Open Algorithm Selection Challenge 2017 Setup and Scenarios

Marius Lindauer LINDAUER@CS.UNI-FREIBURG.DE
Jan N. van Rijn VANRIJN@CS.UNI-FREIBURG.DE
Department of Computer Science, University of Freiburg, Germany

Lars Kotthoff LARSKO@UWYO.EDU
Department of Computer Science, University of Wyoming, USA

Editors: Marius Lindauer, Jan N. van Rijn and Lars Kotthoff

Abstract

The 2017 algorithm selection challenge provided a snapshot of the state of the art in algorithm selection and garnered submissions from four teams. In this chapter, we describe the setup of the challenge and the algorithm scenarios that were used.

Keywords: Algorithm Selection, Competition

1. Introduction

In many areas of AI, tremendous advances have been achieved in the last decades. Approaches often leverage problem specific characteristics for high performance; they specialize to particular problem instances. One way of leveraging this complementarity between algorithms are algorithm portfolios combined with a selector that chooses the best algorithm for a given instance – the algorithm selection problem (Rice, 1976). Formally, given a portfolio of algorithms $\mathcal{A} \in \mathcal{P}$, a set of instances $\mathcal{I}$ and a performance metric $m : \mathcal{P} \times \mathcal{I} \to \mathbb{R}$ (e.g., runtime), the algorithm selection problem is to determine a mapping $s : \mathcal{I} \to \mathcal{P}$ from an instance $i \in \mathcal{I}$ to an algorithm $\mathcal{A} \in \mathcal{P}$ such that the performance across all instances is maximized (w.l.o.g):

$$\max_s \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} m(s(i), i) \tag{1}$$

A common approach to the algorithm selection problem is to characterize the instances by instance features (Nudelman et al., 2004; Brazdil et al., 2008; Xu et al., 2011; Hoos et al., 2014; Fawcett et al., 2014) and use machine learning to learn the mapping s.

Many different algorithm selection systems have been proposed since. Modifications of the original algorithm selection problem include for example the use of pre-solving schedules (Xu et al., 2008), per-instance algorithm schedules (Amadini et al., 2014b) or parallel portfolio selection (Lindauer et al., 2015). In addition, different machine learning approaches to learn s were proposed, e.g. regression models (Nudelman et al., 2004; Xu et al., 2008), k-nearest neighbor (Leite and Brazdil, 2005; Kadioglu et al., 2011; van Rijn et al., 2015a), pair-wise cost-sensitive random forests (Xu et al., 2011), stacked models (Kotthoff, 2012;

© 2017 M. Lindauer, J.N. van Rijn & L. Kotthoff.

Samulowitz et al., 2013) and dynamic portfolios (van Rijn et al., 2015b). For a thorough overview on algorithm selection, we refer the interested reader to Kotthoff (2014).

ASlib (Bischl et al., 2016) is a benchmark library for algorithm selection that collects data from the literature to provide a reproducible way of comparing different approaches and evaluating the state of the art. It enabled the first challenge on algorithm selection in 2015 (Kotthoff et al., 2017), and the open algorithm selection challenge (OASC) in 2017. In this document, we describe the setup of the OASC and the way we selected a new set of interesting algorithm selection benchmarks for it.

2. Setup

The series of algorithm selection challenges is built upon the algorithm selection library (ASlib; Bischl et al. (2016)). An ASlib *scenario* contains pre-computed results for a portfolio of algorithms on a set of instances (e.g., a SAT instance or a Machine Learning dataset); i.e. $m(\mathcal{A}, i)$ is known for all pairs of $\mathcal{A} \in \mathcal{P}$ and $i \in \mathcal{I}$. Furthermore for each instance, a set of pre-computed instance features are available, as well as the time required to compute the feature values. Having access to this data (and additional meta-data such as cutoff times) enables algorithm selection researchers to perform reproducible comparisons of approaches. ASlib distinguishes between two types of scenarios: *runtime scenarios* and *quality scenarios*. In runtime scenarios the goal is to minimize the time to select an algorithm that solves all instances (e.g. SAT, TSP), whereas in quality scenarios the goal is to find the algorithm that obtains the highest score according to some metric (e.g. Machine Learning). The main practical difference between the two types of scenarios is that the cost of feature computation adds overhead in the former, but not in the latter case. Part of the data for a scenario was given to participants to train their approaches on and another part was held out to enable verification and a fair comparison.

