A Retrospective on the CP 2006 paper "Performance Prediction and Automated Tuning of Randomized and Parametric Algorithms"

Frank Hutter
University of Freiburg

Youssef Hamadi
Uber AI

Holger Hoos
Leiden University and
University of British Columbia

Kevin Leyton-Brown
University of British Columbia

Over the past decades, a considerable body of work has shown how to use supervised machine learning methods to build regression models that can predict the running time of black-box algorithms based on observed performance data. Such *empirical performance models (EPMs)* are useful in many practical contexts:

Algorithm selection. One widely adopted approach to the classic problem of selecting the best from a given set of algorithms on a per-instance basis [Rice, 1976] is to use EPMs to predict the performance of all candidate algorithms and select the one predicted to perform best [see, e.g., Brewer, 1995; Allen and Minton, 1996; Lobjois and Lemaître, 1998; Fink, 1998; Howe *et al.*, 2000; Nudelman *et al.*, 2003; Roberts and Howe, 2007; Xu *et al.*, 2008; Kotthoff *et al.*, 2012]

Parameter tuning and algorithm configuration. EPMs can model the performance of a parameterized algorithm as a function of its parameters; this is useful for sequential model-based optimization, which alternates between learning an EPM and using it to identify promising settings to evaluate next [see, e.g., Jones *et al.*, 1998; Bartz-Beielstein *et al.*, 2005; Hutter *et al.*, 2011; Arbelaez *et al.*, 2012]. EPMs can also model algorithm performance as a function of both problem instance features and algorithm parameter settings; such models can then be used to select parameter settings with good predicted performance on a per-instance basis [Hutter and Hamadi, 2005; Hutter *et al.*, 2006].

Generating hard benchmarks. An EPM can be used to identify parameter values for an instance generator that lead to benchmark instances that are hard for one or more given algorithms, and thus facilitate the improvement of algorithm performance [Leyton-Brown *et al.*, 2003, 2009].

Gaining insights into instance hardness and algorithm performance. EPMs can be used to assess which instance features and algorithm parameter values most impact empirical performance. Some models support such assessments directly [see, e.g., Ridge and Kudenko, 2007; Mersmann *et al.*, 2013; Hutter *et al.*, 2014a]. For other models, generic feature selection methods, such as forward selection, can be used to identify a small number of key model inputs that explain algorithm performance almost as well as the entire set [see, e.g., Leyton-Brown *et al.*, 2009; Hutter *et al.*, 2013].

The main contribution of our 2006 CP paper *Performance Prediction and Automated Tuning of Randomized and Parametric Algorithms* was to extend EPMs to:

- **Randomized Algorithms.** We demonstrated that EPMs can also make surprisingly accurate predictions of the runtime distributions of incomplete and randomized search methods, such as stochastic local search algorithms.
- **Parametric Algorithms.** We showed for the first time how information about an algorithm's parameter settings can be incorporated into an EPM, and how EPMs can be used to automatically adjust algorithm parameters on a per-instance basis in order to optimize performance.

An empirical analysis for Novelty+ [Hoos, 2002] and SAPS [Hutter *et al.*, 2002] on structured and unstructured SAT instances showed very good predictive performance, as well as significant speedups of our automatically determined parameter settings, when compared to the default and best fixed distribution-specific parameter settings.

Following our 2006 paper, we worked on many of the aforementioned applications of EPMs, especially algorithm selection (SATzilla [Xu *et al.*, 2008]) and algorithm configuration (SMAC [Hutter *et al.*, 2011]). We subsequently published a comprehensive article in AIJ [Hutter *et al.*, 2014b] that presented further methodological advancements on EPMs:

- **More sophisticated modeling techniques.** Random forests turned out to work particularly well for predictions based on a large number of instance features and (both categorical and continuous) algorithm parameters.
- **New instance features.** We introduced a comprehensive set of features for propositional satisfiability (SAT), the travelling salesman problem (TSP) and mixed integer programming (MIP) problems—in particular, novel probing and timing features.
- **Techniques from the statistical literature on survival analysis.** These offer better ways to handle data from runs that were prematurely terminated (censored runs).

We look forward to more exciting work in the years to come on building better EPMs and leveraging them in practice.

