The International SAT Solver Competitions

Matti Järvisalo, Daniel Le Berre, Olivier Roussel, Laurent Simon

■ *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.*

Determining whether a given propositional logic formula is satisfiable is one of the most fundamental problems in computer science, known as the canonical NP-complete Boolean satisfiability (SAT) problem (Biere et al. 2009). In addition to its theoretical importance, major advances in the development of robust implementations of decision procedures for SAT, SAT solvers, have 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[1] 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.

A Short History

The first SAT competition took place in Paderborn in 1992 and was organized by Michael Buro and Hans Kleine Büning (Buro and Büning 1993). The second SAT competition took place during the second Dimacs challenge in 1993 (Johnson and Trick 1996). The Dimacs conjunctive normal form (CNF) formula input format used in the competition has become the standard input format for CNF SAT solvers.[2] 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 (Braunstein and Zecchina 2004), a new approach to efficiently solve randomly generated (satisfiable) instances, and Chaff (Moskewicz et al. 2001), one of the first efficient implementations of the conflict-driven clause learning (CDCL) algorithm (Biere et al. [2009], chapter 4), (Silva and Sakallah 1999). 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.

The strong emphasis on application benchmarks led the community to organize a SAT race[3] in 2006, an event especially dedicated to industrial application problems. Since then, the SAT competition and the SAT race have alternated, the former having been organized in the odd years, and the latter in even years.

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 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 time-out. 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 (for example, sequential solvers), and another one based on wall clock time, which promotes solvers using all available resources to answer as quickly as possible (for example, 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 that are challenging to typical SAT solvers (including, for example, instances arising from puzzle games). These benchmarks are typically small. The motivation behind this category is to highlight current challenging problem domains that reveal the limits of current SAT solver technology.

Random instances are randomly generated uniform random k-SAT formulas (Biere et al. [2009], 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 time-out contained only 141 variables, 292 clauses, and 876 literals in total. In contrast, the biggest application instance solved by at least one solver contained 10 million variables, 32 million clauses, and a total of 76 million 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 (for example, 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) (Biere et al. [2009], chapter 11) of CNF formulas was organized with the help of Joao Marques Silva.

An additional track worth mentioning, lowering the threshold for, for example, 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 (Eén and Sörensson 2004). In principle, only minor, specific modifications to the base Minisat solver are allowed. The best Minisat hack is given a special award.

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 (for example, 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, and so on. The organization team consults the panel of judges on all important decisions, including benchmark selection, disqualification of unsound solvers, solver ranking, and so on. The panel of judges has the ultimate word over all decisions.

Competition Entrants

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 (Biere et al. [2009], chapter 4; Silva and Sakallah [1999]) (see figure 1). While Satzilla (Xu et al. 2008) 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 multiengine 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, that is, the CDCL (or plain DPLL) (Davis, Logemann, and Loveland 1962) solver with the fastest exploration of the search space. Because many of the the benchmarks are designed to be hard for resolution-based solvers, most simplification techniques (clause minimization, preprocessing) and heuristics (phase saving) do not often help. As a recent development, the best nonportfolio solver in the satisfiable crafted category was incomplete (based on local search (Biere et al. [2009], 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 (Biere et al. [2009], chapter 5; Heule et al. [2005]) 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 multicore computers (Hamadi, Jabbour, and Sais 2009). An easy way to perform well in the competition is to build a portfolio based on 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