The main differences between the 2017 and the 2015 algorithm selection challenges are as follows:

1. In 2015, all scenarios were known, but the splits into training and test instances were unknown. In 2017, the participants had access to performance and feature data on training instances (2/3), and only the instance features for the test instances (1/3).

2. In addition, new scenarios that had not been published as part of ASlib before were included in the 2017 evaluation. All scenarios were obfuscated by replacing scenario, algorithm, instance, and feature names and multiplying all performance and feature values by 4.

3. In 2015, the participants submitted their algorithm selection systems which were trained and run by the organizers. In 2017, participants submitted their predictions for the test set.

4. The 2017 challenge allowed arbitrary schedules of feature computation and algorithm steps.

5. In 2017, each team was allowed a maximum of two submissions.

We used the *closed gap* metric to compare the performance of algorithm selection systems across different scenarios, as in the 2015 challenge. Given the optimal performance of the virtual best solver (oracle) m_{VBS}, the baseline performance of always selecting the best average solver m_{SBS} (single best solver), and the algorithm selection system at hand m_s, the closed gap for an algorithm selection benchmark is defined as:

$$\frac{m_{\text{SBS}} - m_s}{m_{\text{SBS}} - m_{\text{VBS}}} \tag{2}$$

In this metric, 1.0 corresponds to a perfect score (i.e. the algorithm selection system always selects the best algorithm for each instance and does not generate overhead due to instance feature computation) and 0.0 corresponds to the baseline (i.e. the single best solver). A value of less than 0.0 indicates that the algorithm selection system is worse than the single best solver, i.e. chooses algorithms that perform worse than it.[1]

3. Algorithm Selection Scenarios

Apart from giving the snapshot of the state of the art in algorithm selection at the time, the algorithm selection challenge is also an opportunity to collect new scenarios for ASlib. We build new scenarios for recent competitions, the CSP Minizinc Competition (Stuckey et al., 2014), the MaxSAT Evaluation (Argelich et al., 2008), Mittelmann's annual MIPLib evaluation (Koch et al., 2011), the QBF evaluation (Pulina, 2016), and `SAT03-16_INDU`, which covers the international SAT competition from 2003 to 2016 (Balyo et al., 2017). The instance features for all of these scenarios were computed with publicly-available software (BNSL (Malone et al., 2018), CSP (Amadini et al., 2014a), Machine Learning (Pfahringer et al., 2000; Sun and Pfahringer, 2013; van Rijn, 2016), MaxSAT (Ansótegui et al., 2016), MIP (Leyton-Brown et al., 2009), QBF (Pulina and Tacchella, 2010), SAT (Xu et al., 2008; Alfonso and Manthey, 2014) and TTP (Wagner et al., 2017)). The runtimes of the algorithms and the feature computation costs were not measured on the same hardware. However, since the feature costs are typically quite small compared to the algorithm runtimes, the estimation of an algorithm selection system's performance should be quite close to its real performance.

Table 1 shows the variety of different algorithm selection scenarios we used. We collected scenarios from 8 application different domains and with different characteristics, ranging across different numbers of algorithms (5-31) and instances (100-9720). The 2017 challenge included scenarios with solution quality as the performance metric for the first time.

To study the effect of small changes between scenarios, we included two pairs of very similar scenarios. `CSP-Minizinc-Obj-2016` and `CSP-Minizinc-Time-2016` consider the same algorithms, instances and features, but the performance metric is different. In `MAXSAT-PMS-2016` and `MAXSAT-WPMS-2016`, the algorithms and instances are different, but the features are the same and the instances are typically considered to be quite similar.

The remainder of this volume gives the descriptions of the submissions.

1. The full rules and submission instructions are available at http://www.coseal.net/open-algorithm-selection-challenge-2017-oasc/.

