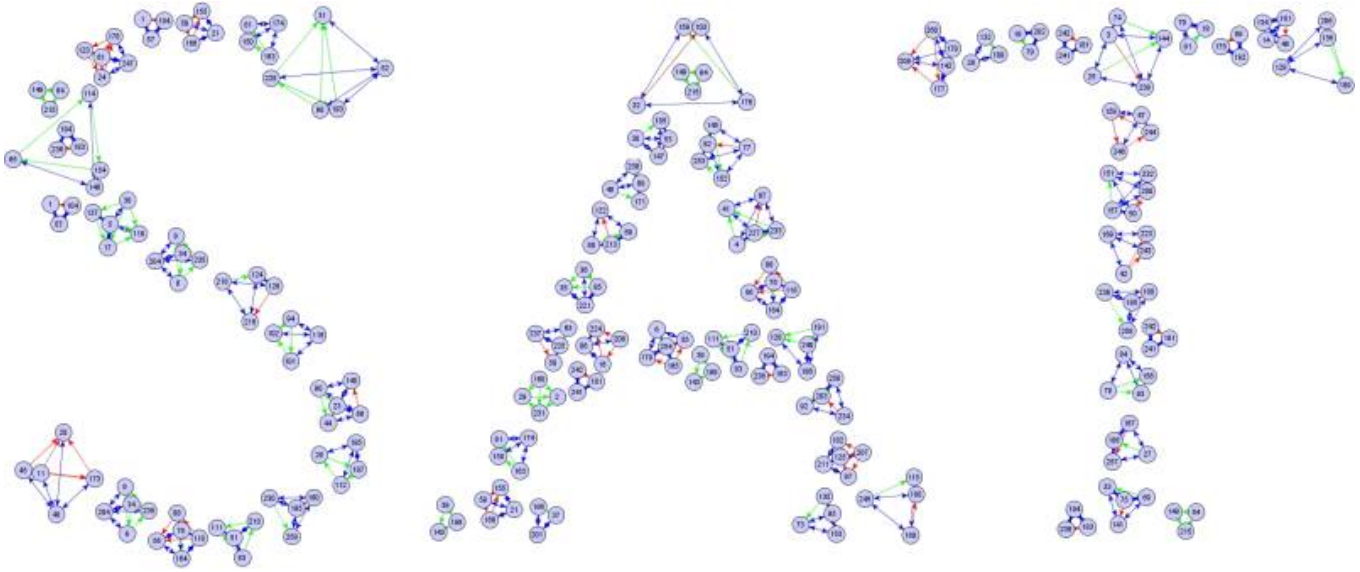


C O N T E S T

Some Results and Lessons

Laurent Simon

simon@lri.fr
LRI - CNRS
Université Paris Sud 11,

France

and

Daniel le Berre

leberre@cril.univ-artois.fr
CRIL - CNRS
Université d'Artois,

France

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?

If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is SAT?

The SAT Problem

Given a propositional formula f in conjunctive normal form, is it satisfiable?
If so, give its assignement.

The Canonical NP-Complet problem

Studied by Computer scientist, Mathematicians, Physicians, Engineers, Industrial recruters...

Found in

- Random problems, Planning, Constraints, Chip Design
- Ice

What is a Competition?

It's all about an evaluation

Evaluating Solver is an easy part?

It brings more problems than solutions

Competition principles

Fixes the best rules as we can

Find the best solver according to this rules

All is said?No (I hope)

What is a Competition?

It's all about an evaluation

Evaluating Solver is an easy part?

It brings more problems than solutions

Competition principles

Fixes the best rules as we can

Find the best solver according to this rules

All is said?No (I hope)

What is a Competition?

It's all about an evaluation

Evaluating Solver is an easy part?

It brings more problems than solutions

Competition principles

Fixes the best rules as we can
Find the best solver according to this rules

All is said?No (I hope)

What is a Competition?

It's all about an evaluation

Evaluating Solver is an easy part?

It brings more problems than solutions

Competition principles

Fixes the best rules as we can
Find the best solver according to this rules

All is said? No (I hope)

What is a Competition?

It's all about an evaluation

Evaluating Solver is an easy part?

It brings more problems than solutions

Competition principles

Fixes the best rules as we can
Find the best solver according to this rules

All is said?No (I hope)

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!
On two (very) different architectures

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!

On two (very) different architectures

The First SAT experimentation

Computer vs Human

« The superiority of the present procedure (i.e. DP) over those previously available is indicated in part by the fact that a formula on which Gilmore's routine for the IBM 704 causes the machine to compute for 21 minutes without obtaining a result was worked successfully by hand computation using the present method in 30 minutes » [Davis and Putnam, 1960]

« The well-formed formula (...) which was beyond the scope of Gilmore's program was proved in under two minutes with the present program » [Davis et al., 1962]

The SAT Contest started 45 years ago!
On two (very) different architectures

SAT'05 Contest Roadmap

1. **Competition Historics**
   - SAT Contest over the years
2. **Rules... and errors**
   - The SAT'02-04 contest
3. **SAT'05 Contest Results**
   - A new scoring scheme: SAT'05 contest
   - Ways of presenting results
4. **Lessons**
   - Lessons
   - Conclusion
   - Bibliography

The next serious comparison

First Cycle of SAT Contest

First Cycle of SAT Contest

1991/1992	Paderborn
1992/1993	Second DIMACS Challenge
1996	Beijing Competition

And then a gap of five years...

All the Dimacs benchmarks were still considered hard in the end of the 1990's... while beeing very easy for some solvers.

The next serious comparison

First Cycle of SAT Contest

First Cycle of SAT Contest

1991/1992	Paderborn
1992/1993	Second DIMACS Challenge
1996	Beijing Competition

And then a gap of five years...

All the Dimacs benchmarks were still considered hard in the end of the 1990's... while beeing very easy for some solvers.

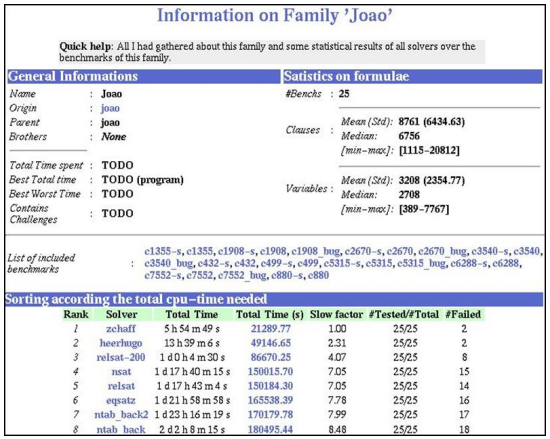
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	**Joao**
Origin	joao
Parent	**joao**
Brothers	***None***
Total Time spent	**TODO**
Best Total time	**TODO (program)**
Best Worst Time	**TODO**
Contains Challenges	**TODO**

Satistics on formulae

#Benchs		**25**
Clauses	*Mean (Std):*	**8761 (6434.63)**
	Median:	**6756**
	[min–max]:	**[1115–20812]**
Variables	*Mean (Std):*	**3208 (2354.77)**
	Median:	**2708**
	[min–max]:	**[389–7767]**

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerlugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

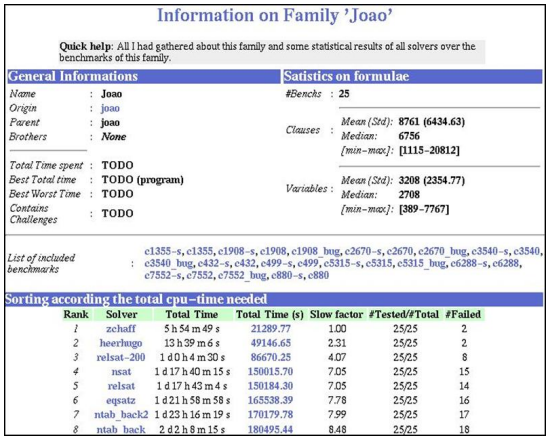
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	**Joao**
Origin	joao
Parent	**joao**
Brothers	***None***
Total Time spent	**TODO**
Best Total time	**TODO (program)**
Best Worst Time	**TODO**
Contains Challenges	**TODO**

Satistics on formulae

#Benchs		**25**
Clauses	*Mean (Std):*	**8761 (6434.63)**
	Median:	**6756**
	[min–max]:	**[1115–20812]**
Variables	*Mean (Std):*	**3208 (2354.77)**
	Median:	**2708**
	[min–max]:	**[389–7767]**

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerlugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

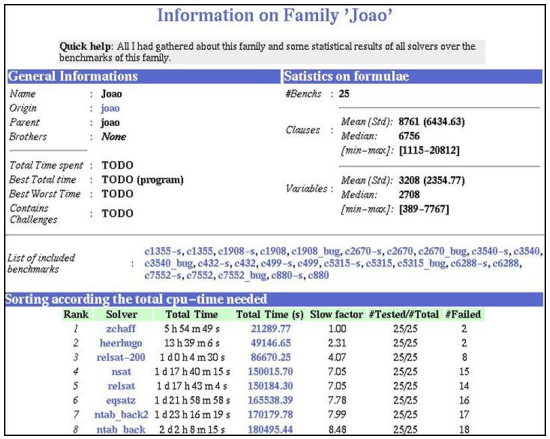
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	:	**Joao**
Origin	:	joao
Parent	:	**joao**
Brothers	:	***None***
Total Time spent	:	**TODO**
Best Total time	:	**TODO (program)**
Best Worst Time	:	**TODO**
Contains Challenges	:	**TODO**

