Robust Strategy Schedule Optimization for an Automatic Theorem Prover*

Filip Bártek and Martin Suda

Czech Technical University in Prague, Czech Republic
{filip.bartek,martin.suda}@cvut.cz

Motivation

Automatic theorem provers (ATPs) such as Vampire [5] typically expose many parameters that control the proof search. In the context of automatic theorem proving, each valid configuration of the parameters is called a *strategy*. The performance of an ATP on an input problem often varies greatly depending on the chosen strategy. When dealing with a heterogeneous input problem distribution such as TPTP [9], a large performance gain can be achieved by employing a diverse portfolio of strategies instead of just one strong strategy. By ordering the strategies in the portfolio and assigning each of them a time limit, we get a *strategy schedule*.

In this short paper, we describe a procedure we used to construct strategy schedules for the ATP Vampire for the 11th IJCAR ATP System Competition (CASC-J11). While the task of schedule optimization has been tackled both in the context of automatic theorem proving [3, 6, 11] and in general [12], our solution is original in several aspects:

- We used a randomized version of Vampire [8] as the target prover to obtain robust strategies. In effect, we disregarded strategies that rely on a particular implementation of a don't-care non-deterministic choice point to solve a given problem.
- We used a greedy approximation algorithm for the budgeted maximum coverage problem [4] to fit a strategy schedule to performance measurements. This way, we got a strong schedule quickly, even when fitting many observations. This was useful because we used many strategies and training problems, and each strategy-problem combination was evaluated several times to utilize the non-deterministic nature of the target prover.

For many years, Vampire's schedules for CASC were constructed by Andrei Voronkov with the help of a tool called Spider (unpublished). While we believe the above two points are indeed unique to our approach, some of its other aspects (e.g., the strategy simplification described below) could have been indirectly inspired by Spider through discussions with its author.

Architecture

Our process of constructing a strong schedule for Vampire 4.7 on a given training problem set consisted of four phases:

*This work was supported by the Czech Science Foundation project no. 20-06390Y (JUNIOR grant), the European Regional Development Fund under the Czech project AI&Reasoning no. CZ.02.1.01/0.0/0.0/15_003/00004, the project RICAIP no. 857306 under the EU-H2020 programme, and the Grant Agency of the Czech Technical University in Prague, grant no. SGS20/215/OHK3/3T/37.

Division	Problems	SnakeForV4.7		Strongest competitor	
		Solved	Rank	Name	Solved
TFA	250	**218**	1/5	cvc5	195
FOF	500	**460**	1/14	Vampire 4.7	451
FNT	250	159	3/6	Vampire 4.6	**167**
UEQ	250	207	4/9	Twee 2.4.1	**216**

Table 1: Performance of SnakeForV4.7 in CASC-J11

Strategy sampling. First, we sampled strategies from a fixed distribution on the strategy space. We evaluated each sampled strategy on one problem from the training set. We sampled the problems adaptively using the upper confidence bound method [1] to concentrate on hard problems. We sampled the instruction limit from a log-uniform distribution on the interval from 5×10^9 to 5×10^{10} instructions.

Strategy selection. We assessed the empirical hardness of the problems by counting successful and unsuccessful proof attempts using the sampled strategies. When a strategy solved a problem considered hard, we selected this strategy for simplification. We stored the problem that testifies to the strategy's usefulness along with the strategy.

Strategy simplification. We optimized each selected strategy by local search (modifying one parameter at a time). We considered a neighbor strategy better if it improved performance on the respective witness problem. As a tie-breaker, we favored the default values of the parameters over the non-default values. Furthermore, once a parameter was set to its default value, we no longer searched for alternatives. In effect, we gradually reduced (and never increased) the number of non-default parameter values, thus simplifying the strategy.

Schedule optimization. We evaluated each simplified strategy on the whole training problem set a number of times. The number of evaluations was higher for strategies considered useful by the schedule optimizer. We used the results of these evaluations to construct a strategy schedule that approximately optimized the expected performance on the training problem set.

