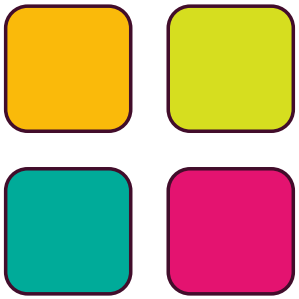
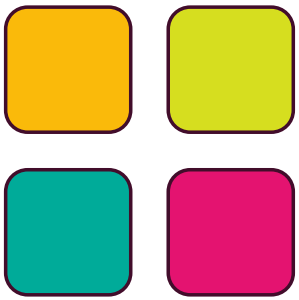


Improving Glucose for Incremental SAT Solving with Assumptions: Application to MUS Extraction

Gilles Audemard
Jean-Marie Lagniez and Laurent Simon

SAT 2013



Introduction and Motivations

Minimum Unsatisfiable Subformula

$x \vee y \vee z$

$\neg x \vee y \vee z$

$\neg x \vee \neg y$

$x \vee \neg y$

$x \vee w$

$\neg x \vee \neg z$

UNSAT

$x \vee \neg z$

$w \vee z \vee \neg y$

$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses

Minimum Unsatisfiable Subformula

$x \vee y \vee z$	$x \vee \neg y$	$x \vee \neg z$
$\neg x \vee y \vee z$	$x \vee w$	$w \vee z \vee \neg y$
$\neg x \vee \neg y$	$\neg x \vee \neg z$	$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses

Minimum Unsatisfiable Subformula

$x \vee y \vee z$ $x \vee \neg y$ $x \vee \neg z$

$\neg x \vee y \vee z$ $x \vee w$ $w \vee z \vee \neg y$

$\neg x \vee \neg y$ $\neg x \vee \neg z$ $w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$\neg x \vee y \vee z$

$\neg x \vee \neg y$

$x \vee \neg y$

$x \vee w$

$\neg x \vee \neg z$

SAT

$x \vee \neg z$

$w \vee z \vee \neg y$

$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$x \vee y \vee z$	$x \vee \neg y$	$x \vee \neg z$
$\neg x \vee y \vee z$	$x \vee w$	$w \vee z \vee \neg y$
$\neg x \vee \neg y$	$\neg x \vee \neg z$	$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$x \vee y \vee z$	$x \vee \neg y$	$x \vee \neg z$
$\neg x \vee y \vee z$	$x \vee w$	$w \vee z \vee \neg y$
$\neg x \vee \neg y$	$\neg x \vee \neg z$	$w \vee \neg x \vee \neg z$

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$x \vee y \vee z$	$x \vee \neg y$	$x \vee \neg z$
$\neg x \vee y \vee z$		$w \vee z \vee \neg y$
$\neg x \vee \neg y$	$\neg x \vee \neg z$	$w \vee \neg x \vee \neg z$
	UNSAT	

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$x \vee y \vee z$	$x \vee \neg y$	$x \vee \neg z$
$\neg x \vee y \vee z$		
$\neg x \vee \neg y$	$\neg x \vee \neg z$	$w \vee \neg x \vee \neg z$
	UNSAT	

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Minimum Unsatisfiable Subformula

$x \vee y \vee z$ $x \vee \neg y$ $x \vee \neg z$

$\neg x \vee y \vee z$

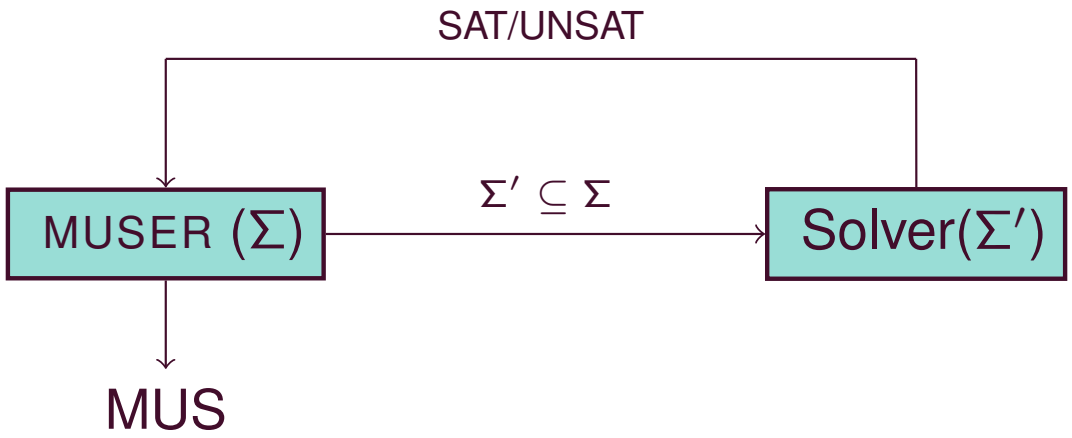
$\neg x \vee \neg y$ $\neg x \vee \neg z$

MUS !