Satistics on formulae

#Benchs	:		**25**
Clauses	:	*Mean (Std):*	**8761 (6434.63)**
		Median:	**6756**
		[min–max]:	**[1115–20812]**
Variables	:	*Mean (Std):*	**3208 (2354.77)**
		Median:	**2708**
		[min–max]:	**[389–7767]**

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerlugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

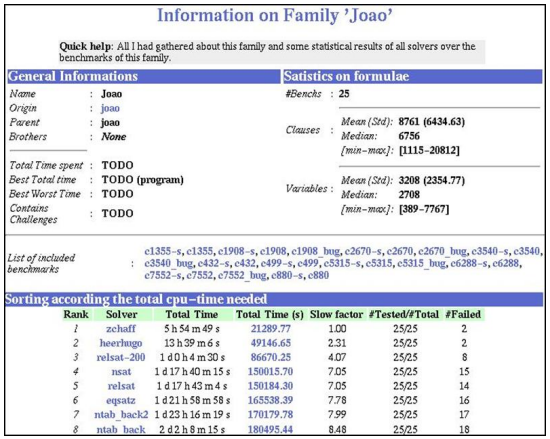
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	**Joao**
Origin	joao
Parent	**joao**
Brothers	***None***
Total Time spent	**TODO**
Best Total time	**TODO (program)**
Best Worst Time	**TODO**
Contains Challenges	**TODO**

Satistics on formulae

#Benchs		**25**
Clauses	*Mean (Std):*	**8761 (6434.63)**
	Median:	**6756**
	[min–max]:	**[1115–20812]**
Variables	*Mean (Std):*	**3208 (2354.77)**
	Median:	**2708**
	[min–max]:	**[389–7767]**

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerlugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

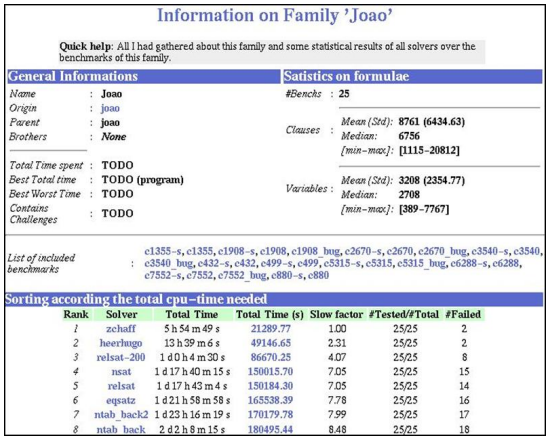
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	**Joao**
Origin	joao
Parent	**joao**
Brothers	***None***
Total Time spent	**TODO**
Best Total time	**TODO (program)**
Best Worst Time	**TODO**
Contains Challenges	**TODO**

Satistics on formulae

#Benchs		**25**
Clauses	*Mean (Std):*	**8761 (6434.63)**
	Median:	**6756**
	[min–max]:	**[1115–20812]**
Variables	*Mean (Std):*	**3208 (2354.77)**
	Median:	**2708**
	[min–max]:	**[389–7767]**

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerlugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

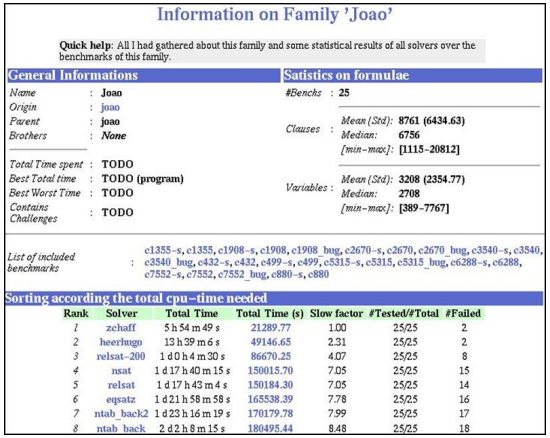
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	Joao
Origin	joao
Parent	joao
Brothers	*None*
Total Time spent	TODO
Best Total time	TODO (program)
Best Worst Time	TODO
Contains Challenges	TODO

Satistics on formulae

#Benchs		25
Clauses	*Mean (Std):*	8761 (6434.63)
	Median:	6756
	[min–max]:	[1115–20812]
Variables	*Mean (Std):*	3208 (2354.77)
	Median:	2708
	[min–max]:	[389–7767]

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerhugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

Aims:

- What are the good solvers?
- What are the hard benchmarks?
- What is a good evaluation?

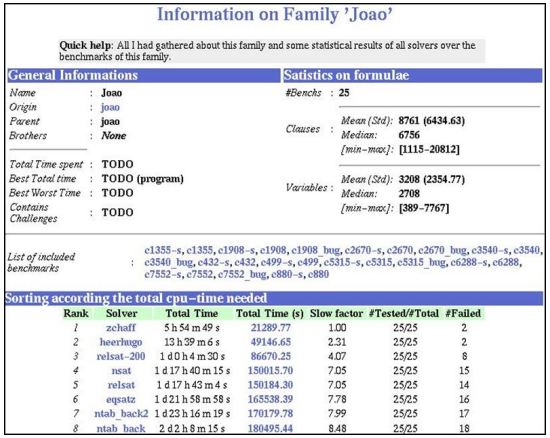
SatEx Experience

Back to the end of the 1990's

Doing my bibliography I faced a problem

- All papers presented brand new (efficient) methods
- All solvers were the best, ever
- Where were the bad results?
- What could I do with my spare CPU time?

SatEx was based on my single (dual) computer



Information on Family 'Joao'

Quick help: All I had gathered about this family and some statistical results of all solvers over the benchmarks of this family.

General Informations

Name	Joao
Origin	joao
Parent	joao
Brothers	*None*
Total Time spent	TODO
Best Total time	TODO (program)
Best Worst Time	TODO
Contains Challenges	TODO

Satistics on formulae

#Benchs		25
Clauses	*Mean (Std):*	8761 (6434.63)
	Median:	6756
	[min–max]:	[1115–20812]
Variables	*Mean (Std):*	3208 (2354.77)
	Median:	2708
	[min–max]:	[389–7767]

List of included benchmarks : c1355–s, c1355, c1908–s, c1908, c1908_bug, c2670–s, c2670, c2670_bug, c3540–s, c3540, c3540_bug, c432–s, c432, c499–s, c499, c5315–s, c5315, c5315_bug, c6288–s, c6288, c7552–s, c7552, c7552_bug, c880–s, c880

Sorting according the total cpu–time needed

Rank	Solver	Total Time	Total Time (s)	Slow factor	#Tested/#Total	#Failed
1	zchaff	5 h 54 m 49 s	21289.77	1.00	25/25	2
2	heerhugo	13 h 39 m 6 s	49146.65	2.31	25/25	2
3	relsat–200	1 d 0 h 4 m 30 s	86670.25	4.07	25/25	8
4	nsat	1 d 17 h 40 m 15 s	150015.70	7.05	25/25	15
5	relsat	1 d 17 h 43 m 4 s	150184.30	7.05	25/25	14
6	eqsatz	1 d 21 h 58 m 58 s	165538.39	7.78	25/25	16
7	ntab_back2	1 d 23 h 16 m 19 s	170179.78	7.99	25/25	17
8	ntab_back	2 d 2 h 8 m 15 s	180495.44	8.48	25/25	18

Problems:

- Adding benchmark is hard
- How to integrate new computers?

Around the SAT World

Running solvers on your PC(s)
can make you travel the world

A lot of room left for future competitions

Around the SAT World

Running solvers on your PC(s)
can make you travel the world

A lot of room left for future competitions

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

Fundations of the contest

2001: The SAT face is changing

- **zchaff** [Moskewicz et al., 2001] on Industrials
- **kcnfs** [Dubois and Dequen, 2001] on Random

Motivations

- Clarify which methods works
- Make *official* snapshot, five years later
- Maintain a pool of hard, interesting benchmarks
- Localize promizing methods
- Make progress in the way of comparing solvers
- More political issues: motivate the field
  - Inside (students, ...)
  - Outside (industrials, ...)

SAT'05 Contest Roadmap

1. Competition Historics
   - SAT Contest over the years
2. **Rules... and errors**
   - The SAT'02-04 contest
3. SAT'05 Contest Results
   - A new scoring scheme: SAT'05 contest
   - Ways of presenting results
4. Lessons
   - Lessons
   - Conclusion
   - Bibliography

The big question

What is a good SAT Solver?

– One that participates

The big question

What is a good SAT Solver?

– *One that participates*

SAT'02 Contest Fundation ideas

The big idea:

- Award solvers that solve a wide variety of benchmarks
- Run solvers with a very large time out

Dealing with categories

- We had to separate **Solvers**:
  - Complete (SAT+UNSAT) solvers
  - Uncomplete (one-sided, SAT) solvers

  And:
  - Randomized (need a seed, expected behavior)
  - Deterministic
- And **Benchmarks**:
  - Random
  - Industrial
  - the others (tricky formulae, toys, ...)

SAT'02 Contest Fundation ideas

The big idea:

- Award solvers that solve a wide variety of benchmarks
- Run solvers with a very large time out

Dealing with categories

- We had to separate **Solvers**:
  - Complete (SAT+UNSAT) solvers
  - Uncomplete (one-sided, SAT) solvers

  And:
  - Randomized (need a seed, expected behavior)
  - Deterministic
- And **Benchmarks**:
  - Random
  - Industrial
  - the others (tricky formulae, toys, ...)

SAT'02 Contest Fundation ideas

The big idea:

- Award solvers that solve a wide variety of benchmarks
- Run solvers with a very large time out

Dealing with categories

- We had to separate **Solvers**:
  - Complete (SAT+UNSAT) solvers
  - Uncomplete (one-sided, SAT) solvers

  And:
  - Randomized (need a seed, expected behavior)
  - Deterministic
- And **Benchmarks**:
  - Random
  - Industrial
  - the others (tricky formulae, toys, ...)

