

The International SAT Solver Competitions

Matti Järvisalo
HIIT & Dept. Computer Science,
University of Helsinki,
Finland

Daniel Le Berre
CRIL/CNRS,
Université d'Artois,
Lens, France

Olivier Roussel
CRIL/CNRS,
Université d'Artois,
Lens, France

Laurent Simon
LRI/CNRS,
Université Paris Sud,
Orsay, France

Abstract

The International SAT Solver Competition is today an established series of competitive events aiming at objectively evaluating the progress in state-of-the-art procedures for solving Boolean satisfiability (SAT) instances. Over the years, the competitions have significantly contributed to the fast progress in SAT solver technology that has made SAT a practical success story of computer science. This short article provides an overview of the SAT solver competitions.

1 Introduction

Determining whether a given propositional logic formula is satisfiable is one of the most fundamental problems in computer science. This canonical NP-Complete question is known as the Boolean Satisfiability (SAT) problem [2]. In addition to its theoretical importance, major advances in the development of robust implementations of decision procedures for SAT, *SAT solvers*, has established SAT as an important declarative approach for attacking various complex search and optimization problems. Modern SAT solvers are routinely used as core solving engines in vast numbers of different AI and industrial applications.

The *International SAT Solver Competition* (<http://www.satcompetition.org/>) is today an established series of competitive events aiming at objectively evaluating the progress in state-of-the-art SAT solving techniques. In this short article, we will provide an overview of the SAT solver competitions.

1.1 A Short History

The first SAT Competition took place in Paderborn in 1992 and was organized by Buro and Kleine Büning [4]. The second SAT Competition took place during the second Dimacs challenge in 1993 [10]. The *Dimacs* CNF (for *conjunctive normal form* formulas) input format used in the competition has become the standard input format for CNF SAT solvers (see, e.g., [9] for details). Another SAT competition took place in Beijing in 1996, organized by James Crawford.

Starting the current line of SAT Competitions, a new series of competitions started in 2002, taking place during the SAT 2002 conference, as a consequence of the design of two new kind of solvers: survey propagation [3], a new approach to efficiently solve randomly generated (satisfiable) instances, and Chaff [13], one of the first efficient implementations of the *conflict-driven clause learning* (CDCL) algorithm [2, Chapter 4][14]. The underlying idea of this series of competitions is to evaluate objectively – by third parties – the efficiency of new SAT solvers on a wide range of benchmarks. Numerous research groups contributed both solvers and benchmarks to the 2002 competition, which led to the decision of organizing a competition on a yearly basis.

The SAT competition was initiated in 2002 by John Franco and Hans van Maaren, and was organized by Edward Hirsch, Daniel Le Berre and Laurent Simon. Daniel Le Berre and Laurent Simon organized the SAT competition from 2003 to 2009. Olivier Roussel joined the team in 2007. Matti Järvisalo joined the team in 2011.