Results and Future Work

Our system entered the demonstration division of CASC-J11 [10] under the name *SnakeForV4.7*. It was evaluated in four competition divisions: Typed (monomorphic) First-order with Arithmetic theorems (TFA), First-Order Form theorems (FOF), First-order form Non-Theorems (FNT), and Unit EQuality clause normal form theorems (UEQ). For each of these divisions, we optimized a schedule using the corresponding problems from TPTP v7.5.0. The final results are summarized in Table 1. Note especially that our system outperformed all competitors in divisions TFA and FOF.[1]

For the presentation at AITP, we plan to analyze the data collected during our schedule construction to answer questions such as which parameter value combinations make up for successful proving strategies, or which distribution on the strategy space should be used to quickly discover complementary strategies useful for constructing strong schedules.

[1] Vampire 4.7 (and 4.6) relied on schedules constructed by Spider in 2019, so did not reflect on features added to Vampire more recently, such as layered clause selection [2] or new arithmetic reasoning rules [7].

References

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002. doi:10.1023/A:1013689704352.

[2] Bernhard Gleiss and Martin Suda. Layered clause selection for saturation-based theorem proving. In Pascal Fontaine, Konstantin Korovin, Ilias S. Kotsireas, Philipp Rümmer, and Sophie Tourret, editors, *Joint Proceedings of the 7th Workshop on Practical Aspects of Automated Reasoning (PAAR) and the 5th Satisfiability Checking and Symbolic Computation Workshop (SC-Square) Workshop, 2020 co-located with the 10th International Joint Conference on Automated Reasoning (IJCAR 2020), Paris, France, June-July, 2020 (Virtual)*, volume 2752 of *CEUR Workshop Proceedings*, pages 34–52. CEUR-WS.org, 2020. URL `http://ceur-ws.org/Vol-2752/paper3.pdf`.

[3] Edvard K. Holden and Konstantin Korovin. Heterogeneous heuristic optimisation and scheduling for first-order theorem proving. In Fairouz Kamareddine and Claudio Sacerdoti Coen, editors, *Intelligent Computer Mathematics - 14th International Conference, CICM 2021, Timisoara, Romania, July 26-31, 2021, Proceedings*, volume 12833 of *Lecture Notes in Computer Science*, pages 107–123. Springer, 2021. ISBN 978-3-030-81096-2. doi:10.1007/978-3-030-81097-9_8.

[4] Samir Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999. doi:10.1016/S0020-0190(99)00031-9.

[5] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2013. ISBN 978-3-642-39798-1. doi:10.1007/978-3-642-39799-8_1.

[6] Daniel Kühlwein and Josef Urban. Males: A framework for automatic tuning of automated theorem provers. *J. Autom. Reason.*, 55(2):91–116, 2015. doi:10.1007/s10817-015-9329-1.

[7] Giles Reger, Johannes Schoisswohl, and Andrei Voronkov. Making theory reasoning simpler. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part II*, volume 12652 of *Lecture Notes in Computer Science*, pages 164–180. Springer, 2021. ISBN 978-3-030-72012-4. doi:10.1007/978-3-030-72013-1_9.

[8] Martin Suda. Vampire getting noisy: Will random bits help conquer chaos? (system description). In Jasmin Blanchette, Laura Kovács, and Dirk Pattinson, editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 659–667. Springer, 2022. ISBN 978-3-031-10768-9. doi:10.1007/978-3-031-10769-6_38.

[9] Geoff Sutcliffe. The TPTP problem library and associated infrastructure. *Journal of Automated Reasoning*, 59(4), December 2017. ISSN 0168-7433. doi:10.1007/s10817-017-9407-7.

[10] Geoff Sutcliffe and Martin Desharnais. The 11th IJCAR automated theorem proving system competition - CASC-J11. *AI Commun.*, Preprint:1–19, 2023. ISSN 1875-8452. doi:10.3233/AIC-220244.

[11] Tanel Tammet. Towards efficient subsumption. In Claude Kirchner and Hélène Kirchner, editors, *Automated Deduction - CADE-15, 15th International Conference on Automated Deduction, Lindau, Germany, July 5-10, 1998, Proceedings*, volume 1421 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 1998. ISBN 3-540-64675-2. doi:10.1007/BFb0054276.

[12] Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Hydra: Automatically configuring algorithms for portfolio-based selection. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1929`.