What can be compared?

Complete vs Uncomplete?

Uncomplete (SAT) solvers can be compared with Complete on SAT instances only.

Randomized vs Deterministic?

Expected behavior of randomized algorithm (multiple launchs) can be compared with one launch of deterministics algorithms. **We have to launch randomized algorithm several times**

We had 6 Solver Prices

For each benchmark category, we have 2 price:

- Complete on All benchmarks
- All solvers on SAT benchmarks

What can be compared?

Complete vs Uncomplete?

Uncomplete (SAT) solvers can be compared with Complete on SAT instances only.

Randomized vs Deterministic?

Expected behavior of randomized algorithm (multiple launchs) can be compared with one launch of deterministics algorithms.
We have to launch randomized algorithm several times

We had 6 Solver Prices

For each benchmark category, we have 2 price:

- Complete on All benchmarks
- All solvers on SAT benchmarks

What can be compared?

Complete vs Uncomplete?

Uncomplete (SAT) solvers can be compared with Complete on SAT instances only.

Randomized vs Deterministic?

Expected behavior of randomized algorithm (multiple launchs) can be compared with one launch of deterministics algorithms. **We have to launch randomized algorithm several times**

We had 6 Solver Prices

For each benchmark category, we have 2 price:

- Complete on All benchmarks
- All solvers on SAT benchmarks

Restricting the number of solvers

We have to restrict the number of solvers

- Selecting the TOP-N solvers imply to have one variant
- Submitters may just send a solver just to test it

The Timeout is flexible

We first have a minimal number of benchmarks,
then a maximal number of solvers,
then we compute the *timeout* to use.

If some cpu remains, add some benchmarks

Any number of solvers may enter as Hors-Concours
not guaranted to be finished at the end of the contest
(dropped in case of emergency)

Restricting the number of solvers

We have to restrict the number of solvers

- Selecting the TOP-N solvers imply to have one variant
- Submitters may just send a solver just to test it

The Timeout is flexible

We first have a minimal number of benchmarks,
then a maximal number of solvers,
then we compute the *timeout* to use.

If some cpu remains, add some benchmarks

Any number of solvers may enter as Hors-Concours
not guaranted to be finished at the end of the contest
(dropped in case of emergency)

Restricting the number of solvers

We have to restrict the number of solvers

- Selecting the TOP-N solvers imply to have one variant
- Submitters may just send a solver just to test it

The Timeout is flexible

We first have a minimal number of benchmarks,
then a maximal number of solvers,
then we compute the *timeout* to use.

If some cpu remains, add some benchmarks

Any number of solvers may enter as Hors-Concours
not guaranted to be finished at the end of the contest
(dropped in case of emergency)

Restricting the number of benchmarks

Do not restrict submitters!

After submissions, benchmarks too similar to known ones were discarded by organizers.

Constraint: Equilibrate them between categories/series.

In the SAT'05 contest

Benchmark specialist helped us:

- Oliver Kullman for uniform-random instances
- Armin Biere and Emmanuel Zarpas for Industrial ones

What can we count?

Recall: Award solvers that solve a wide variety of benchmarks

Do not count CPU-Time (well, not directly)

The winner is the solver that solves the maximum number of series, independently of the time needed.

Series: Similar benchmarks of different scales

- same generator (same parameters but the size)
- same problems

A series is solved if at least one of its benchmarks is solved. Ties are breaked by the number of solved benchmarks, then by the total CPU time needed.

SAT'02 Rules

A 3-stage Contest

1. Check I/O solver compliance
2. Run all solvers on all benchmarks
3. Run top-5 solvers on the hardest benchmarks, with a very large cpu time

- All benchmarks are shuffled
- Top-5 solvers are decided for each of the 6 categories
- The third stage run during the conference

8 Prices

- 6 prices for solvers,
- 2 prices for the smallest unsolved benchmarks

SAT'02 Rules

A 3-stage Contest

1. Check I/O solver compliance
2. Run all solvers on all benchmarks
3. Run top-5 solvers on the hardest benchmarks, with a very large cpu time

- All benchmarks are shuffled
- Top-5 solvers are decided for each of the 6 categories
- The third stage run during the conference

8 Prices

- 6 prices for solvers,
- 2 prices for the smallest unsolved benchmarks

SAT'02 Rules

A 3-stage Contest

1. Check I/O solver compliance
2. Run all solvers on all benchmarks
3. Run top-5 solvers on the hardest benchmarks, with a very large cpu time

- All benchmarks are shuffled
- Top-5 solvers are decided for each of the 6 categories
- The third stage run during the conference

8 Prices

- 6 prices for solvers,
- 2 prices for the smallest unsolved benchmarks

SAT'02 Problems

Source code submission

- To be able to track different versions of solvers when buggy solver returned
- To have clear compiler versions

A submitter send us *Machine-Code source*
(output of a Proprietary Compiler)

We used a heuristic to solve CPU Time

If a solver does not solve a benchmark of size X, do not try it on size $X + 1$

Wrong hypothesis, especially on BMC benchmarks

SAT'02 Problems

Source code submission

- To be able to track different versions of solvers when buggy solver returned
- To have clear compiler versions

A submitter send us *Machine-Code source*
(output of a Proprietary Compiler)

We used a heuristic to solve CPU Time

If a solver does not solve a benchmark of size X, do not try it on size $X + 1$

Wrong hypothesis, especially on BMC benchmarks

SAT'02 Problems

Source code submission

- To be able to track different versions of solvers when buggy solver returned
- To have clear compiler versions

A submitter send us *Machine-Code source*
(output of a Proprietary Compiler)

We used a heuristic to solve CPU Time

If a solver does not solve a benchmark of size X, do not try it on size $X + 1$

Wrong hypothesis, especially on BMC benchmarks

SAT'02 Problems

Source code submission

- To be able to track different versions of solvers when buggy solver returned
- To have clear compiler versions

A submitter send us *Machine-Code source*
(output of a Proprietary Compiler)

We used a heuristic to solve CPU Time

If a solver does not solve a benchmark of size X, do not try it on size $X + 1$
Wrong hypothesis, especially on BMC benchmarks

How to deal with bugs

Real Bugs

First aids against bugs

If a solver crashs (sigfault for instance) then relaunch it.
If the pool of crashs decreases, repeat the process on the remaining crashes untill some fixed point is reached.

Only method to deal with solver bugs

Let submitters follow the results lively, they know how their solver behaves. We provide, daily:

- CPU time, answers and complete traces
- Size of benchmarks
- Relative performances of other solver on this benchmark

When submitter ask us, we may send the problematic benchmark

How to deal with bugs

Real Bugs

First aids against bugs

If a solver crashs (sigfault for instance) then relaunch it.
If the pool of crashs decreases, repeat the process on the remaining crashes untill some fixed point is reached.

Only method to deal with solver bugs

Let submitters follow the results lively, they know how their solver behaves. We provide, daily:

- CPU time, answers and complete traces
- Size of benchmarks
- Relative performances of other solver on this benchmark

When submitter ask us, we may send the problematic benchmark

How to deal with bugs

Buggy solvers

What is a *buggy* solver?

All solvers crash on some instance. Are they all buggy?
A buggy solver is a solver that answer UNSAT on a SAT instance

Main principle: Buggy solvers are kept, but ran *Hors-Concours*.

Buggy solver to one-sided solver

Apply this patch:

- If the answer is SAT, check the certificate
  - Correct: answer *SAT*
  - Uncorrect: answer *Unknown*
- If the answer is *UNSAT*, answer *Unknown*

Lessons from SAT'02 Contest

1. Do not have a second stage during the conference
2. (very) Large CpuTime does not really matter. Only a very few benchmarks were solved in the last round.
3. Random must look like random instances.
4. Do not have strict rules for second stage solvers. But how to choose solvers entering the second stage?
5. Be able to be flexible with not-really bugs? (internal timeouts, data structure size limits, ...)

Lessons from SAT'02 Contest

1. Do not have a second stage during the conference
2. (very) Large CpuTime does not really matter. Only a very few benchmarks were solved in the last round.
3. Random must look like random instances.
4. Do not have strict rules for second stage solvers. But how to choose solvers entering the second stage?
5. Be able to be flexible with not-really bugs? (internal timeouts, data structure size limits, ...)

Lessons from SAT'02 Contest

1. Do not have a second stage during the conference
2. (very) Large CpuTime does not really matter. Only a very few benchmarks were solved in the last round.
3. Random must look like random instances.
4. Do not have strict rules for second stage solvers. But how to choose solvers entering the second stage?
5. Be able to be flexible with not-really bugs? (internal timeouts, data structure size limits, . . .)

Lessons from SAT'02 Contest

1. Do not have a second stage during the conference
2. (very) Large CpuTime does not really matter. Only a very few benchmarks were solved in the last round.
3. Random must look like random instances.
4. Do not have strict rules for second stage solvers. But how to choose solvers entering the second stage?
5. Be able to be flexible with not-really bugs? (internal timeouts, data structure size limits, . . .)

Lessons from SAT'02 Contest

1. Do not have a second stage during the conference
2. (very) Large CpuTime does not really matter. Only a very few benchmarks were solved in the last round.
3. Random must look like random instances.
4. Do not have strict rules for second stage solvers. But how to choose solvers entering the second stage?
5. Be able to be flexible with not-really bugs? (internal timeouts, data structure size limits, . . .)

SAT'03: Introducing the judges

Decisions have to be objective... and human

The judges

Judges take decisions on an anonymised basis. They are thus free to take *good* decisions.