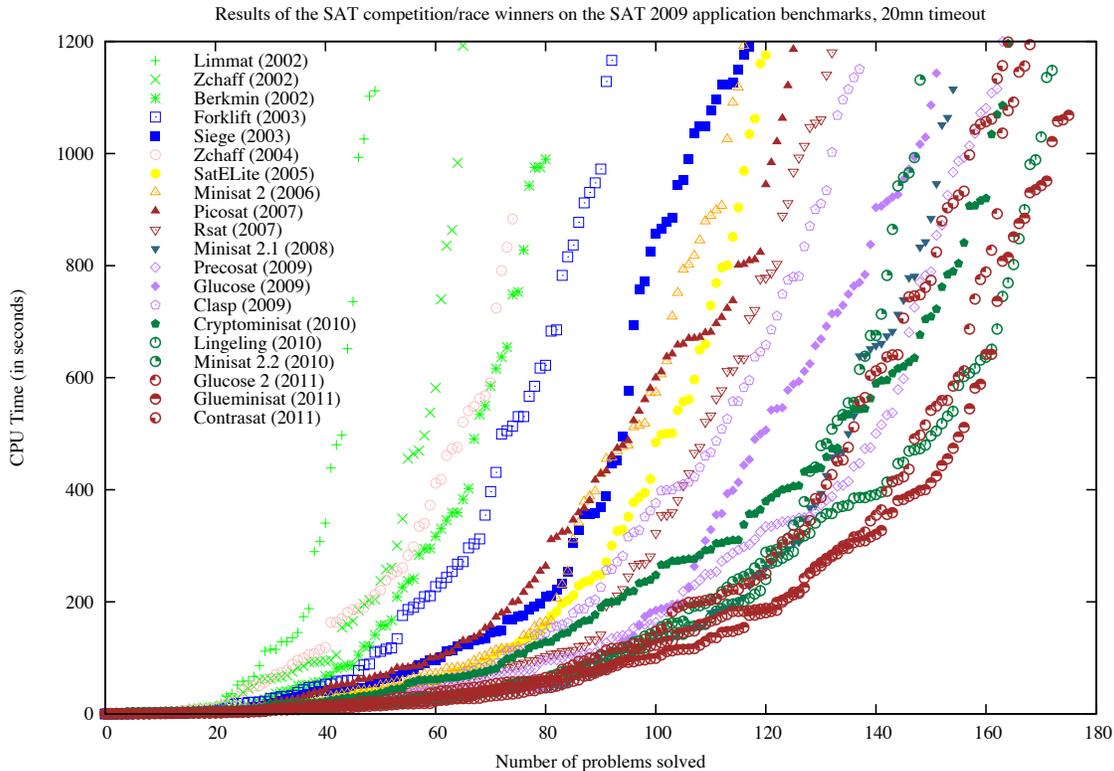


Figure 1: Evolution of the best solvers from 2002 to 2010 on the application benchmarks from the SAT 2009 competition using the cumulative number of problems solved (x axis) within a specific amount of time (y axis). The farther to the right the data points are, the better the solver.

The strong emphasis on application benchmarks led the community to organize a SAT Race in 2006, an event especially dedicated to industrial application problems (for details on the latest SAT Race, see <http://baldur.iti.uka.de/sat-race-2010/>, chaired by Carsten Sinz). Since then, SAT Competition and SAT Race have alternated, the former having been organized in the odd years, and the latter in even years.

2 Details on the Competitions

In the main track of the competition, the goal is to determine whether a given SAT instance in conjunctive

normal form (CNF) is satisfiable or not as quickly as possible. For satisfiable formulas, solvers are required to output a model of the formula as a certificate.

The main track is run in two phases. The best solvers of the first phase (selected by the competition jury) enter the second phase and are allocated a longer timeout. Solvers are awarded according to the number of benchmarks solved during the second stage, using the cumulated time required to solve those benchmarks to break ties. In 2011, two different rankings were used: one based on CPU time which promotes solvers using resources as efficiently as possible (e.g. sequential solvers) and another one based on wall clock time which promotes solvers using all available resources to answer as quickly as possible

(e.g. parallel solvers).

A solver that answers incorrectly is disqualified and does not appear in the rankings. A solver answers incorrectly if it reports *satisfiable* but reports an assignment that is not a model of the input CNF, or reports *unsatisfiable* on a formula that is known to be satisfiable.

In the main track there are three competition categories: application, crafted, and random. Each category is defined through the type of instances used as benchmarks.

Application instances (formerly called the “industrial” category) encode various “application” problems in CNF. These instances are typically large (containing up to tens of millions of variables and clauses). The motivation behind this category is to highlight the kind of applications SAT solvers may be useful for.

Crafted instances are often designed to give a hard time to SAT solvers, or represent otherwise problems which are challenging to typical SAT solvers (including, e.g., instances arising from puzzle games). These benchmarks are typically small. The motivation behind this category is to highlight current challenge problem domains that reveal the limits of current SAT solver technology.

Random instances are randomly generated uniform random k -SAT formulas [2, Chapter 8]. This category is motivated by the fact that the instances can be fully characterized and by its connection especially to statistical physics.

