*. Ed. by R. Yokota, M. Weiland, D. Keyes, and C. Trinitis. Cham: Springer International Publishing, 2018, pp. 44–62.
- [107] M. Mlakar, T. Tušar, and B. Filipic. “Comparing Random Forest and Gaussian Process Modeling in the Gp-demo Algorithm”. In: *Information Security Education Journal (ISEJ)* 6.1 (June 2019), p. 9.
- [108] J. Mockus. “On Bayesian Methods for Seeking the Extremum”. In: *Optimization Techniques* (1974), pp. 400–404.
- [109] J. J. Moré and S. M. Wild. “Benchmarking Derivative-Free Optimization Algorithms”. In: *SIAM Journal on Optimization* 20.1 (Jan. 2009), pp. 172–191.
- [110] D. Nan, X. Wei, J. Xu, X. Haoyu, and S. Zhenya. “CESMTuner: An auto-tuning framework for the community earth system model”. In: *2014 IEEE intl conf on high performance computing and communications, 2014 IEEE 6th intl symp on cyberspace safety and security, 2014 IEEE 11th intl conf on embedded software and syst (HPCC, CSS, ICESS)*. IEEE, 2014, pp. 282–289.
- [111] L. Nardi, A. Souza, D. Koeplinger, and K. Olukotun. “HyperMapper: a Practical Design Space Exploration Framework”. In: *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, Oct. 2019, pp. 425–426.

Bibliography

- [112] G. Niemeyer, S. Celles, and F.-J. Willemsen. *Python-Constraint*. Version 2.4.0. July 7, 2005.
- [113] F. Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python*. 2014.
- [114] C. Nugteren and V. Codreanu. “CLTune: a generic auto-tuner for OpenCL kernels”. In: *2015 IEEE 9th international symposium on embedded multicore/many-core systems-on-chip (MC-SoC)*. 2015, pp. 195–202.
- [115] C. Nugteren. “CLBlast”. In: *Proceedings of the International Workshop on OpenCL*. ACM, May 2018.
- [116] P. de Oliveira Castro, E. Petit, J. C. Beyler, and W. Jalby. “ASK: Adaptive Sampling Kit for Performance Characterization”. In: *Euro-Par 2012 Parallel Processing*. Ed. by C. Kaklamanis, T. Papatheodorou, and P. G. Spirakis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 89–101.
- [117] A. V. Papadopoulos, L. Versluis, A. Bauer, N. Herbst, J. v. Kistowski, A. Ali-Eldin, et al. “Methodological Principles for Reproducible Performance Evaluation in Cloud Computing”. In: *IEEE Transactions on Software Engineering* 47.8 (Aug. 1, 2021), pp. 1528–1543.
- [118] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12 (Oct 2011), pp. 2825–2830.
- [119] M. Pelikan. “Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes”. In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. GECCO '08. New York, NY, USA: Association for Computing Machinery, July 12, 2008, pp. 1033–1040.
- [120] F. Petrovič and J. Filipovič. “Kernel tuning toolkit”. In: *SoftwareX* 22 (2023), p. 101385.
- [121] F. Petrovič, D. Střelák, J. Hozzová, J. Ol’ha, R. Trembecký, S. Benkner, et al. “A benchmark set of highly-efficient CUDA and OpenCL kernels and its dynamic autotuning with Kernel Tuning Toolkit”. In: *Future Generation Computer Systems* 108 (July 1, 2020), pp. 161–177.
- [122] R. Pincus, E. J. Mlawer, and J. S. Delamere. “Balancing Accuracy, Efficiency, and Flexibility in Radiation Calculations for Dynamical Models”. In: *Journal of Advances in Modeling Earth Systems* 11.10 (Oct. 2019), pp. 3074–3089.
- [123] D. C. Price, M. A. Clark, B. R. Barsdell, R. Babich, and L. J. Greenhill. “Optimizing performance-per-watt on GPUs in high performance computing”. In: *Computer Science-Research and Development* 31.4 (2016), pp. 185–193.
- [124] W. A. Pruett and R. L. Hester. “The Creation of Surrogate Models for Fast Estimation of Complex Model Outcomes”. In: *PLOS ONE* 11.6 (2016), pp. 1–11.