Other modified rules

- Random instances must look like random instances (hidden formulae are allowed)
- Only 3 generated instances per point
- Last year winner automatically reenter the competition

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...

– but based on zchaff architecture

Clear progress in the performances

– but mainly due to simple bug correction

Progress in the random category

– but best known solvers were still not submitted

Clear winner in the industrial category

– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Competition

A one-slide results summary

A large number of brand new solver...
– but based on zchaff architecture

Clear progress in the performances
– but mainly due to simple bug correction

Progress in the random category
– but best known solvers were still not submitted

Clear winner in the industrial category
– but no description available

SAT'03 Problems

The *lisa* syndrom

Shuffling a benchmark may completly change the result.

How to win the SAT'03 Contest

- Claim to be a randomized complete solver
- Shuffle the benchmark, then solve it

Lots of solvers have exponential distribution of their CPU-time on the same, shuffled benchmark

SAT'03 Problems

The *lisa* syndrom

Shuffling a benchmark may completly change the result.

How to win the SAT'03 Contest

- Claim to be a randomized complete solver
- Shuffle the benchmark, then solve it

Lots of solvers have exponential distribution of their CPU-time on the same, shuffled benchmark

SAT'03 Problems

The *lisa* syndrom

Shuffling a benchmark may completly change the result.

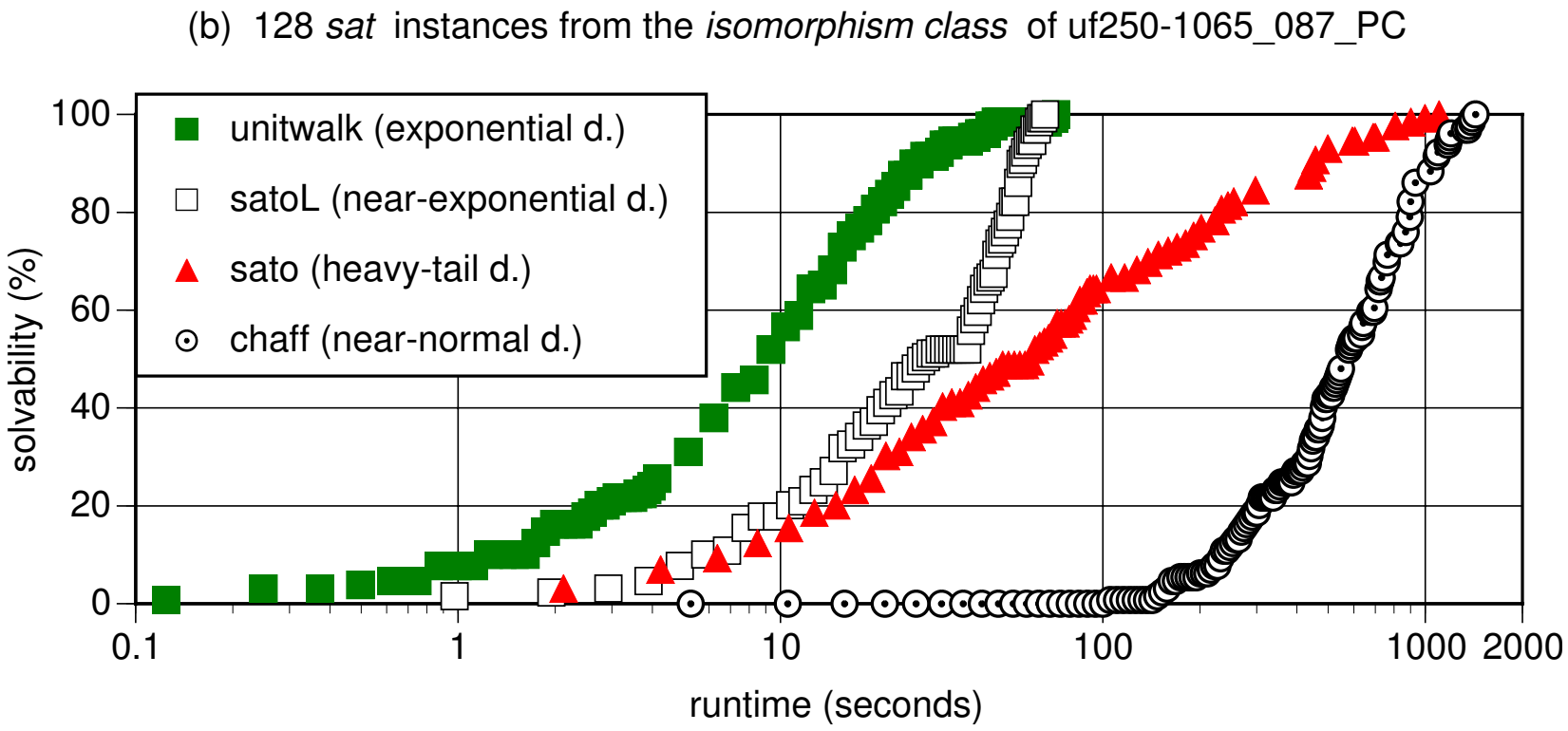
How to win the SAT'03 Contest

- Claim to be a randomized complete solver
- Shuffle the benchmark, then solve it

Lots of solvers have exponential distribution of their CPU-time on the same, shuffled benchmark

SAT'03 Problems

The *lisa* syndrom, studied [Brglez et al., 2003]



Picture from [Brglez et al., 2003]

Lessons from SAT'03 Contest

- Random must be only pure random instances
- Do not launch 3 times randomized solvers
- Shuffling instances may be a problem
- Do not allow black boxes (like `Forklift`) to win the contest
- Stop the first I/O compliance stage (too close to debugging)

Lessons from SAT'03 Contest

- Random must be only pure random instances
- Do not launch 3 times randomized solvers
- Shuffling instances may be a problem
- Do not allow black boxes (like `Forklift`) to win the contest
- Stop the first I/O compliance stage (too close to debugging)

Lessons from SAT'03 Contest

- Random must be only pure random instances
- Do not launch 3 times randomized solvers
- Shuffling instances may be a problem
- Do not allow black boxes (like `Forklift`) to win the contest
- Stop the first I/O compliance stage (too close to debugging)

Lessons from SAT'03 Contest

- Random must be only pure random instances
- Do not launch 3 times randomized solvers
- Shuffling instances may be a problem
- Do not allow black boxes (like `Forklift`) to win the contest
- Stop the first I/O compliance stage (too close to debugging)

Lessons from SAT'03 Contest

- Random must be only pure random instances
- Do not launch 3 times randomized solvers
- Shuffling instances may be a problem
- Do not allow black boxes (like `Forklift`) to win the contest
- Stop the first I/O compliance stage (too close to debugging)

SAT'04: no Black-Boxes

Anti Black-Box rule

Submitters that do not send a 2-page description paper of their solver are *Hors-Concours*.

Ensures that we learn something from the contest

Do not shuffle all benchmarks

Some solvers performed much better on original formulae:

- Industrial solvers exploits the locality of clauses and literals
- Some submitters were planning to de-shuffle benchmarks before trying to solve them

One industrial series gives 3 series: 1 original and 2 shuffled

SAT'04: no Black-Boxes

Anti Black-Box rule

Submitters that do not send a 2-page description paper of their solver are *Hors-Concours*.

Ensures that we learn something from the contest

Do not shuffle all benchmarks

Some solvers performed much better on original formulae:

- Industrial solvers exploits the locality of clauses and literals
- Some submitters were planning to de-shuffle benchmarks before trying to solve them

One industrial series gives 3 series: 1 original and 2 shuffled

SAT'04: no Black-Boxes

Anti Black-Box rule

Submitters that do not send a 2-page description paper of their solver are *Hors-Concours*.

Ensures that we learn something from the contest

Do not shuffle all benchmarks

Some solvers performed much better on original formulae:

- Industrial solvers exploits the locality of clauses and literals
- Some submitters were planning to de-shuffle benchmarks before trying to solve them

One industrial series gives 3 series: 1 original and 2 shuffled

Lessons from SAT'04 Contest

Efficient local search made some benchs trivials

Random instances were generated according to SAT'03 solver performances.
But the introduction of very efficient local search algorithm (`Novelty+`) have made the SAT, Random benchmarks trivial.

No more Random SAT benchs left for the second stage!

Other lessons...

- Random instances must be generated according to actual, best, solver performances
- To be *industrial relevant*: larger timeout

SAT'05 Contest Roadmap

1. Competition Historics
   - SAT Contest over the years
2. Rules... and errors
   - The SAT'02-04 contest
3. **SAT'05 Contest Results**
   - A new scoring scheme: SAT'05 contest
   - Ways of presenting results
4. Lessons
   - Lessons
   - Conclusion
   - Bibliography

What's new in the rules

The Anti-Black-Boxes rule, reinforced

Competition and *Demonstration* divisions

Competition the source code of the solver must be available after the competition.

Demonstration a binary version of the solver must be available for research purpose.

Principles

Participation to the competition must benefit to the community

- By providing source code, binary or benchmarks
- By supporting the conference and the competition

The new scoring scheme

New Scoring Scheme [van Gelder, 2005]

Benchmark purse to be divided equally among the solvers able to solve it.

Speed purse to be divided unequally among the solvers able to solve a given benchmark.

Series an extra credit is given for each series solved.

Solver his score is the sum of the credits obtained per benchmarks solved.

The new award scheme: 27 awards

- Three categories: industrial, crafted and random
- Three specialties: SAT, UNSAT and SAT+UNSAT
- Three medals: gold, silver and bronze

The new scoring scheme

New Scoring Scheme [van Gelder, 2005]

Benchmark purse to be divided equally among the solvers able to solve it.