The number of benchmarks in each category has typically been around 300. In 2011, the smallest crafted instance not solved by any solver within the timeout contained only 141 variables, 292 clauses, and 876 literals in total. In contrast, the biggest application instance solved by at least one solver contained 10M variables, 32M clauses, and a total of 76M literals.

The competition is open for everyone, with an open call for participation. The community at large is

invited to submit new benchmarks to the competition. Based on submitted benchmarks, the competition jury selects each year the actual benchmarks used in each of the competition categories from both benchmarks used in the previous competitions and the submitted ones.

Traditionally, the three best solvers in each category are awarded medals (gold, silver, bronze). Furthermore, within each category, a distinction is made between unsatisfiable and satisfiable instances: restricted to satisfiable and unsatisfiable instances, respectively, three best solvers within each category are also awarded medals (e.g., gold in the satisfiable application category). Furthermore, in 2011 solvers were separately awarded based on the CPU and wall clock time based rankings.

As an important design principle, the competition rules state that the source codes of the awarded solvers have to be made open for research purposes. The competition data, including all benchmarks and the output of each solver on each benchmark, is also made publicly available on the competition website by the organizers.

Additional competition tracks In addition to the main track, various special tracks have been organized within the competition during the years. The aim of a special track is to encourage emerging domains. As an example, Allen Van Gelder has multiple times organized a *certified unsat track*, in which solvers compete on efficiently providing certificates of unsatisfiability for unsatisfiable SAT instances. The *Pseudo-Boolean competition* organized by Olivier Roussel and Vasco Manquinho started in 2005 as a special track of the SAT competition. It is now organized independently on a yearly basis. Most recently, in 2011 a special track on finding *Minimally Unsatisfiable Subsets* (MUSes) [2, Chapter 11] of CNF formulas was organized with the help of Joao Marques Silva.

An additional track worth mentioning, lowering the threshold for e.g. students to participate in the competition, is the *Minisat hack track*. In this track, the idea is to submit a modified version of the frequently used Minisat solver [6]. In principle, only mi-

nor, specific modifications to the base Minisat solver are allowed. The best Minisat hack is given a special award.

2.1 Organizational Structure

The organization structure of each competition is composed of the main *organization team* and a *panel of judges*. The organization team invites the judges to the panel, with the aim of constructing a panel with a wide overall perspective on SAT (e.g., both academics and industrial partners).

The organization team takes main responsibility for practical arrangements: competitions calls, setting up the competition, running the solvers on the competition benchmarks, gathering the competition data, etc. The organization team consults the panel of judges on all important decisions, including benchmark selection, disqualification of unsound solvers, solver ranking, etc. The panel of judges has the ultimate word over all decisions.

3 Competition Entrants

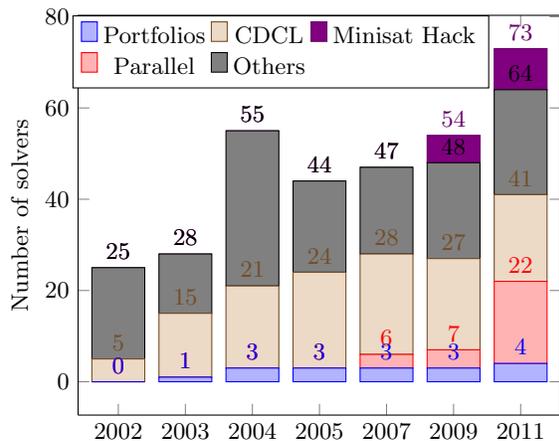


Figure 2: Evolution of SAT competition participation

In 2011, 78 solvers were submitted for competing in the main track. A single submitter (person or team)

can submit up to three solvers per category (application, crafted, random) and per kind (parallel, sequential). Most participants are coming from academia. Intel submitted several times its own SAT solver Eureka. There are also a few independent individuals submitting solvers each year.

A majority of the SAT solvers submitted are CDCL-based [2, Chapter 4][14] (see Figure 2). While Satzilla [18] has been the only solver portfolio to compete during 2003-2009 (with a specialized portfolio submitted to each category), it did not participate in 2011, which was the first year when other similar multi-engine systems took part in the competition.