-
- [125] A. Rasch, M. Haidl, and S. Gorlatch. “ATF: A Generic Auto-Tuning Framework”. In: *Proceedings of the 19th International Conference on High Performance Computing and Communications*. 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). Dec. 2017, pp. 64–71.
- [126] A. Rasch, R. Schulze, M. Steuwer, and S. Gorlatch. “Efficient auto-tuning of parallel programs with interdependent tuning parameters via auto-tuning framework (ATF)”. In: *ACM Trans. Archit. Code Optim.* 18.1 (Jan. 2021).
- [127] C. Rasmussen and C. Williams. “Chapter 4: Covariance Functions”. In: *Gaussian Processes for Machine Learning*. The MIT Press, 2006, pp. 79–104.
- [128] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, and C.-J. Wu. “MLPerf inference benchmark”. In: *2020 ACM/IEEE 47th annual international symposium on computer architecture (ISCA)*. May 2020, pp. 446–459.
- [129] B. Romera-Paredes, M. Barekatin, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, et al. “Mathematical discoveries from program search with large language models”. In: *Nature* 625.7995 (2024), pp. 468–475.
- [130] B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts. “Bayesian Optimisation over Multiple Continuous and Categorical Inputs”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 13, 2020, pp. 8276–8285.
- [131] Y. Rubanova, D. Dohan, K. Swersky, and K. Murphy. “Amortized Bayesian Optimization over Discrete Spaces”. In: *Proceedings of Machine Learning Research / Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)* 124 (2020), pp. 769–778.
- [132] T. P. Runarsson and X. Yao. “Stochastic ranking for constrained evolutionary optimization”. In: *IEEE Transactions on Evolutionary Computation* 4.3 (2000), pp. 284–294.
- [133] S. Ryoo, C. Rodrigues, S. Stone, S. Baghsorkhi, S. Ueng, J. Stratton, et al. “Program optimization space pruning for a multithreaded gpu”. In: *Code generation and optimization. International symposium on*. ACM, 2008, pp. 195–204.
- [134] R. Schoonhoven, B. van Werkhoven, and K. J. Batenburg. “Benchmarking optimization algorithms for auto-tuning GPU kernels”. In: *IEEE Transactions on Evolutionary Computation* (2022), pp. 1–1.
- [135] R. Schulze, S. Gorlatch, and A. Rasch. “pyATF: Constraint-based auto-tuning in python”. In: *Proceedings of the 34th ACM SIGPLAN international conference on compiler construction*. Cc ’25. Las Vegas, NV, USA and New York, NY, USA: Association for Computing Machinery, 2025, pp. 35–47.

Bibliography

- [136] A. Sclocco, H. E. Bal, J. Hessels, J. v. Leeuwen, and R. V. v. Nieuwpoort. “Auto-Tuning Dedispersion for Many-Core Accelerators”. In: *2014 IEEE 28th International Parallel and Distributed Processing Symposium*. 2014 IEEE International Parallel & Distributed Processing Symposium (IPDPS). Phoenix, AZ, USA: IEEE, May 2014, pp. 952–961.
- [137] A. Sclocco, S. Heldens, and B. van Werkhoven. “AMBER: a real-time pipeline for the detection of single pulse astronomical transients”. In: *SoftwareX* 12 (2020), p. 100549.
- [138] C. Seger. “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing”. 2018.
- [139] K. Seymour, H. You, and J. Dongarra. “A comparison of search heuristics for empirical code optimization”. In: *Cluster computing, 2008 IEEE international conference on*. IEEE, 2008, pp. 421–429.
- [140] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. d. Freitas. “Taking the Human Out of the Loop: A Review of Bayesian Optimization”. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175.
- [141] J. Snoek, H. Larochelle, and R. P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. 2012.
- [142] B. Solnik, D. Golovin, G. Kochanski, J. E. Karro, S. Moitra, and D. Sculley. “Bayesian Optimization for a Better Dessert”. In: *Proceedings of the 2017 NIPS Workshop on Bayesian Optimization*. Long Beach, USA, Dec. 9, 2017.
- [143] K. Spafford, J. Meredith, and J. Vetter. “Maestro: Data orchestration and tuning for OpenCL devices”. In: *Euro-par 2010 - parallel processing*. Ed. by P. D’Ambra, M. Guarracino, and D. Talia. Lecture notes in computer science. Springer Berlin Heidelberg, 2010, pp. 275–286.
- [144] M. Stachowski, A. Fiebig, and T. Rauber. “Autotuning based on frequency scaling toward energy efficiency of blockchain algorithms on graphics processing units”. In: *The Journal of Supercomputing* 77.1 (Jan. 1, 2021), pp. 263–291.
- [145] M. L. Stein. “Matern class”. In: *Interpolation of Spatial Data*. Springer New York, 1999, pp. 31–33.
- [146] N. van Stein and T. Bäck. *LLaMEA: a large language model evolutionary algorithm for automatically generating metaheuristics*. 2025.
- [147] N. van Stein, A. V. Kononova, H. Yin, and T. Bäck. “BLADE: Benchmark suite for LLM-driven automated design and evolution of iterative optimisation heuristics”. In: *GECCO ’25 companion*. Proceedings of the genetic and evolutionary computation conference companion. Association for Computing Machinery, 2025.
- [148] N. van Stein, D. Vermetten, and T. Bäck. “In-the-loop hyper-parameter optimization for LLM-based automated design of heuristics”. In: *ACM Trans. Evol. Learn. Optim.* (Apr. 2025).