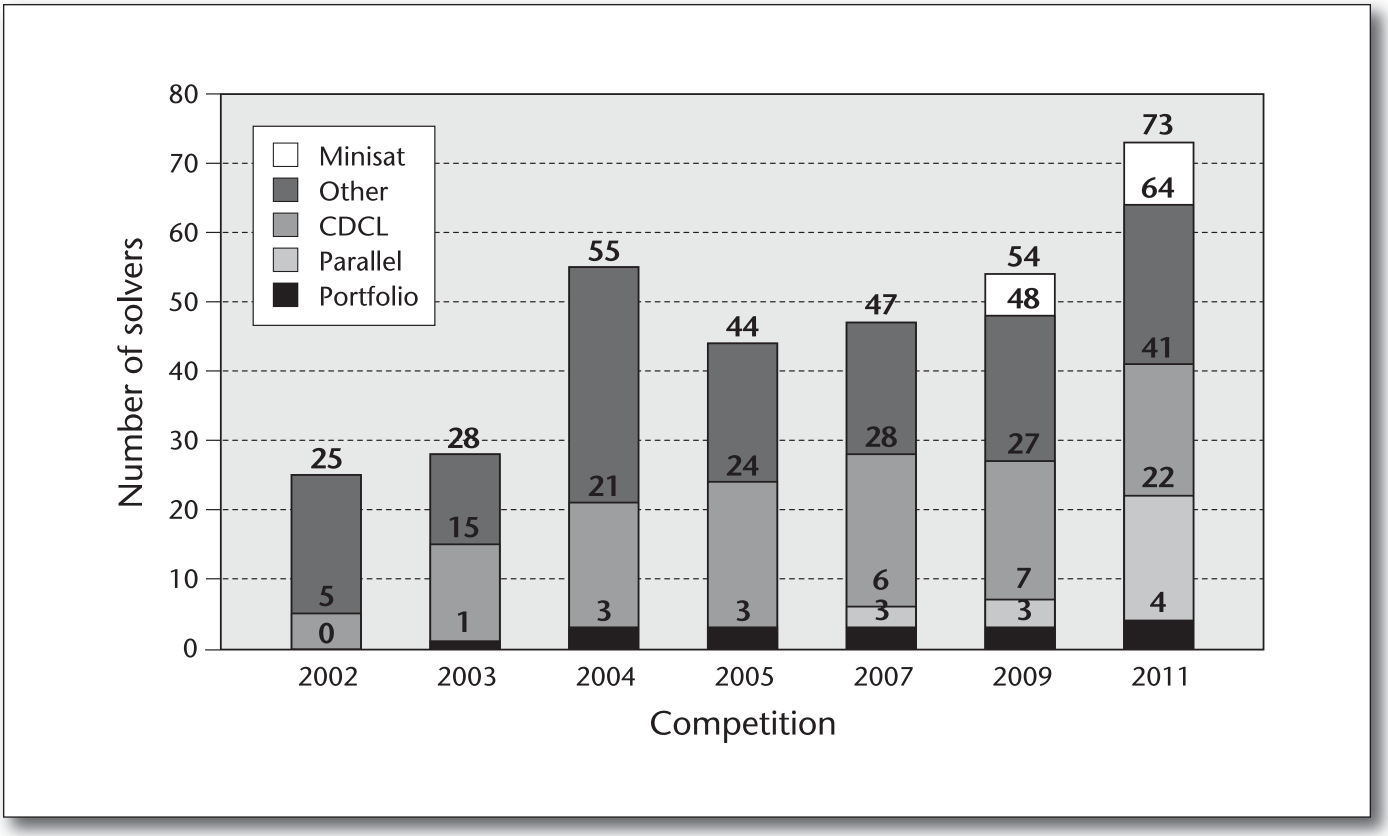

Figure 1. Evolution of SAT Competition Participation.

approaches is illustrated by ppfolio, 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.

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 were 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 (Eén and Sörensson 2004) and Picosat (Biere 2008) solvers are widely reused within and outside the SAT community.

Figure 2 shows the evolution of the best solvers from 2002 to 2011 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).

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 website. 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 provide a standardized benchmark collection for the use of researchers.