Speed purse to be divided unequally among the solvers able to solve a given benchmark.

Series an extra credit is given for each series solved.

Solver his score is the sum of the credits obtained per benchmarks solved.

The new award scheme: 27 awards

- Three categories: industrial, crafted and random
- Three specialties: SAT, UNSAT and SAT+UNSAT
- Three medals: gold, silver and bronze

Past and New rules

Restricting the number of solvers in competition division

- First stage: 3 solvers per submitter
- Second stage:1 solver per submitter

Other rules...

- Anti Black Boxes rules, reinforced
- Introduction of the Demonstration division
- Industrial are ranked on a non-shuffled basis (second stage)
- Relaunch all first-stage timeouts in the second stage
- Back to a very-long timeout (12,000s for industrial benchs.)

Past and New rules

Restricting the number of solvers in competition division

- First stage: 3 solvers per submitter
- Second stage:1 solver per submitter

Other rules...

- Anti Black Boxes rules, reinforced
- Introduction of the Demonstration division
- Industrial are ranked on a non-shuffled basis (second stage)
- Relaunch all first-stage timeouts in the second stage
- Back to a very-long timeout (12,000s for industrial benchs.)

SAT'05 New scoring scheme

Example on Industrial UNSAT specialty

satELiteGTI	**27,518**	**87**
zchaff-rand	26,792	94
haifaSat	23,666	91
minisat	19,863	84
csat	15,892	91
zchaff	13,829	76
jerusat-B	10,225	80
hsat-5	10,029	54
vallst	9,192	69
compsat	9,097	75
sat4j	8,654	70
wllsatv1	1,053	6

Table: The winner is now a clear winner

High cost of the contest

Hard time for submitters and organizers

We had to keep ensuring that is worth organizing it.

High coast of the contest

Good time for submitters and organizers

It is worth organizing it.

Progress?

Have we measured some progress in the contest?

Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Progress?

Have we measured some progress in the contest?
Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Progress?

Have we measured some progress in the contest?
Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Progress?

Have we measured some progress in the contest?
Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Progress?

Have we measured some progress in the contest?
Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Progress?

Have we measured some progress in the contest?
Let's postpone this deep answer to the end

	2002	2003	2004	2005
#Solvers	28	34	55	43
#Submit.	19	22	27	22
Time 1st	2,400†	900	600	1,200
Time 2nd	21,600†	7,200	2,400	6,000/12,000
Total	750	522	516	1,328
Judges		J. Franco, H. van Maaren, T. Walsh	F. Bacchus, J. Marques-Silva, H. Kleine Buning	A. van Gelder, A. Biere, O. Kullman

Table: SAT Competition Overall Progress

Once the winner is known...

Let's analyse the results

Nobody knows what a good solver is. Winning does not mean that the solver is the best. A lot of other analysis are possible...

Once the winner is known...

Let's analyse the results

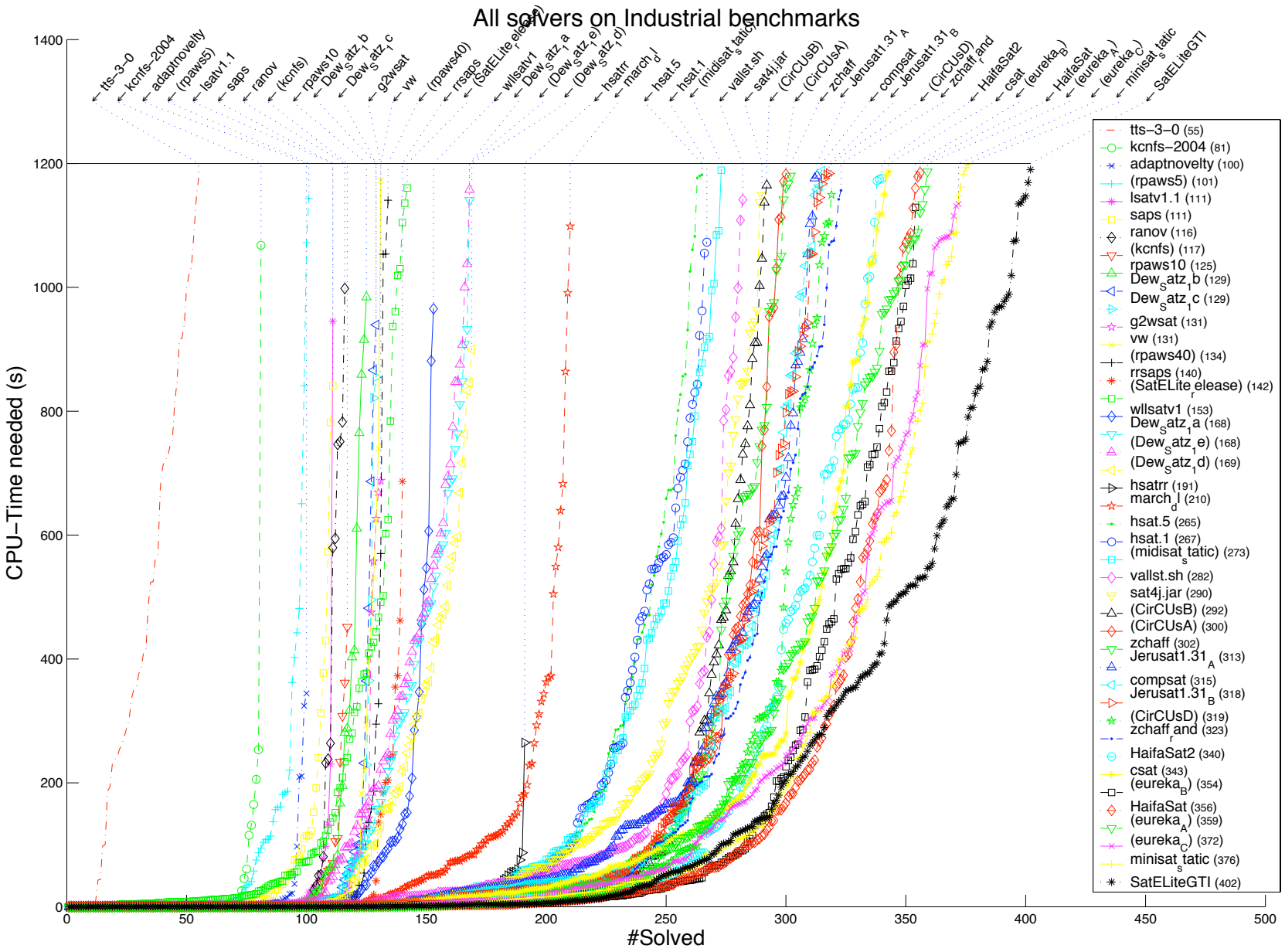
Nobody knows what a good solver is. Winning does not mean that the solver is the best. A lot of other analysis are possible...

Once the winner is known...

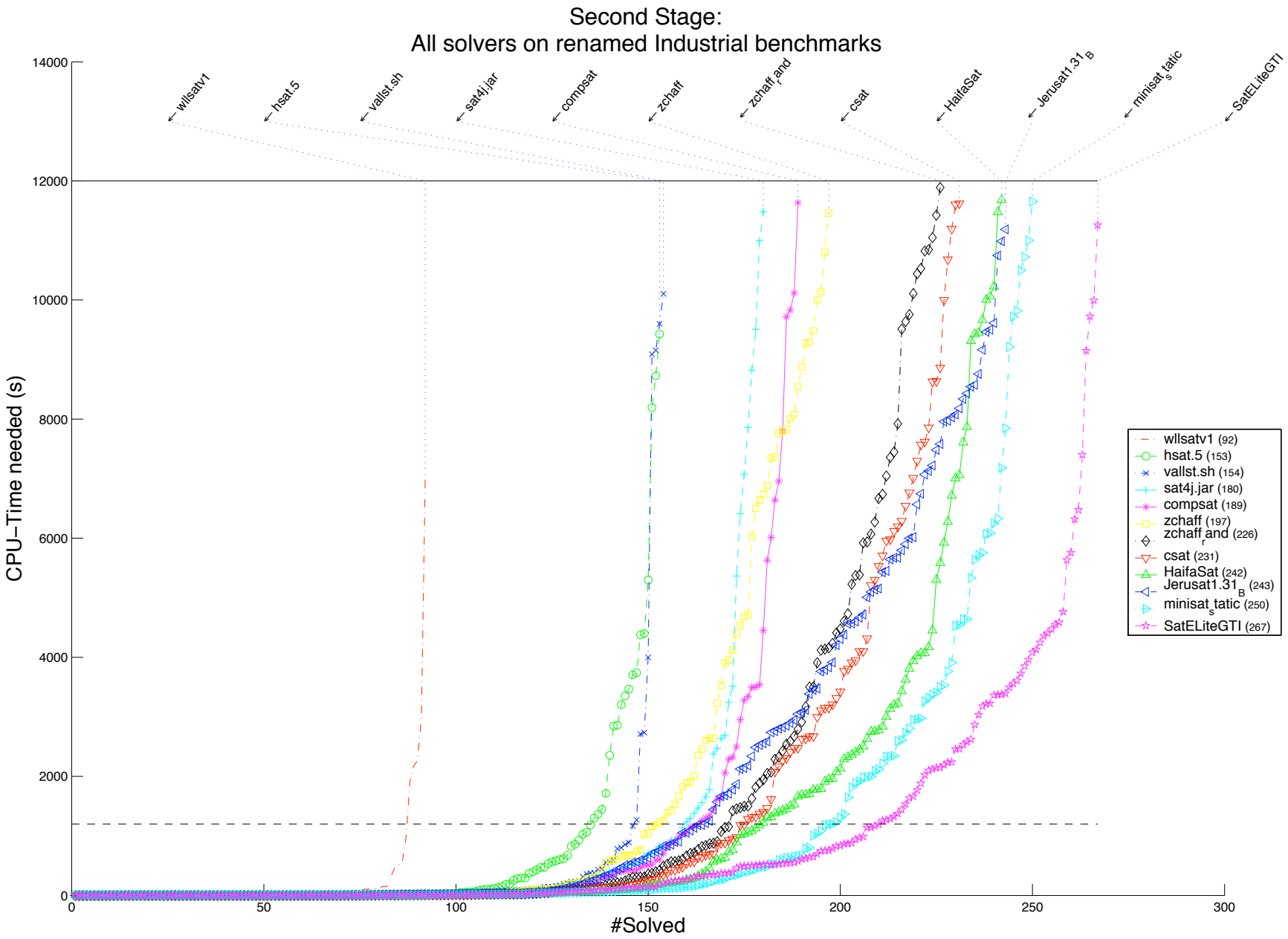
Let's analyse the results

Nobody knows what a good solver is. Winning does not mean that the solver is the best. A lot of other analysis are possible...