-
- [149] J. A. Stratton, C. Rodrigues, I.-J. Sung, N. Obeid, L.-W. Chang, N. Anssari, et al. “Parboil: A Revised Benchmark Suite for Scientific and Commercial Throughput Computing”. In: *Center for Reliable and High-Performance Computing* 127 (Mar. 2, 2012), p. 27.
- [150] P. D. Surry, N. J. Radcliffe, and I. D. Boyd. “A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method”. In: *AISB workshop on evolutionary computing*. Springer, 1995, pp. 166–180.
- [151] L. P. Swiler, P. D. Hough, P. Qian, X. Xu, C. Storlie, and H. Lee. “Surrogate Models for Mixed Discrete-Continuous Variables”. In: *Constraint Programming and Decision Making* (2014), pp. 181–202.
- [152] C. Tapus, I.-H. Chung, and J. Hollingsworth. “Active harmony: Towards automated performance tuning”. In: *Proceedings from the conference on high performance networking and computing*. Proceedings from the Conference on High Performance Networking and Computing. Dec. 2002, pp. 44–44.
- [153] P. Team. *PySMT: A solver-agnostic library for SMT Formulae manipulation and solving*. Version 0.9.5. May 28, 2022.
- [154] TechPowerUp. *NVIDIA A100 SXM4 40 GB*. May 2020.
- [155] TechPowerUp. *NVIDIA GeForce GTX TITAN X specifications*. Mar. 2015.
- [156] TechPowerUp. *NVIDIA GeForce RTX 2070 SUPER specifications*. July 2019.
- [157] B. Tessema and G. G. Yen. “A self adaptive penalty function based algorithm for constrained optimization”. In: *2006 IEEE international conference on evolutionary computation*. IEEE, 2006, pp. 246–253.
- [158] J. O. Tørring and A. C. Elster. “Analyzing Search Techniques for Autotuning Image-based GPU Kernels: The Impact of Sample Sizes”. In: *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). May 2022, pp. 972–981.
- [159] J. O. Tørring, B. van Werkhoven, F. Petrovč, F.-J. Willemsen, J. Filipović, and A. C. Elster. “Towards a benchmarking suite for kernel tuners”. In: *2023 IEEE international parallel and distributed processing symposium workshops (IPDPSW)*. IEEE, 2023, pp. 724–733.
- [160] C. C. Van Heerwaarden, B. J. Van Stratum, T. Heus, J. A. Gibbs, E. Fedorovich, and J. P. Mel-lado. “MicroHH 1.0: A computational fluid dynamics code for direct numerical simulation and large-eddy simulation of atmospheric boundary layer flows”. In: *Geoscientific Model Development* 10.8 (2017), pp. 3145–3165.
- [161] P. J. Van Laarhoven and E. H. Aarts. “Simulated annealing”. In: *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.
- [162] S. Venkatraman and G. G. Yen. “A generic framework for constrained optimization using genetic algorithms”. In: *IEEE Transactions on Evolutionary Computation* 9.4 (2005), pp. 424–435.