The SAT competition has been inspired by the

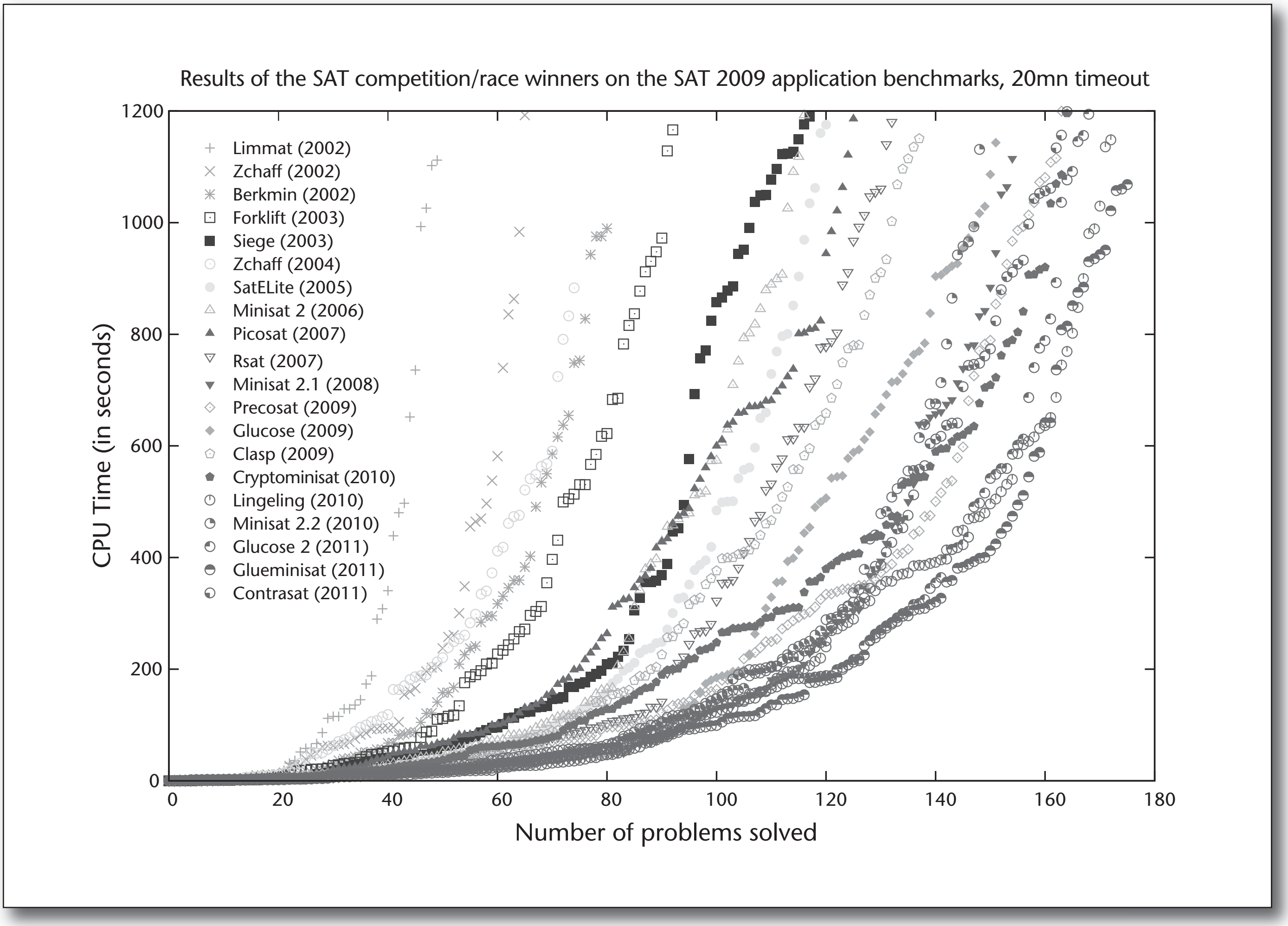

Figure 2. Performance Evolution of the Best SAT Solvers from 2002 to 2011.

The farther to the right the data points are, the better the solver.

CASC automated theorem proving competition (Sutcliffe and Suttner 2001) and the earlier work of Laurent Simon on SAT-Ex (Simon and Chatalic 2001). 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 (DXC), among others.

For more details on the SAT Competitions, visit the competition website.[4] Reports on some of the previous competition instantiations have also been published (Simon, Le Berre, and Hirsch 2005; Le Berre and Simon 2005; 2004). A full article on the details of the 2011 competition is currently under preparation. This year, the SAT Challenge 2012 will be organized, combining features from both SAT Competitions and SAT-Races.

Acknowledgements

Matti Järvisalo is financially supported by Academy of Finland under grant 132812. Daniel Le Berre and Olivier Roussel 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. Laurent Simon is partly supported by the French National Project ANR UNLOC BLAN08-1 328904.