Scenario	Alias	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{F}\rvert$	F-Costs	Objective	Speedup
BNSL-2016*	Bado	8	1179	86	✓	Time	41
CSP-Minizinc-Obj-2016	Camilla	8	100	95	×	Quality	n/a
CSP-Minizinc-Time-2016	Caren	8	100	95	✓	Time	61
MAXSAT-PMS-2016	Magnus	19	601	37	✓	Time	25
MAXSAT-WPMS-2016	Monty	18	630	37	✓	Time	16
MIP-2016	Mira	5	218	143	✓	Time	11
OPENML-WEKA-2017	Oberon	30	105	103	×	Quality	n/a
QBF-2016	Qill	24	825	46	✓	Time	265
SAT12-ALL*	Svea	31	1614	115	✓	Time	30
SAT03-16_INDU	Sora	10	2000	483	✓	Time	13
TTP-2016*	Titus	22	9720	50	×	Quality	n/a

Table 1: Overview of algorithm selection scenarios used in 2017 showing the alias in the competition, the number of algorithms $|\mathcal{A}|$, the number of instances $|\mathcal{I}|$, the number of instance features $|\mathcal{F}|$, whether costs for feature computation are available (F-Costs), the performance objective and for runtime scenarios, the speedup of the virtual best solver (VBS) over the single best solver ($m_{\text{SBS}}/m_{\text{VBS}}$). Scenarios marked with an asterisk were available in ASlib before the challenge.

Acknowledgements

We thank Rolf-David Bergdoll for collecting the data for the new algorithm selection benchmarks. M. Lindauer acknowledges funding by the DFG (German Research Foundation) under Emmy Noether grant HU 1900/2-1. J. N. van Rijn acknowledges funding by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant no. 716721. Lars Kotthoff thanks Marius for including him.

References

E. Alfonso and N. Manthey. New CNF features and formula classification. In D. Le Berre, editor, *Proceedings of Fifth Pragmatics of SAT workshop (POS)*, volume 27 of *EPiC Series in Computing*, pages 57–71, 2014.

R. Amadini, M. Gabbrielli, and J. Mauro. An enhanced features extractor for a portfolio of constraint solvers. In Y. Cho, S. Shin, S. Kim, C. Hung, and J. Hong, editors, *Proceedings of Symposium on Applied Computing*, pages 1357–1359. ACM, 2014a.

R. Amadini, M. Gabbrielli, and J. Mauro. SUNNY: a lazy portfolio approach for constraint solving. *Theory and Practice of Logic Programming*, 14(4-5):509–524, 2014b.

C. Ansótegui, J. Gabàs, Y. Malitsky, and M. Sellmann. Maxsat by improved instance-specific algorithm configuration. *Artifical Intelligence*, 235:26–39, 2016.

J. Argelich, C. Min Li, F. Manyà, and J. Planes. The first and second max-sat evaluations. *JSAT*, 4(2-4):251–278, 2008.

T. Balyo, M. Heule, and M. Järvisalo. SAT competition 2016: Recent developments. In S.Singh and S. Markovitch, editors, *Proceedings of the Conference on Artificial Intelligence (AAAI'17)*, pages 5061–5063. AAAI Press, 2017.

B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Frechétte, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, and J. Vanschoren. ASlib: A benchmark library for algorithm selection. *Artificial Intelligence*, pages 41–58, 2016.

P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to Data Mining*. Springer Publishing Company, Incorporated, 1 edition, 2008.

C. Fawcett, M. Vallati, F. Hutter, J. Hoffmann, H. Hoos, and K. Leyton-Brown. Improved features for runtime prediction of domain-independent planners. In S. Chien, D. Minh, A. Fern, and W. Ruml, editors, *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS-14)*. AAAI, 2014.

H. Hoos, M. Lindauer, and T. Schaub. claspfolio 2: Advances in algorithm selection for answer set programming. *Theory and Practice of Logic Programming*, 14:569–585, 2014.

S. Kadioglu, Y. Malitsky, A. Sabharwal, H. Samulowitz, and M. Sellmann. Algorithm selection and scheduling. In J. Lee, editor, *Proceedings of the Seventeenth International Conference on Principles and Practice of Constraint Programming (CP'11)*, volume 6876 of *Lecture Notes in Computer Science*, pages 454–469. Springer-Verlag, 2011.