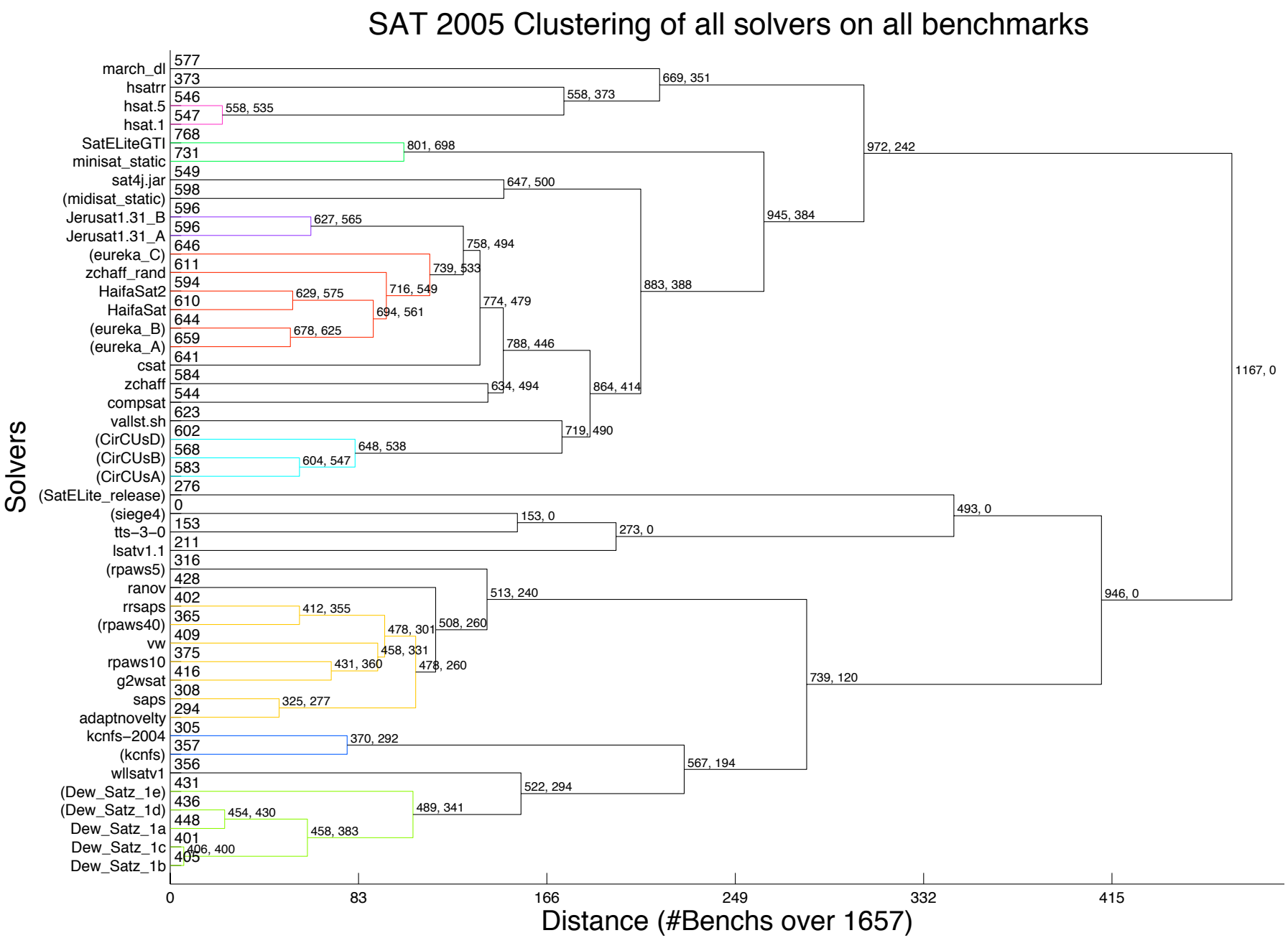
Solvers on SAT'05 industrial benchmarks



Solvers on SAT'05 industrial benchmarks



SAT'05 Solver Clustering



SOTA-Contributors

The State-Of-The-Art Solvers

Solver	Total	Random	Ind.	Crafted
circush0	6	–	–	6 (0/6)
kcnfs	6	5 (0/5)	–	1 (1/0)
jerusat1.3	5	–	4 (2/2)	1 (1/0)
brchaff	4	–	3 (0/3)	1 (1/0)
zchaff	4	–	4 (4/0)	–
quantor	3	–	3 (0/3)	–
satzoo-1.02	3	–	1 (0/1)	2 (0/2)
adaptnovelty	2	–	–	2 (2/0)
circush1	2	–	2 (2/0) –	
compsat	2	–	2 (1/1)	–
rsaps	2	1 (1/0)	–	1 (1/0)
zchaff-rand	2	–	2 (0/2)	–
cquest	1	–	1 (0/1)	–
march-eq-100	1	–	–	1 (1/0)
novelty35	1	–	1 (1/0)	–
novelty50	1	1 (1/0)	–	–
sapsnr	1	1 (1/0)	–	–
satzilla	1	–	1 (1/0)	–

Table: SAT'04 SOTAC ranking

Ranking *à la* SatEx

Solver Name	Time (hours)	#Solved (#SAT/#UNSAT)
`satelitegti`	**125**	**267** (**180**/**87**)
`minisat-static`	*+59*	*-17* (*-14*/*-3*)
`haifasat`	*+93*	*-25* (*-29*/*4*)
`jerusat1.31-b`	*+129*	*-24* (*-17*/*-7*)
`csat`	*+133*	*-36* (*-40*/*4*)
`zchaff-rand`	*+154*	*-41* (*-48*/*7*)
`compsat`	*+230*	*-78* (*-66*/*-12*)
`zchaff`	*+236*	*-70* (*-59*/*-11*)
`sat4j.jar`	*+253*	*-87* (*-70*/*-17*)
`hsat.5`	*+336*	*-114* (*-81*/*-33*)
`wllsatv1`	*+521*	*-175* (*-94*/*-81*)

Table: SAT'05 results on industrials, non shuffled, second stage only.

Relative efficiency of solvers

How to deal with timeouts? [Le Berre and Simon, 2003]

Let $s(X)$ be the set of benchmarks solved by X.

Relative Efficiency (re)

$$re(X,Y) = \frac{|s(X) \cap s(Y)|}{|s(Y)|}$$

CPU-time reduction ratio (crr)

Let $CPU(X,b)$ be the time for X to solve all the benchmarks in b (defined only if $b \subseteq s(X)$)

$$crr(X,Y) = \frac{CPU(X, s(X) \cap s(Y))}{CPU(Y, s(X) \cap s(Y))}$$

$re(X)$ and $crr(X)$ are the averaged values over all Y

Relative efficiency of solvers

How to deal with timeouts? [Le Berre and Simon, 2003]

Let $s(X)$ be the set of benchmarks solved by X.

Relative Efficiency (re)

$$re(X,Y) = \frac{|s(X) \cap s(Y)|}{|s(Y)|}$$

CPU-time reduction ratio (crr)

Let $CPU(X,b)$ be the time for X to solve all the benchmarks in b (defined only if $b \subseteq s(X)$)

$$crr(X,Y) = \frac{CPU(X, s(X) \cap s(Y))}{CPU(Y, s(X) \cap s(Y))}$$

$re(X)$ and $crr(X)$ are the averaged values over all Y

Relative efficiency of solvers

How to deal with timeouts? [Le Berre and Simon, 2003]

Let $s(X)$ be the set of benchmarks solved by X.