Notes

1. See www.satcompetition.org.
2. See, for example, Järvisalo, Le Berre, and Roussel, Rules

of the 2011 SAT Competition, for details (www.satcompetition.org/2011/rules.pdf).

3. For details on the latest SAT Race, see baldur.iti.uka.de/sat-race-2010, chaired by Carsten Sinz.

4. See www.satcompetition.org.

References

Biere, A. 2008. PicoSAT Essentials. *Journal of Satisfiability, Boolean Modeling and Computation* 4(2–4): 75–97.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds. 2009. Handbook of Satisfiability. In *Frontiers in Artificial Intelligence and Applications* volume 185. Amsterdam, Holland: IOS Press.

Braunstein, A., and Zecchina, R. 2004. Survey and Belief Propagation on Random K-SAT. In *Revised Selected Papers from the Sixth International Conference on Theory and Applications of Satisfiability Testing* (SAT 2003), volume 2919 of Lecture Notes in Computer Science, 519–528. Berlin: Springer.

Buro, M., and Büning, H. K. 1993. Report on a SAT Competition. *Bulletin of the European Association for Theoretical Computer Science* 49(1): 143–151.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A Machine Program for Theorem-Proving. *Communications of the ACM* 5(7): 394–397.

Eén, N., and Sörensson, N. 2004. An Extensible SAT-Solver. In *Revised Selected Papers from the Sixth International Conference on Theory and Applications of Satisfiability Testing* (SAT 2004), volume 2919 of Lecture Notes in Computer Science, 502–518. Berlin: Springer.

Hamadi, Y.; Jabbour, S.; and Sais, L. 2009. ManySAT: a Parallel SAT Solver. *Journal of Satisfiability, Boolean Modeling and Computation* 6(4): 245–262.

Heule, M.; Dufour, M.; van Zwieten, J.; and van Maaren, H. 2005. March_eq: Implementing Additional Reasoning into an Efficient Look-Ahead SAT Solver. In *Revised Selected Papers from the Seventh International Conference on Theory and Applications of Satisfiability Testing* (SAT 2004), volume 3542 of Lecture Notes in Computer Science, 345–359. Berlin: Springer.

Johnson, D., and Trick, M., eds. 1996. *Second DIMACS Implementation Challenge: Cliques, Coloring, and Satisfiability,* volume 26 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Providence, RI: American Mathematical Society.

Le Berre, D., and Simon, L. 2005. Fifty-Five Solvers in Vancouver: The SAT 2004 Competition. In *Revised Selected Papers from the Seventh International Conference on Theory and Applications of Satisfiability Testing* (SAT 2004), volume 3542 of Lecture Notes in Computer Science, 321–344. Berlin: Springer.

Le Berre, D., and Simon, L. 2004. The Essentials of the SAT 2003 Competition. In *Revised Selected Papers from the Sixth International Conference on Theory and Applications of Satisfiability Testing* (SAT 2003), volume 2919 of Lecture Notes in Computer Science, 452–467. Berlin: Springer.

Moskewicz, M.; Madigan, C.; Zhao, Y.; Zhang, L.; and Malik, S. 2001. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the Thirty-Eighth Design Automation Conference,* 530–535. New York: Association for Computing Machinery.

Silva, J. M., and Sakallah, K. 1999. GRASP: A Search Algorithm for Propositional Satisfiability. *IEEE Transactions on Computers* 48(5): 506–521.

Simon, L., and Chatalic, P. 2001. SatEx: A Web-Based Framework for SAT Experimentation. *Electronic Notes in Discrete Mathematics* 9(9): 129–149.

Simon, L.; Le Berre, D.; and Hirsch, E. 2005. The SAT2002 Competition. *Annals of Mathematics and Artificial Intelligence* 43(1): 307–342.

Sutcliffe, G., and Suttner, C. 2001. Evaluating General Purpose Automated Theorem Proving Systems. *Artificial Intelligence* 131(1–2): 39–54.

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

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 an 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 on the design of efficient techniques for boolean reasoning and their application to artificial intelligence and software engineering. He coinitiated the SAT competition in 2002 and has been involved in all SAT competitive events since then.

Olivier Roussel is an 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 coinitiated the SAT competition in 2002, in which he was involved until 2009.