Bibliography

- [163] Y. Wang and B. Vinter. “Auto-tuning for large-scale image processing by dynamic analysis method on multicore platforms”. In: *International Journal of Embedded Systems* 8.4 (2016), pp. 313–322.
- [164] N. Weber and M. Goesele. “MATOG: Array Layout Auto-Tuning for CUDA”. In: *ACM Transactions on Architecture and Code Optimization* 14.3 (Aug. 30, 2017), 28:1–28:26.
- [165] B. van Werkhoven. “Kernel Tuner: A search-optimizing GPU code auto-tuner”. In: *Future Generation Computer Systems* 90 (2019), pp. 347–358.
- [166] B. van Werkhoven, J. Maassen, H. E. Bal, and F. J. Seinstra. “Optimizing convolution operations on GPUs using adaptive tiling”. In: *Future Generation Computer Systems* 30 (Jan. 2014), pp. 14–26.
- [167] B. van Werkhoven, W. J. Palenstijn, and A. Sclocco. “Lessons learned in a decade of research software engineering GPU applications”. In: *Computational science – ICCS 2020*. 2020, pp. 399–412.
- [168] R. C. Whaley, A. Petitet, and J. J. Dongarra. “Automated empirical optimizations of software and the ATLAS project”. In: *Parallel Computing* 27.1 (2001), pp. 3–35.
- [169] F.-J. Willemsen, S. Heldens, R. V. van Nieuwpoort, and B. van Werkhoven. “Constraint-aware Optimization in Auto-Tuning”. In: *2026 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. New Orleans, LA, USA, 2026.
- [170] F.-J. Willemsen, R. van Nieuwpoort, and B. van Werkhoven. “Bayesian Optimization for auto-tuning GPU kernels”. In: *2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS) at SuperComputing 2021*. 2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS) at SuperComputing 2021. Nov. 2021, pp. 106–117.
- [171] F.-J. Willemsen, R. V. van Nieuwpoort, and B. van Werkhoven. “Efficient construction of large search spaces for auto-tuning”. In: *Proceedings of the 54th international conference on parallel processing (ICPP '25)*. San Diego, CA, USA and New York, NY, USA: Association for Computing Machinery, Sept. 2025.
- [172] F.-J. Willemsen, R. V. van Nieuwpoort, and B. van Werkhoven. “Tuning the Tuner: Introducing Hyperparameter Optimization for Auto-Tuning”. In: *Proceedings of the 2025 IEEE eScience conference*. 21st IEEE International Conference on eScience. Chicago, IL, USA, Sept. 2025.
- [173] F.-J. Willemsen, R. Schoonhoven, J. Filipović, J. O. Tørring, R. van Nieuwpoort, and B. van Werkhoven. “A methodology for comparing optimization algorithms for auto-tuning”. In: *Future Generation Computer Systems* 159 (2024), pp. 489–504.
- [174] F.-J. Willemsen, N. van Stein, and B. van Werkhoven. “Automated Algorithm Design for Auto-Tuning Optimizers”. In: *Proceedings of Machine Learning and Systems*. MLSys 2026. Seattle, WA, USA, 2026.

- [175] X. Wu, P. Balaprakash, M. Kruse, J. Koo, B. Videau, P. Hovland, et al. “ytop: Autotuning Scientific Applications for Energy Efficiency at Large Scales”. In: *Concurrency and Computation: Practice and Experience* (Oct. 30, 2024), e8322.
- [176] X. Wu, M. Kruse, P. Balaprakash, H. Finkel, P. Hovland, V. Taylor, et al. “Autotuning Poly-Bench Benchmarks with LLVM Clang/Polly Loop Optimization Pragmas Using Bayesian Optimization”. In: *2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. 2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS). Nov. 2020, pp. 61–70.
- [177] Yadolah Dodge. “Central Limit Theorem”. In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer, 2008, pp. 66–68.
- [178] X.-S. Yang. *Nature-inspired optimization algorithms*. Academic Press, 2020.
- [179] H. Ye, J. Wang, Z. Cao, F. Berto, C. Hua, H. Kim, et al. “Reevo: Large language models as hyper-heuristics with reflective evolution”. In: *Advances in neural information processing systems* 37 (2024), pp. 43571–43608.
- [180] F. Zehra, M. Javed, D. Khan, and M. Pasha. *Comparative analysis of C++ and python in terms of memory and time*. 2020.