CDCL has been the dominating approach for the application and crafted categories invariably during each competition. Indeed, currently the “must-have” features to win the SAT competition in the application category is to use a CDCL-based SAT solver, incorporating various modern techniques such as rapid restarts, phase savings, aggressive cleanup of learned clauses, and clause minimization, along with powerful preprocessing. In the crafted category, the best “brute force” solver usually wins, i.e., the CDCL (or plain DPLL [5]) solver with the fastest exploration of the search space: since the benchmarks are designed to be hard for resolution based solvers, most simplification techniques (clause minimization, preprocessing) and heuristics (phase saving) do not help. As a recent development, the best non-portfolio solver in the satisfiable crafted category was incomplete (based on local search [2, Chapter 6]). In the random category, local search solvers outperform the other solvers on satisfiable benchmarks while CDCL solvers perform rather weakly. Simple but fast DPLL-based solvers incorporating sophisticated look-ahead heuristics [2, Chapter 5][8] perform currently best on unsatisfiable random instances.

Running several instances of the same solver with different settings, sharing only a small amount of information (such as learned unit clauses only), appears to be currently one of the best and simplest ways of taking advantage of multi-core computers [7].

An easy way to perform well in the competition is to build a portfolio with the winners of the previous competitions. However, by construction, such portfolios do not improve the state-of-the-art, defined as

the set of problems solvable by the current solvers.

The combination of these portfolio and parallel approaches is illustrated by portfolio, submitted to the SAT 2011 competition by Olivier Roussel. It simply executes a few fixed solvers in parallel, with no communication at all between the individual solvers. It was submitted to the competition in order to identify a lower bound on what can be achieved with both portfolios and parallel solvers. Despite the naive approach, it obtained unexpectedly good results, which clearly demonstrates that there is room for improvements in both approaches.

4 Lessons Learned

Over the years, the competitions have significantly contributed to the fast progress in SAT solver technology that has made SAT a practical success story of computer science. The SAT competition allowed the community to provide robust, reliable, and generic purposes SAT solvers to other research communities: while many solvers crashed or where found incorrect in the SAT 2002 competition, the current SAT solvers hardly crash, and can in cases be run even for days when attempting to solve complex instances. The widespread use of SAT technology in many areas also pushed the community to provide easily embeddable solvers. For instance, the CDCL-based Minisat [6] and Picosat [1] solvers are widely reused within and outside that SAT community.

Winning the SAT competition has become a major challenge, providing incentives for inventing and implementing novel solver techniques. The competition encourages young researchers, including students, to take part in the competition by implementing their ideas on the latest version of Minisat, which contributes to ensuring an active future of SAT research.

The openness and peer-verification of the competition is important. Each single run can be checked by the community on the competition web site. Each competitor is responsible for checking the results of his or her solver during the competition. The competition data and the source code of the competitors are made available for analysis to the community at large. The SAT Competition benchmark sets pro-

vide a standardized benchmark collection for the use of researchers.

The SAT competition has been inspired by the CASC automated theorem proving competition [17] and the earlier work of Laurent Simon on SAT-Ex [15]. The SAT competitions have inspired the establishment of similar competitions in related areas, including satisfiability modulo theories (SMT) competitions, quantified Boolean formula (QBF) evaluations, answer set programming (ASP) competitions, pseudo-Boolean and MaxSAT competitions, CSP competitions, Hardware Model Checking competitions (HWMCC) and most recently the diagnostic competitions, among others.

For more details on the SAT Competitions, visit the competition website at <http://www.satcompetition.org>. Reports on some of the previous competition instantiations have also been published [16, 12, 11]. A full article on the details of the 2011 competition is currently under preparation.

Acknowledgements

M.J. is financially supported by Academy of Finland under grant 132812. D.L.B. and O.R. are partly supported by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the Contrat de Projets Etat Region (CPER) 2007-2013. L.S is partly supported by the French National Project ANR UNLOC BLAN08-1 328904.