Relative Efficiency (re)

$$re(X,Y) = \frac{|s(X) \cap s(Y)|}{|s(Y)|}$$

CPU-time reduction ratio (crr)

Let $CPU(X,b)$ be the time for X to solve all the benchmarks in b (defined only if $b \subseteq s(X)$)

$$crr(X,Y) = \frac{CPU(X, s(X) \cap s(Y))}{CPU(Y, s(X) \cap s(Y))}$$

$re(X)$ and $crr(X)$ are the averaged values over all Y

Relative efficiency of solvers

Examples on last year Contest [Le Berre and Simon, 2004]

Random
12 Solvers

Solver	re	crr
adaptnovelty	0.95	0.37
walksatrnp	0.95	0.49
sapsnr	0.95	0.56
saps	0.95	0.67
novelty35	0.93	0.86
qingting	0.81	2.81
satzilla-nr	0.80	0.93
unitwalk	0.79	2.49
gasat	0.78	1.82
cls	0.50	10.81
kcnfs	0.43	280.54
march-007	0.33	284.76

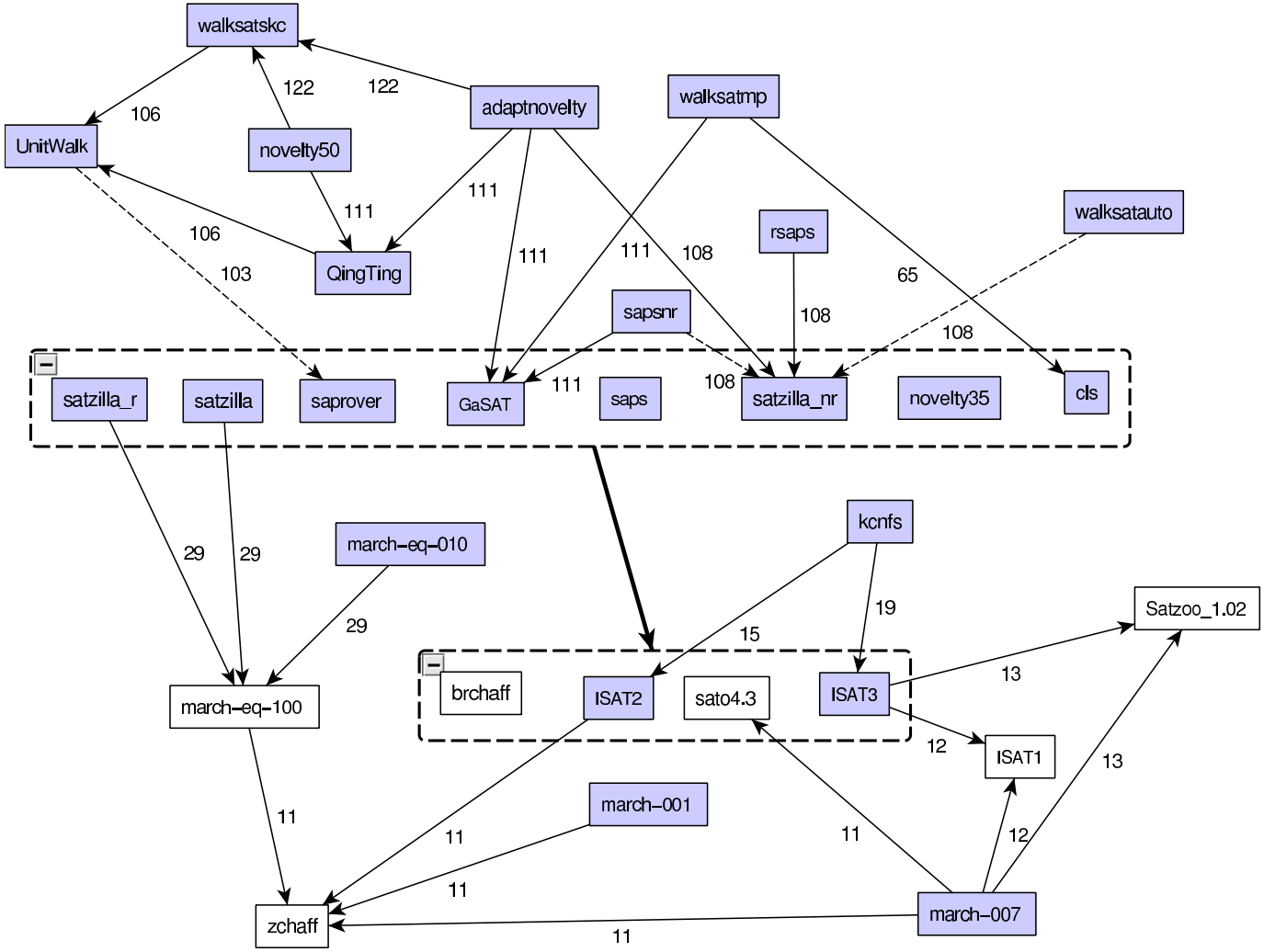
Industrial
11 Solvers

Solver	re	crr
oepirc	0.86	0.74
forklift	0.77	0.68
zchaff-rand	0.73	0.76
brchaff	0.66	0.89
compsat	0.64	0.70
satzoo-1.02	0.64	1.94
jerusat1.3	0.63	1.55
circush1	0.62	1.05
cquest	0.61	1.17
quantor	0.58	1.43
minilearning.jar	0.56	1.48

Table: *crr* and *re* values for the second stage of the SAT'04 Contest

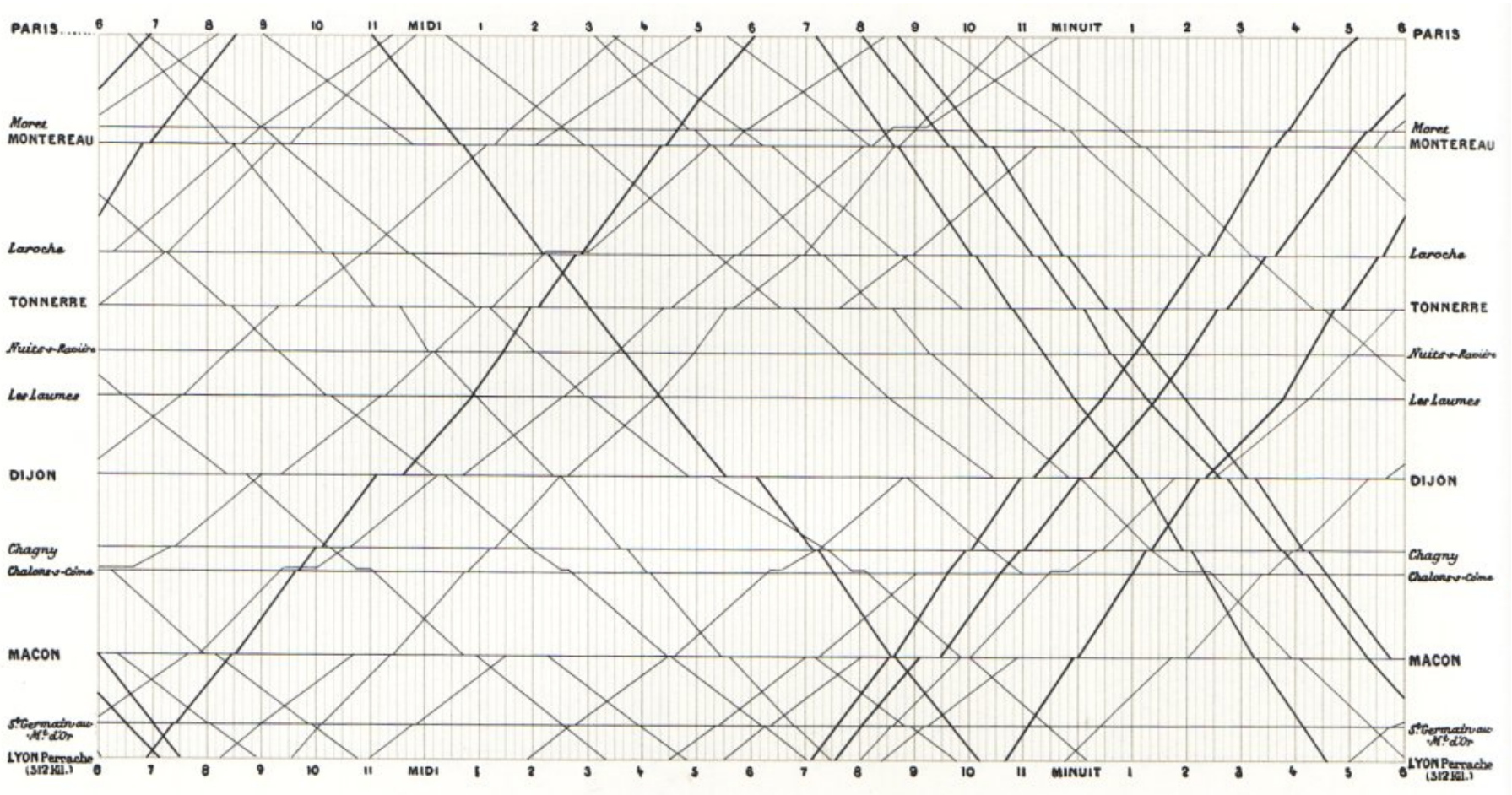
Subsumptions between solvers

Random problems, SAT'O4 results [Le Berre and Simon, 2004]

