SAT 2022 Test of Time Award

Niklas Eén, Niklas Sörensson

An Extensible SAT-solver

published at SAT'03

25th International Conference on

Theory and Applications of Satisfiability Testing (SAT'22)

Friday, August 5, 2022

Federated Logic Conference (FLOC'22)

Haifa, Israel

presenter Armin Biere

International Conference on Theory and Applications of Satisfiability Testing

↳ SAT 2003: **Theory and Applications of Satisfiability Testing** pp 502–518 | Cite as

An Extensible SAT-solver

Niklas Eén & Niklas Sörensson

Conference paper

Part of the Lecture Notes in Computer Science book series (LNCS,volume 2919)

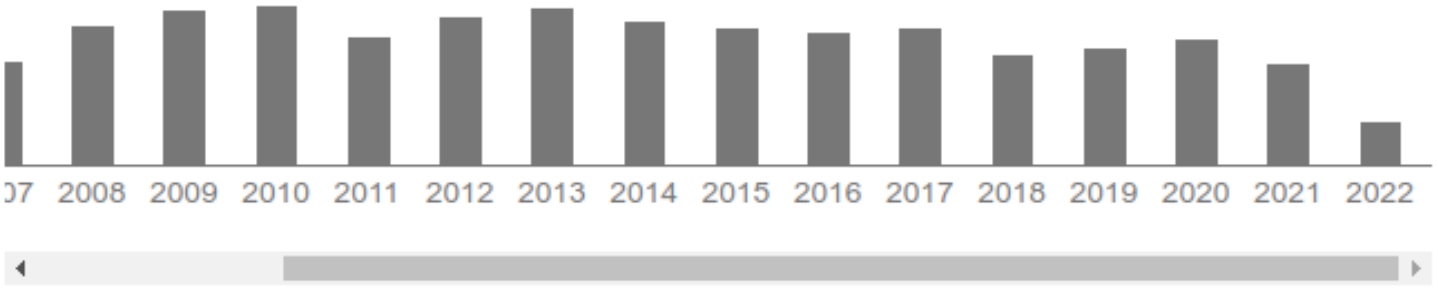
Abstract

In this article, we present a small, complete, and efficient SAT-solver in the style of conflict-driven learning, as exemplified by CHAFF. We aim to give sufficient details about implementation to enable the reader to construct his or her own solver in a very short time. This will allow *users* of SAT-solvers to make domain specific extensions or adaptions of current state-of-the-art SAT-techniques, to meet the needs of a particular application area. The presented solver is designed with this in mind, and includes among other things a mechanism for adding arbitrary boolean constraints. It also supports solving a series of related SAT-problems efficiently by an incremental SAT-interface.

An extensible SAT-solver

[PDF] from cmu.edu

Authors	Niklas Eén, Niklas Sörensson
Publication date	2003/5/5
Conference	International conference on theory and applications of satisfiability testing
Pages	502-518
Publisher	Springer, Berlin, Heidelberg
Description	In this article, we present a small, complete, and efficient SAT-solver in the style of conflict-driven learning, as exemplified by CHAFF. We aim to give sufficient details about implementation to enable the reader to construct his or her own solver in a very short time. This will allow *users* of SAT-solvers to make domain specific extensions or adaptions of current state-of-the-art SAT-techniques, to meet the needs of a particular application area. The presented solver is designed with this in mind, and includes among other things a mechanism for adding arbitrary boolean constraints. It also supports solving a series of related SAT-problems efficiently by an incremental SAT-interface.
Total citations	Cited by 4006



Scholar articles

An extensible SAT-solver
N Eén, N Sörensson - International conference on theory and applications of …, 2003
Cited by 4006 Related articles All 20 versions

Short summary

Inbox ×

Niklas Een Tue, Jul 19, 12:21 AM (11 days ago)

to me

Hi Armin,
It was great to catch up with you today. This was the short summary I wrote to Olaf:

"The paper pertaining to this award, started off as an invited paper for SAT 2003. My solver 'satnik', slightly improved and submitted under the name 'satzoo', had won one of the categories of the SAT-competition. I decided I wanted to use the opportunity to write a tutorial on implementing SAT solvers, and drafted the first version of the paper, later soliciting help from Niklas to co-author the final version. When we submitted the paper, 25 pages in total, the organizers informed us for the first time that the page limit was 6 pages. We pointed out that the error was on their part, not ours. To resolve the situation, the program committee decided to include the paper with the regular submissions (page limit 20, which we edited the paper down to), despite being weeks after the deadline. It was reviewed as a regular paper, as far as we know by reviewers unaware of its origin story. Their score came back "Reject". The sentiment was that the paper contained nothing novel. One reviewer stated, in his first sentence, that he disliked the paper, "and not only for the pointless exercises" (quote from memory). Despite this, the paper was included in the proceedings, for reasons never relayed to us. But it seems the tutorial found its audience and was eventually appreciated. "

// N.


Free
125 yds
P
LOMOND
TAVERN
Thistle
Cottage

How the authors entered SAT

- connection to CAV'97 in Haifa (25 years ago)
  - Stålmarck's company had an invited talk at CAV'97 in Haifa!
  - Ed Clarke saw potential of SAT in model checking
  - BMC at TACAS'99
- two master theses around 2001 / 2002 on SAT-based model checking
  - Niklas Eén as intern at Prover Technology
  - Niklas Sörensson at Chalmers but also using `prover`
- Limmat SAT solver
  - winner in one of the tracks of the SAT competition 2002
  - attempt to understand CHAFF by implementing the paper (DAC'01)
  - open source and worked well
- both thought about to implement a SAT solver too but first hesitated
  - Limmat even though simpler than Chaff still hard to "grasp"
  - I met with Niklas Eén at FLOC'02 (Copenhagen) to "go over the code"

SAT 2005 Competition

Organizing committee	Daniel Le Berre and Laurent Simon
Judges	Armin Biere, Oliver Kullmann and Allen Van Gelder
Reference	Daniel Le Berre and Laurent Simon Editors, Journal on Satisfiability, Boolean Modeling and Computation, Volume 2, Special Volume on the SAT 2005 competitions and evaluations, March 2006.
Benchmarks	Random (.tar.bz2, 25MB), Crafted (.tar.bz2, 360MB), Industrial (.tar.bz2, 205MB) See also IBM and Velev web sites.

	Industrial			handmade			Random		
	Gold	Silver	Bronze	Gold	Silver	Bronze	Gold	Silver	Bronze
SAT+UNSAT	SatELiteGTI	MiniSAT 1.13	zChaff_rand and HaifaSAT	Vallst	SatELiteGTI	March_dl	kcnf-2004	March_dl	Dew_Satz1a
SAT	SatELiteGTI	MiniSAT 1.13	Jerusat 1.31_fl and HaifaSAT	Vallst	March_dl	Hsatt	ranov	g2wsat	VW
UNSAT	SatELiteGTI	Zchaff_rand	HaifaSat	SatELiteGTI	MiniSAT 1.13	Vallst and March-dl	kcnf-2004	March_dl	Dew_Satz1a

Special tracks		
CERTIFIED UNSAT	zChaff	TTSP-3.0
NON CLAUSAL	No solver submitted	
PSEUDO BOOLEAN	Go to official web site	

SAT 2004 Competition

Organizing committee	Daniel Le Berre and Laurent Simon
Judges	Fahiem Bacchus, Hans Kleine Buning and Joao Marques Silva
Reference	55 Solvers in Vancouver: The SAT 2004 competition. Daniel Le Berre and Laurent Simon. Proceedings of the Seventh International Conference on Theory and Applications of Satisfiability Testing, SAT 2004, Springer, Lecture Notes in Computer Science, nï¿½ 3542, pp 321-344, 2005. Revised Selected Papers.
Benchmarks	Random (.tar.bz2, 11MB), Crafted (.tar.bz2, 36MB), Industrial (.tar.bz2, 2GB)

	Industrial	handmade	Random
ALL (SAT+UNSAT)	Zchaff 2004	March-eq	AdaptNovelty
SAT	Jerusat	Satzoo	AdaptNovelty
UNSAT	Zchaff 2004	March-eq	Kcnfs

SAT 2003 Competition

Organizing committee	Daniel Le Berre and Laurent Simon
Judges	John Franco, Hans van Maaren and Toby Walsh
Reference	The essentials of the SAT 2003 competition. Daniel Le Berre and Laurent Simon. Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT2003). Lecture Notes in Computer Science 2919, pp 452-467, 2003.
Benchmarks	available from SATLIB (.tar.bz2, 345MB)