T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. Bixby, E. Danna, G. Gamrath, A. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. Steffy, and K. Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.

L. Kotthoff. Hybrid Regression-Classification Models for Algorithm Selection. In *20th European Conference on Artificial Intelligence*, pages 480–485, August 2012.

L. Kotthoff. Algorithm selection for combinatorial search problems: A survey. *AI Magazine*, 35(3):48–60, 2014.

L. Kotthoff, B. Hurley, and B. O'Sullivan. The ICON challenge on algorithm selection. *AI Magazine*, 38(2):91–93, 2017.

R. Leite and P. Brazdil. Predicting Relative Performance of Classifiers from Samples. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, pages 497–503. ACM, 2005.

K. Leyton-Brown, E. Nudelman, and Y. Shoham. Empirical hardness models: Methodology and a case study on combinatorial auctions. *Journal of the ACM*, 56(4), 2009.

M. Lindauer, H. Hoos, and F. Hutter. From sequential algorithm selection to parallel portfolio selection. In C. Dhaenens, L. Jourdan, and M. Marmion, editors, *Proceedings of the Nineth International Conference on Learning and Intelligent Optimization (LION'15)*, Lecture Notes in Computer Science, pages 1–16. Springer-Verlag, 2015.

B. Malone, K. Kangas, M. Järvisalo, M. Koivisto, and P. Myllymäki. Empirical hardness of finding optimal bayesian network structures: Algorithm selection and runtime prediction. *Machine Learning*, 2018. to appear.

E. Nudelman, K. Leyton-Brown, A. Devkar, Y. Shoham, and H. Hoos. Understanding random SAT: Beyond the clauses-to-variables ratio. In M. Wallace, editor, *Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming (CP'04)*, volume 3258 of *Lecture Notes in Computer Science*, pages 438–452. Springer-Verlag, 2004.

B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Tell me who can learn you and I can tell you who you are: Landmarking Various Learning Algorithms. In *Proceedings of the 17th international conference on machine learning (ICML)*, pages 743–750, 2000.

L. Pulina. The ninth QBF solvers evaluation - preliminary report. In F. Lonsing and M. Seidl, editors, *Proceedings of the 4th International Workshop on Quantified Boolean Formulas (QBF'16)*, volume 1719 of *CEUR Workshop Proceedings*, pages 1–13. CEUR-WS.org, 2016.

L. Pulina and A. Tacchella. AQME'10. *JSAT*, 7(2-3):65–70, 2010.

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

H. Samulowitz, C. Reddy, A. Sabharwal, and M. Sellmann. Snappy: A simple algorithm portfolio. In M. Järvisalo and A. Van Gelder, editors, *Proceedings of the 16th International Conference on Theory and Applications of Satisfiability Testing*, volume 7962 of *Lecture Notes in Computer Science*, pages 422–428. Springer, 2013.

P. Stuckey, T. Feydy, A. Schutt, G. Tack, and J. Fischer. The minizinc challenge 2008-2013. *AI Magazine*, 35(2):55–60, 2014.

Q. Sun and B. Pfahringer. Pairwise meta-rules for better meta-learning-based algorithm ranking. *Machine learning*, 93(1):141–161, 2013.

J. N. van Rijn. *Massively Collaborative Machine Learning.* PhD thesis, Leiden University, 2016.

J. N. van Rijn, S. Abdulrahman, P. Brazdil, and J. Vanschoren. Fast Algorithm Selection using Learning Curves. In *Proceedings of the International Symposium on Advances in Intelligent Data Analysis XIV*, pages 298–309. Springer, 2015a.

J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren. Having a Blast: Meta-Learning and Heterogeneous Ensembles for Data Streams. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1003–1008. IEEE, 2015b.

M. Wagner, M. Lindauer, M. Misir, S. Nallaperuma, and F. Hutter. A case study of algorithm selection for the traveling thief problem. *Journal of Heuristics*, pages 1–26, 2017.

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

L. Xu, F. Hutter, H. Hoos, and K. Leyton-Brown. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In *RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion at the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.