References

J. A. Allen and S. Minton. Selecting the right heuristic algorithm: Runtime performance predictors. In Gordon I. McCalla, editor, *Advances in Artificial Intelligence, 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI '96, Toronto, Ontario, Canada, May 21-24, 1996, Proceedings*, volume 1081 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 1996.

A. Arbelaez, Y. Hamadi, and M. Sebag. Continuous search in constraint programming. In Youssef Hamadi, Eric Monfroy, and Frédéric Saubion, editors, *Autonomous Search*, pages 219–243. Springer, 2012.

T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (CEC'05)*, pages 773–780, 2005.

E. A. Brewer. High-level optimization via automated statistical modeling. In *Proceedings of the 5th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming (PPOPP-95)*, pages 80–91, 1995.

E. Fink. How to solve it automatically: Selection among problem-solving methods. In *Proceedings of the Fourth International Conference on AI Planning Systems*, pages 128–136. AAAI Press, 1998.

H. H. Hoos. An adaptive noise mechanism for WalkSAT. In R. Dechter, M. Kearns, and R. Sutton, editors, *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, pages 655–660. AAAI Press / The MIT Press, Menlo Park, CA, USA, 2002.

A. E. Howe, E. Dahlman, C. Hansen, M. Scheetz, and A. Mayrhauser. Exploiting competitive planner performance. In Susanne Biundo and Maria Fox, editors, *Recent Advances in AI Planning (ECP'99)*, volume 1809 of *Lecture Notes in Computer Science*, pages 62–72. Springer Berlin Heidelberg, 2000.

F. Hutter and Y. Hamadi. Parameter adjustment based on performance prediction: Towards an instance-aware problem solver. Technical Report MSR-TR-2005-125, January 2005.

F. Hutter, D. A. D. Tompkins, and H. H. Hoos. Scaling and probabilistic smoothing: efficient dynamic local search for SAT. In *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming (CP'02)*, volume 2470 of *LNCS*, pages 233–248. Springer-Verlag, 2002.

F. Hutter, Y. Hamadi, H. H. Hoos, and K. Leyton-Brown. Performance prediction and automated tuning of randomized and parametric algorithms. In *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP'06)*, volume 4204 of *LNCS*, pages 213–228. Springer-Verlag, 2006.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th Learning and Intelligent Optimization Conference (LION'11)*, volume 6683 of *LNCS*, pages 507–523. Springer-Verlag, 2011.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Identifying key algorithm parameters and instance features using forward selection. In *Proceedings of the 7th Learning and Intelligent Optimization Conference (LION'13)*, 2013.

F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proceedings of International Conference on Machine Learning 2014 (ICML 2014)*, pages 754–762, June 2014.

F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206(0):79–111, January 2014.

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

L. Kotthoff, I. P. Gent, and I. Miguel. An evaluation of machine learning in algorithm selection for search problems. *AI Communications*, 25(3):257–270, 2012.

K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. Boosting as a metaphor for algorithm design. In F. Rossi, editor, *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03)*, volume 2833 of *LNCS*, pages 899–903. Springer-Verlag, 2003.

Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. Empirical hardness models: methodology and a case study on combinatorial auctions. *Journal of the ACM*, 56(4):1–52, 2009.

L. Lobjois and M. Lemaître. Branch and bound algorithm selection by performance prediction. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, pages 353–358, 1998.

O. Mersmann, B. Bischl, H. Trautmann, M. Wagner, J. Bossek, and F. Neumann. A novel feature-based approach to characterize algorithm performance for the traveling salesperson problem. *Annals of Mathematics and Artificial Intelligence (AMAI)*, pages 32 pages, published online: 28 March 2013, 2013.

E. Nudelman, K. Leyton-Brown, G. Andrew, C. Gomes, J. McFadden, B. Selman, and Y. Shoham. Satzilla 0.9. Solver description, 2003 SAT Competition, 2003.

J. R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

E. Ridge and D. Kudenko. Tuning the performance of the MMAS heuristic. In *Proceedings of the International Workshop on Engineering Stochastic Local Search Algorithms (SLS'2007)*, volume 4638 of *LNCS*, pages 46–60. Springer-Verlag, 2007.

Mark Roberts and Adele Howe. Learned models of performance for many planners. In *ICAPS 2007 Workshop AI Planning and Learning*, 2007.

L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. SATzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32:565–606, June 2008.