	Industrial	handmade	Random
Complete on all	Forklift	Satzoo	Kcnfs
ALL on SAT	Forklift	Satzoo	Unitwalk

SAT 2002 Competition

Organizing committee	Edward A. Hirsch, Daniel Le Berre and Laurent Simon
Judges	N/A
Reference	The SAT2002 competition report. Laurent Simon, Daniel Le Berre and Edward A. Hirsch. Annals of Mathematics and Artificial Intelligence, Volume 43, Issue 1-4, pp. 307-342, January 2005 See also A Parsimony Tree for the SAT2002 Competition. Paul W. Purdom, Daniel Le Berre, Laurent Simon Annals of Mathematics and Artificial Intelligence, Volume 43, Issue 1-4, pp. 343-365, January 2005
Benchmarks	available from SATLIB (tgz, 147MB)

	Industrial	handmade	Random
Complete on all	zChaff	zChaff	OKSolver
ALL on SAT	Limmat	Berkmin	OKSolver

2003 Winners now known!!!

The online report will be updated with all the details of the second stage, plus an analysis of the competition. For now, one can take a look at our slides (PPT, PDF) from the SAT2003 conference and the detailled results of the first stage.

Industrial category

We have a local copy of the industrial benchmarks: sc2003benchs.zip.

Forklift, from from Eugene Goldberg and Yakov Novikov (complete deterministic) won both the "best complete solver on industrial benchmarks" and "best solver on satisfiable industrial benchmarks" awards.
Forklift is a new solver that can be viewed as an extension of Berkmin62 with limited resolution at each leaf.

Handmade category

satzoo from Niklas Een (complete deterministic) won both the "best complete solver on handmade benchmarks" and "best solver on satisfiable handmade benchmarks" awards. *Satzoo is a Chaff-like SAT-solver based on watched literals and clause recording. It was developed mainly to be able to experiment with new model checking techniques which requires a tighter integration with the SAT-solver. As such, Satzoo takes a more general view on what a ``SAT'' problem could be, and supports solving a number of related SAT-problems by an incremental SAT interface. Satzoo is also meant to encompass more types of constraints than just clauses. To prototype this, linear constraints over boolean variables were added to the solver. The front end thus supports a subset of the CPLEX lp-format along with DIMACS cnf-format. For lp-files optimization is performed towards the goal function rather than just finding a satisfying assignment.*

Random category

kcnf from Gilles Dequen and Olivier Dubois won the "best complete solver on random benchmarks" award.

unitwalk from Edward Hirsch and Arist Kojevnikov (Incomplete randomized) won the "best solver on satisfiable random benchmarks" award.
UnitWalk is a combination of unit clause elimination (particularly, the idea of Paturi, Pudlak and Zane's randomized unit clause elimination algorithm) and local search. The solver participated in the competition extends this basic algorithm with adding some of 2-resolvents using incBinSat, and mixes its random walks with WalkSAT-like walks.