References

- [1] A. Biere. PicoSAT essentials. *Journal of Satisfiability, Boolean Modeling and Computation*, 4(2-4):75–97, 2008.
- [2] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [3] A. Braunstein and R. Zecchina. Survey and belief propagation on random K-SAT. In *SAT 2003 Revised Selected Papers*, volume 2919 of *LNCS*, pages 519–528. Springer, 2004.

- [4] M. Buro and H. Kleine Büning. Report on a SAT competition. *Bulletin of the European Association for Theoretical Computer Science*, 49:143–151, 1993.
- [5] M. Davis, G. Logemann, and D.W. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [6] N. Eén and N. Sörensson. An extensible SAT-solver. In *SAT 2003 Selected Revised Papers*, volume 2919 of *LNCS*, pages 502–518. Springer, 2004.
- [7] Y. Hamadi, S. Jabbour, and L. Sais. ManySAT: a parallel SAT solver. *Journal of Satisfiability, Boolean Modeling and Computation*, 6(4):245–262, 2009.
- [8] M. Heule, M. Dufour, J. van Zwieten, and H. van Maaren. March_eq: Implementing additional reasoning into an efficient look-ahead SAT solver. In *SAT 2004 Revised Selected Papers*, volume 3542 of *LNCS*, pages 345–359. Springer, 2005.
- [9] M. Järvisalo, D. Le Berre, and O. Roussel. Rules of the 2011 SAT Competition, 2011. <http://www.satcompetition.org/2011/rules.pdf>.
- [10] D. S. Johnson and M. A. Trick, editors. *Second DIMACS implementation challenge: Cliques, coloring and satisfiability*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1996.
- [11] D. Le Berre and L. Simon. The essentials of the SAT 2003 Competition. In *Proc. SAT 2003*, volume 2919 of *LNCS*, pages 452–467. Springer, 2004.
- [12] D. Le Berre and L. Simon. Fifty-five solvers in Vancouver: The SAT 2004 Competition. In *SAT 2004 Selected Paper*, volume 3542 of *LNCS*, pages 321–344. Springer, 2005.
- [13] M.W. Moskewicz, C.F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proc. DAC*, pages 530–535. ACM, 2001.
- [14] J.P. Marques Silva and K.A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, 1999.
- [15] L. Simon and P. Chatalic. SatEx: A web-based framework for SAT experimentation. *Electronic Notes in Discrete Mathematics*, 9:129–149, 2001.
- [16] L. Simon, D. Le Berre, and E.A. Hirsch. The SAT2002 competition. *Annals of Mathematics and Artificial Intelligence*, 43(1):307–342, 2005.
- [17] G. Sutcliffe and C. Suttner. Evaluating general purpose automated theorem proving systems. *Artificial Intelligence*, 131:39–54, 2001.
- [18] 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, 2008.

Author Biographies

Matti Järvisalo is currently a postdoctoral fellow of Academy of Finland at University of Helsinki. He received his doctoral degree in computer science in 2008 from Helsinki University of Technology (today part of Aalto University). Most of his research is centered around efficient and robust techniques for solving computationally hard constraint satisfaction and optimization problems, with a focus on Boolean-based decision and optimization procedures and their contemporary real-world applications.

Daniel Le Berre is currently associate professor (*Maître de Conférences*) at Artois University, France. He received his doctoral degree in computer science in 2000 from Toulouse University. His research is centered around the design of efficient techniques for boolean reasoning and their application to artificial intelligence and software engineering. He co-initiated the SAT competition in 2002 and has been involved in all SAT competitive events since then.

Olivier Roussel is associate professor (*Maître de Conférences*) at IUT of Lens, University of Artois, France. He received his doctoral degree in computer science in 1997 from University of Lille, France. His research interests are focused on satisfiability in a broad sense, including SAT, pseudo-Booleans, CSP, WCSP. He has organized several editions of various competitions (SAT, pseudo-Booleans, CSP).

Laurent Simon is associate professor (*Maître de Conférences*) at Paris-Sud University, France. He received his doctoral degree in computer science in 2001 from the same University. His research focuses on designing efficient SAT solvers and knowledge compilation techniques. He co-initiated the SAT competition in 2002, in which he was involved until 2009.