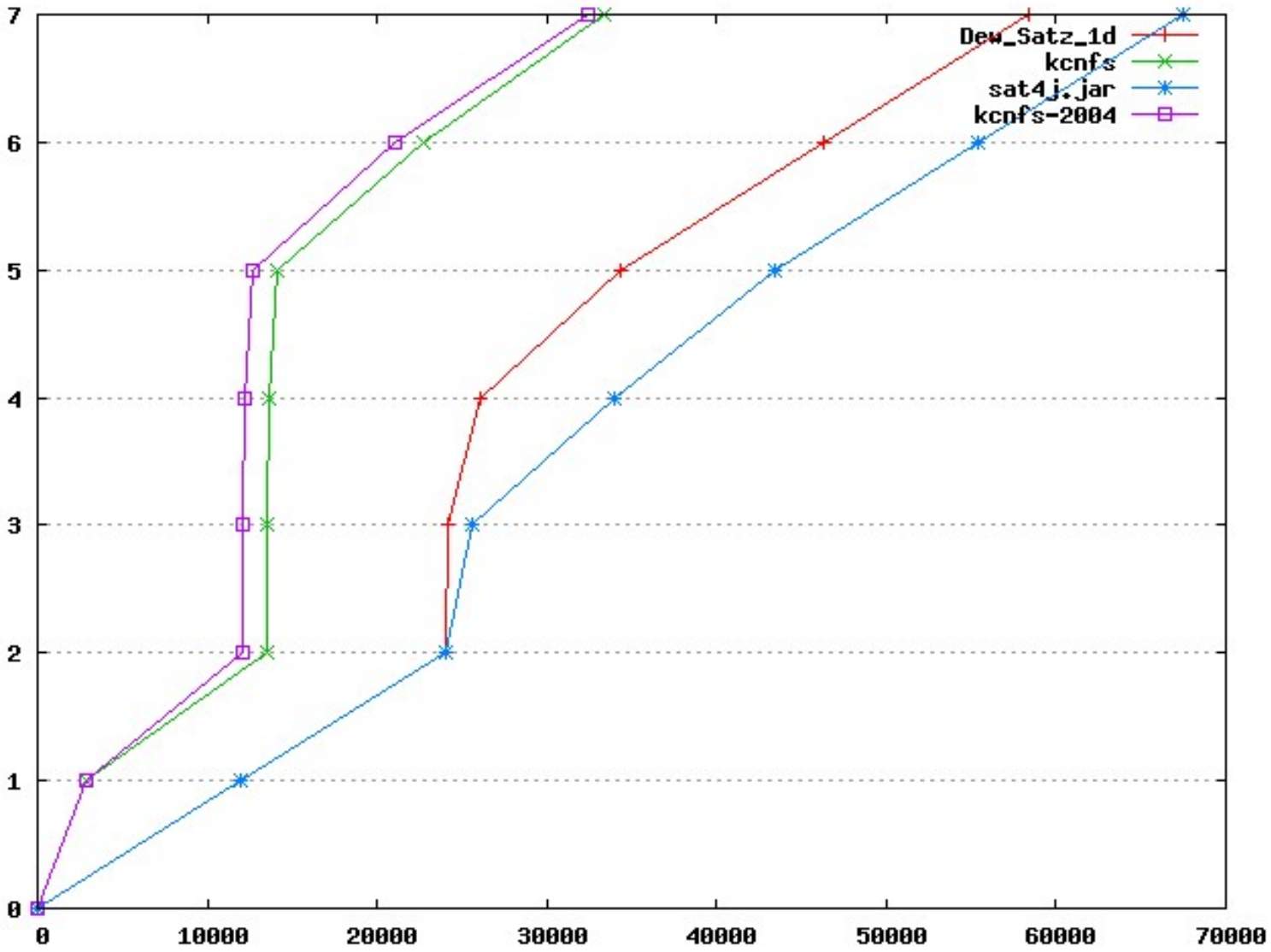


Take the train

At least its time-scheduling graphical representation



Take the SAT train



Let everybody make its own analysis

Results may be exploited by anyone

All results are available for the community in an HTML and a raw format.

For instance, Emmanuel Zarpas (IBM) writes papers on ways of analysis results of the second stage, indutrial benchmarks, based on the contest results.

Actual (and future) tendency

Different point of view in the community

- Submitters analyse the results in their solver papers
- Specialists publish analysis
  - Phylogenetic tree of solvers [Purdom et al., 2005]
  - Results on Random instances (O. Kullman)
  - Results on Industrial instances (E. Zarpas)

Let everybody make its own analysis

Results may be exploited by anyone

All results are available for the community in an HTML and a raw format.

For instance, Emmanuel Zarpas (IBM) writes papers on ways of analysis results of the second stage, indutrial benchmarks, based on the contest results.

Actual (and future) tendency

Different point of view in the community

- Submitters analyse the results in their solver papers
- Specialists publish analysis
  - Phylogenetic tree of solvers [Purdom et al., 2005]
  - Results on Random instances (O. Kullman)
  - Results on Industrial instances (E. Zarpas)

SAT'05 Contest Roadmap

1. Competition Historics
   - SAT Contest over the years
2. Rules... and errors
   - The SAT'02-04 contest
3. SAT'05 Contest Results
   - A new scoring scheme: SAT'05 contest
   - Ways of presenting results
4. **Lessons**
   - Lessons
   - Conclusion
   - Bibliography

First Lesson: let's take a break

Progresses are clear
but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

First Lesson: let's take a break

Progresses are clear

but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

First Lesson: let's take a break

Progresses are clear
but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

First Lesson: let's take a break

Progresses are clear
but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

First Lesson: let's take a break

Progresses are clear
but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

First Lesson: let's take a break

Progresses are clear
but now relatively slow and...

Doing a competition once every two years

- May preserve the competition as a *big event*
- Take some distance in the rush for glory
- Let some time to analyse resuts, learn, and test new methods
- Do not exhaust submitters (and organizers)

Dropped ideas

- Tournament between solvers
- Auto-Classification of benchmarks
  (based on experimental clustering of solvers)
- Global Time for solver, up to them to solve the maximum number of solvers
- ... (each submitter have its own opinion) ...

Dropped ideas

- Tournament between solvers
- Auto-Classification of benchmarks
  (based on experimental clustering of solvers)
- Global Time for solver, up to them to solve the maximum number of solvers
- ... (each submitter have its own opinion) ...

Have we measured a progress?

Industrial benchmarks and solvers

- zchaff-like and minisat-like solvers are dominating
- Most solvers may now work on very large benchmarks
- At a time, it was hard to test new ideas. No more.

Random benchmarks and solvers

- Uncomplete local search are the clear winner on SAT
- Backbone-heuristic search is the winner on UNSAT

Crafted benchmarks and solvers

`lsat` solved all tricky formulae. They disapeared from the contest. The crafted category is now the field of fast solvers.

Solvers now scale well. In the SAT'02 contest, only a very few unsolved benchmarks in 40 minutes were solved in 6 hours.

Have we measured a progress?

Industrial benchmarks and solvers

- zchaff-like and minisat-like solvers are dominating
- Most solvers may now work on very large benchmarks
- At a time, it was hard to test new ideas. No more.

Random benchmarks and solvers

- Uncomplete local search are the clear winner on SAT
- Backbone-heuristic search is the winner on UNSAT

Crafted benchmarks and solvers

`lsat` solved all tricky formulae. They disapeared from the contest. The crafted category is now the field of fast solvers.

Solvers now scale well. In the SAT'02 contest, only a very few unsolved benchmarks in 40 minutes were solved in 6 hours.

Have we measured a progress?

Industrial benchmarks and solvers

- zchaff-like and minisat-like solvers are dominating
- Most solvers may now work on very large benchmarks
- At a time, it was hard to test new ideas. No more.

Random benchmarks and solvers

- Uncomplete local search are the clear winner on SAT
- Backbone-heuristic search is the winner on UNSAT

Crafted benchmarks and solvers

`lsat` solved all tricky formulae. They disapeared from the contest. The crafted category is now the field of fast solvers.

Solvers now scale well. In the SAT'02 contest, only a very few unsolved benchmarks in 40 minutes were solved in 6 hours.

Have we measured a progress?

Industrial benchmarks and solvers

- zchaff-like and minisat-like solvers are dominating
- Most solvers may now work on very large benchmarks
- At a time, it was hard to test new ideas. No more.

Random benchmarks and solvers

- Uncomplete local search are the clear winner on SAT
- Backbone-heuristic search is the winner on UNSAT

Crafted benchmarks and solvers

`lsat` solved all tricky formulae. They disapeared from the contest. The crafted category is now the field of fast solvers.

Solvers now scale well. In the SAT'02 contest, only a very few unsolved benchmarks in 40 minutes were solved in 6 hours.

Lesson: Good things

Scoring scheme helps new original methods to emerge

The judges

- Solvers that enter the second stage
- Winners
- Promizing methods (if any)
- How to deal with special bugs

Associated Events and Special Tracks

- File format for checking UNSAT proof exists
- Other associated competition events (PB, QBF)

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Lesson: Let's face the challenges

Reasonable challenges

- How to correctly scramble benchmarks?
- Strong solver/solver comparison by multiple launchs? (instead of solver X is better than Y, we'll have solver X is better than Y, with 90% confidence)
- Handle UNSAT proofs, and SP-like solvers

Real challenges

- How to evaluate massively distributed solvers
- Experimentally understand solvers behavior

And...

- Introduction of preprocessing methods
- Build bridge between contests

Our Graal?

A real experimental science of algorithms?

Bibliography I

Brglez, F., Stallmann, M., and Li, X. (2003).
Satbed: A configurable environment for reliable performance experiments with sat instance classes and algorithms.
SAT'03.

Davis, M., Logemann, G., and Loveland, D. (1962).
A machine program for theorem proving.
Communications of the ACM, 5(7):394–397.

Davis, M. and Putnam, H. (1960).
A computing procedure for quantification theory.
Journal of the ACM, 7(3):201–215.

Dubois, O. and Dequen, G. (2001).
A backbone-search heuristic for efficient solving of hard 3-sat formulae.
In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle, Washington, USA.

Le Berre, D. and Simon, L. (2003).
Essentials of the sat 03 competition.
SAT'03.

Le Berre, D. and Simon, L. (2004).
Fifty-five solvers in vancouver.
SAT'04.

Bibliography II

Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S. (2001).
Chaff: Engineering an efficient SAT solver.
In *Proceedings of the 38th Design Automation Conference (DAC'01)*, pages 530–535.

Purdom, P., Berre, D. L., and Simon, L. (2005).
A phylogenetic tree for the sat 2002 contest.
Accepted for publication in Annals of Mathematics and Artificial Intelligence (AMAI), 43.

van Gelder, A. (2005).
To appear.
JSAT.