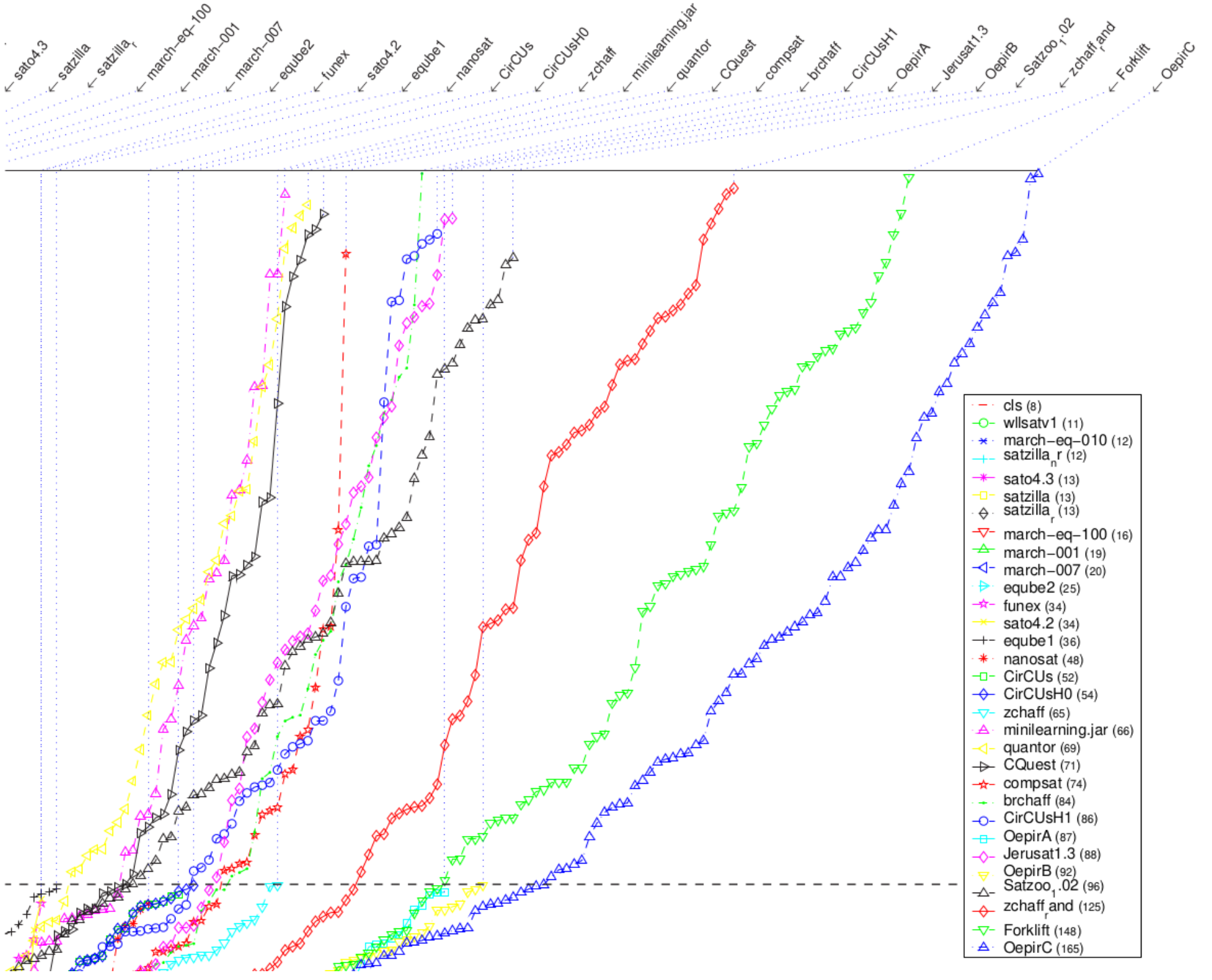
benchmarks awards

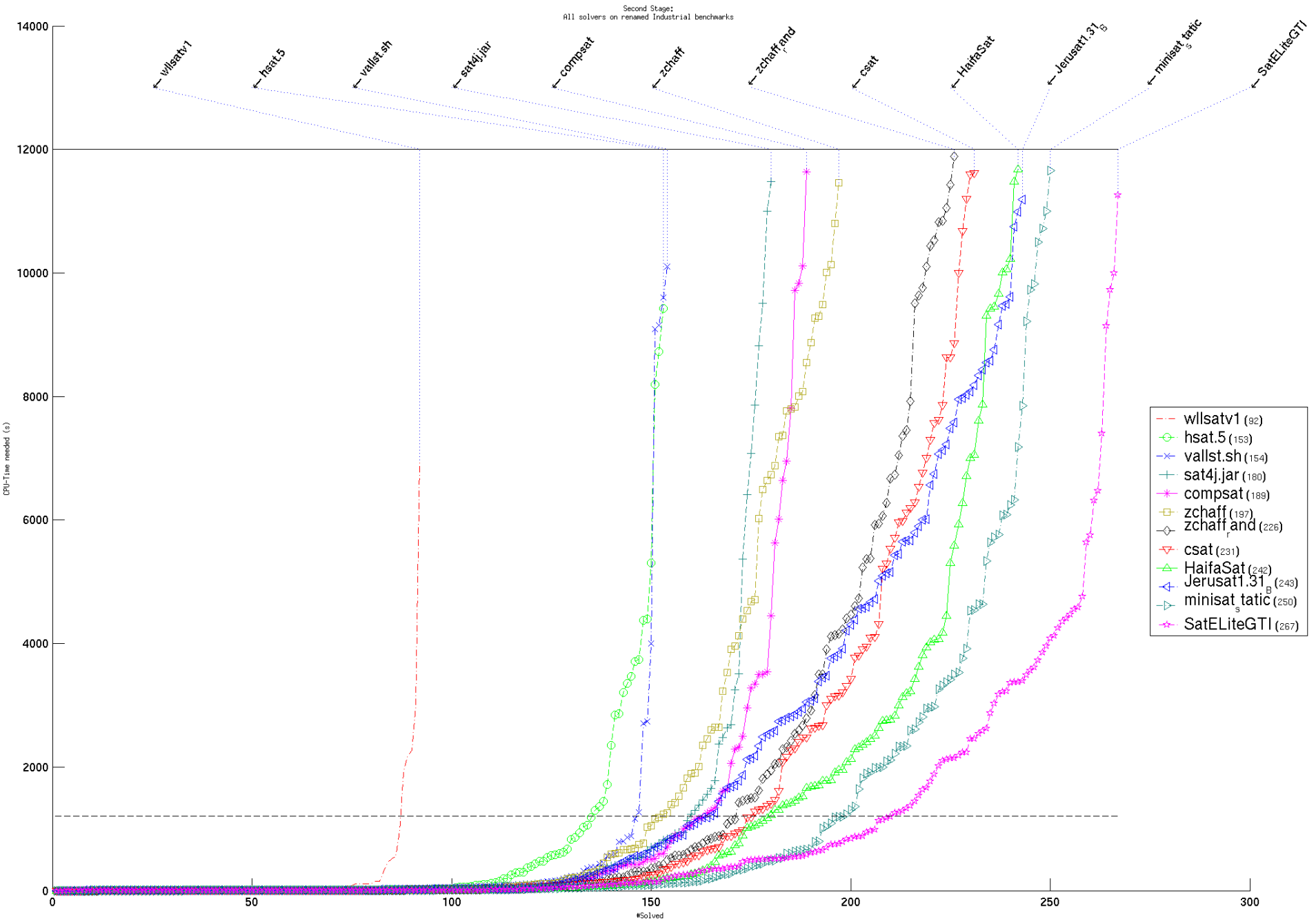
The benchmarks from the SAT2002 competition not solved **during the first stage** were used in this year competition as "SAT2002 competition challenging benchmarks". Those benchmarks were submitted last year, so they cannot be awarded this year.

Smallest unsatisfiable benchmark:
random/hirsch/hgen8/hgen8-n260-01-S1597732451 (260 variables, 391 clauses, 888 literals)

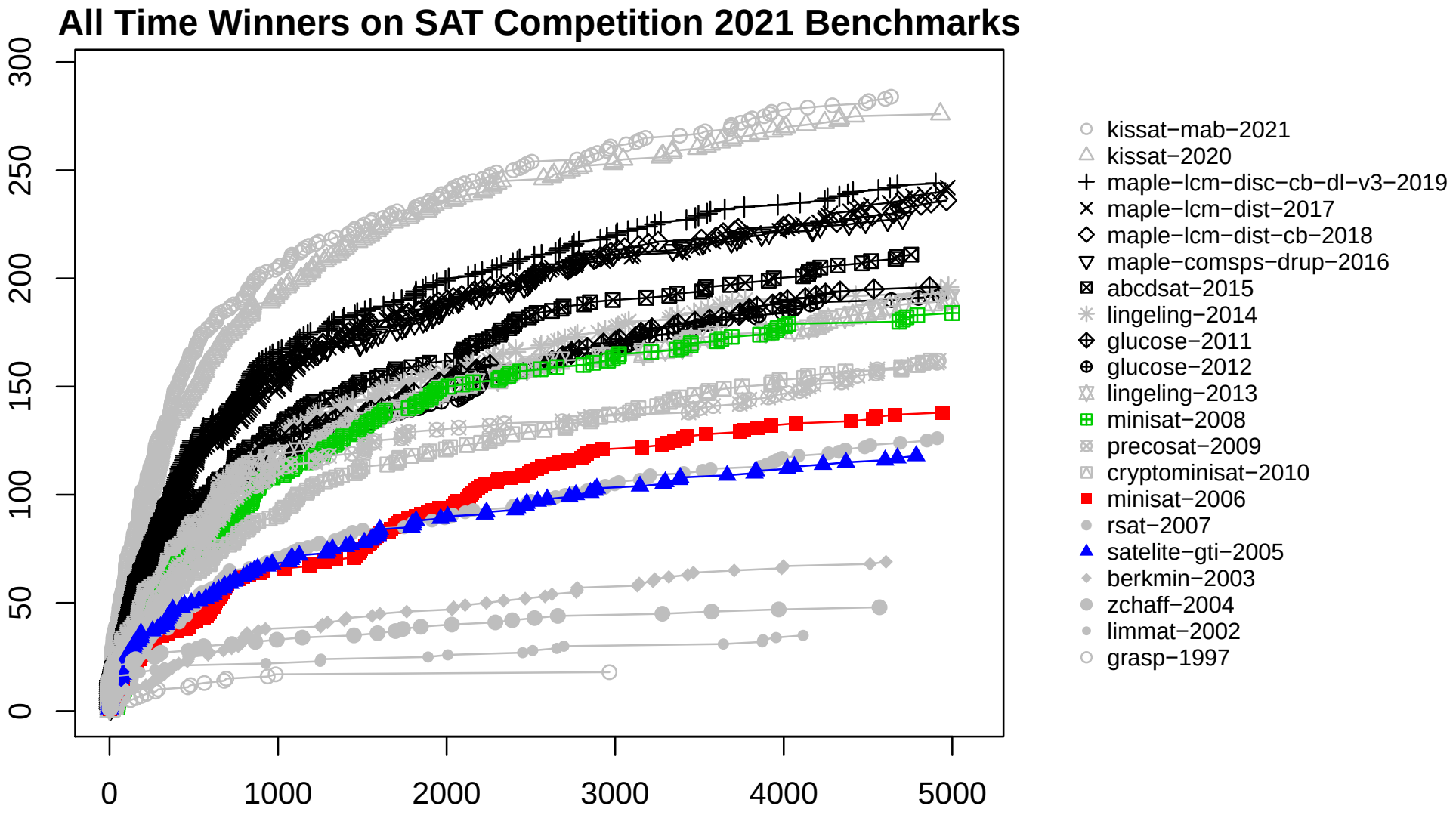
Smallest satisfiable benchmark:
random/simon/sat02-random/hgen2-v400-s161064952 (400 variables, 1400 clauses, 4200 literals). There is no award for that benchmark since it is one of the SAT2002 competition challenging benchmarks. Note that last year, unitwalk solved it during the second stage in 20199s on a PIII 450 (this year the timeout was 7200s on an Athlon 1800+).

Last update: 19 may 2003


← sato4.3
← satzilla
← satzilla$_r$
← march−eq−100
← march−001
← march−007
← eqube2
← funex
← sato4.2
← eqube1
← nanosat
← CirCUs
← CirCUsH0
← zchaff
← minilearning.jar
← quantor
← CQuest
← compsat
← brchaff
← CirCUsH1
← OepirA
← Jerusat1.3
← OepirB
← Satzoo$_1$.02
← zchaff$_r$and
← Forklift
← OepirC
cls (8)
wllsatv1 (11)
march−eq−010 (12)
satzilla$_n$r (12)
sato4.3 (13)
satzilla (13)
satzilla$_r$ (13)
march−eq−100 (16)
march−001 (19)
march−007 (20)
eqube2 (25)
funex (34)
sato4.2 (34)
eqube1 (36)
nanosat (48)
CirCUs (52)
CirCUsH0 (54)
zchaff (65)
minilearning.jar (66)
quantor (69)
CQuest (71)
compsat (74)
brchaff (84)
CirCUsH1 (86)
OepirA (87)
Jerusat1.3 (88)
OepirB (92)
Satzoo$_1$.02 (96)
zchaff$_r$and (125)
Forklift (148)
OepirC (165)


Second Stage:
All solvers on renamed Industrial benchmarks
wllsatv1
hsat.5
vallst.sh
sat4j.jar
compsat
zchaff
zchaff_and
csat
HaifaSat
Jerusat1.3_B
minisat_tatic
SatELiteGTI
CPU-Time needed (s)
#Solved
0
2000
4000
6000
8000
10000
12000
14000
0
50
100
150
200
250
300
wllsatv1 (92)
hsat.5 (153)
vallst.sh (154)
sat4j.jar (180)
compsat (189)
zchaff (197)
zchaff_and (226)
csat (231)
HaifaSat (242)
Jerusat1.3_B (243)
minisat_tatic (250)
SatELiteGTI (267)

MiniSAT Test of Time Olympic Medal FLOC'14


All Time Winners on SAT Competition 2021 Benchmarks
0
50
100
150
200
250
300
0
1000
2000
3000
4000
5000
kissat−mab−2021
kissat−2020
maple−lcm−disc−cb−dl−v3−2019
maple−lcm−dist−2017
maple−lcm−dist−cb−2018
maple−comsps−drup−2016
abcdsat−2015
lingeling−2014
glucose−2011
glucose−2012
lingeling−2013
minisat−2008
precosat−2009
cryptominisat−2010
minisat−2006
rsat−2007
satelite−gti−2005
berkmin−2003
zchaff−2004
limmat−2002
grasp−1997

The MiniSat Page

by Niklas Eén, Niklas Sörensson

Introduction

MiniSat is a minimalistic, open-source SAT solver, developed to help researchers and developers alike to get started on SAT. It is released under the MIT licence, and is currently used in a number of projects (see "Links"). On this page you will find binaries, sources, documentation and projects related to MiniSat, including the Pseudo-boolean solver MiniSat+ and the CNF minimizer/preprocessor SatELite. Together with SatELite, MiniSat was recently awarded in the three industrial categories and one of the "crafted" categories of the SAT 2005 competition (see picture).

Some key features of MiniSat:

- **Easy to modify.** MiniSat is small and well-documented, and possibly also well-designed, making it an ideal starting point for adapting SAT based techniques to domain specific problems.
- **Highly efficient.** Winning all the industrial categories of the SAT 2005 competition, MiniSat is a good starting point both for future research in SAT, and for applications using SAT.
- **Designed for integration.** MiniSat supports incremental SAT and has mechanisms for adding non-clausal constraints. By virtue of being easy to modify, it is a good choice for integrating as a backend to another tool, such as a model checker or a more generic constraint solver.

We would like to start a community to help distribute knowledge about how to use and modify MiniSat. Any questions, comments, bugreports, bugfixes etc. should be directed to minisat@googlegroups.com. The archives can be browsed here. The source code repository for MiniSat 2.2 can be found at github.

— Niklas & Niklas

MiniSat papers...

" An Extensible SAT-solver "
Niklas Een, Niklas Sörensson, SAT 2003.

" MiniSat — A SAT Solver with Conflict-Clause Minimization "
Niklas Een, Niklas Sörensson, poster for SAT 2005.

" Effective Preprocessing in SAT through Variable and Clause Elimination "
Niklas Een, Armin Biere, SAT 2005.

" Translating Pseudo-Boolean Constraints into SAT "
Niklas Een, Niklas Sörensson, JSAT 2006.

" Applying Logic Synthesis for Speeding Up SAT "
Niklas Een, Alan Mishchenko, Niklas Sörensson, SAT 2007.

" Cut Sweeping "
Niklas Een, Cadence Technical Report, 2007.

Applications of MiniSat...

" Temporal Induction by Incremental SAT Solving "
Niklas Een, Niklas Sörensson, BMC 2003.

" New techniques that improve MACE-style finite model finding "
Koen Claessen, Niklas Sörensson, CADE 2003.

" A Single-Instance Incremental SAT Formulation of Proof- and Counterexample-Based Abstraction "
Niklas Een, Alan Mishchenko, Nina Amla, FMCAD 2010.

Conclusion

- High impact paper
  - blue-print for high-performance SAT Solvers
  - Niklas Eén dissertation ”SAT Based Model Checking” Chalmers 2005
  - Niklas Sörensson dissertation ”Effective SAT Solving” Chalmers 2008
- High impact source code
  - used in many industrial SAT applications (EDA, model checking, SMT, . . .)
  - base line of many award winning solvers in the SAT competition
- Contributions
  - Exponential VSIDS (contributed to Niklas Eén)
  - Lazy Priority Decision Queue (contributed to Niklas Sörensson)
  - Assumption Interface for incremental SAT solving