- The formula is inconsistant : Why ?
- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov etal, AI Com 2012]. The tool MUSER

Muser Architecture

Incremental SAT



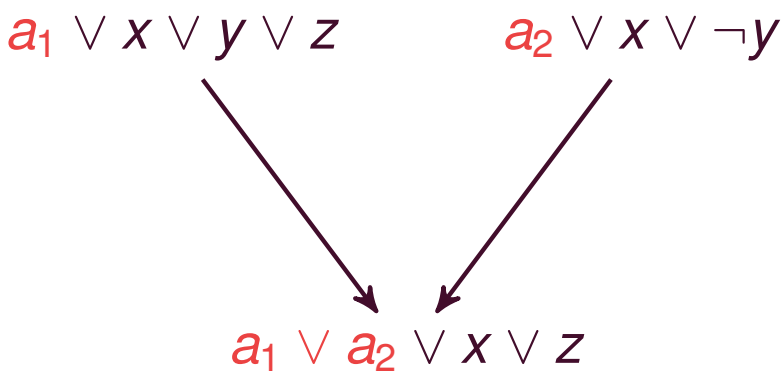
- Successive calls to a SAT oracle
- Non independant calls
- Informations between two calls are preserved
  - Heuristics : VSIDS, phase saving, restarts...
  - **Learnt clauses**

Forget some clauses and some learnt clauses

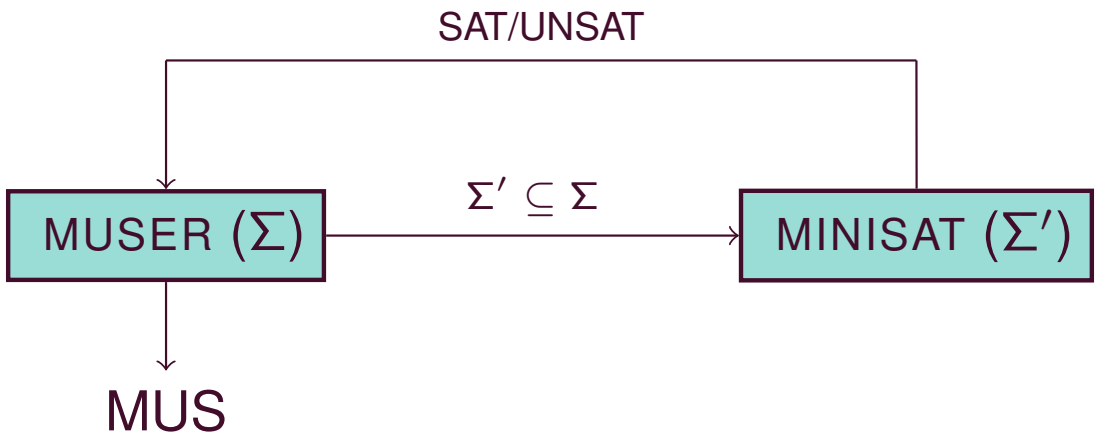
- Add one selector (fresh variable) a_i per clause

$a_1 \vee x \vee y \vee z$ $a_2 \vee x \vee \neg y$ $a_3 \vee x \vee \neg z$
$a_4 \vee \neg x \vee y \vee z$ $a_5 \vee x \vee w$ $a_6 \vee w \vee z \vee \neg y$
$a_7 \vee \neg x \vee \neg y$ $a_8 \vee \neg x \vee \neg z$ $a_9 \vee w \vee \neg x \vee \neg z$

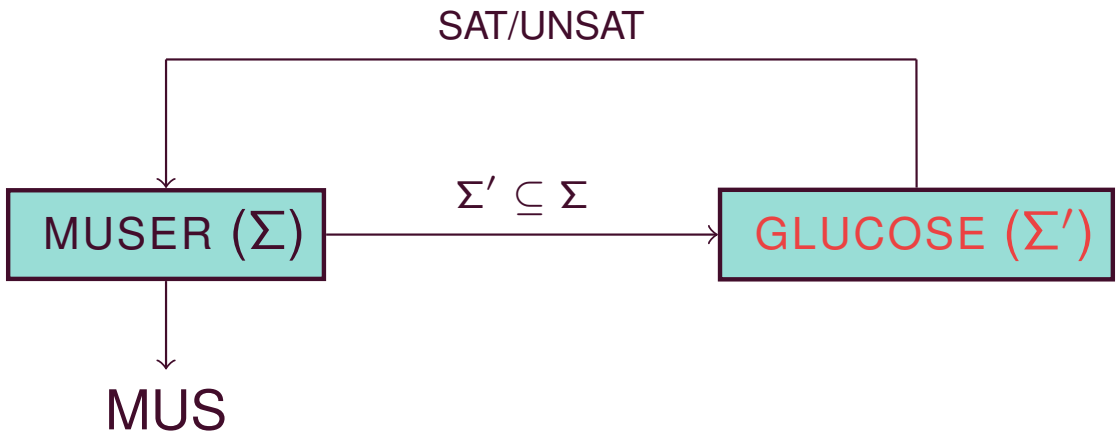
- Assign a_i (as an assumption) to false to activate the clause i
- Assign a_i (as an assumption) to true to disable the clause i
- All learnt clauses related to a disable clause will be disabled !



Our work

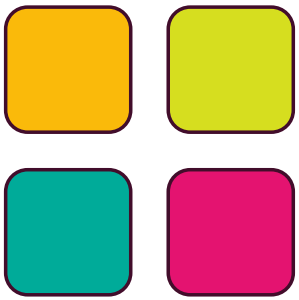


Our work



- Plug GLUCOSE in MUSER
- Adapt and modify GLUCOSE to improve MUSER performances

Improve SAT oracle in order to improve the MUSER tool

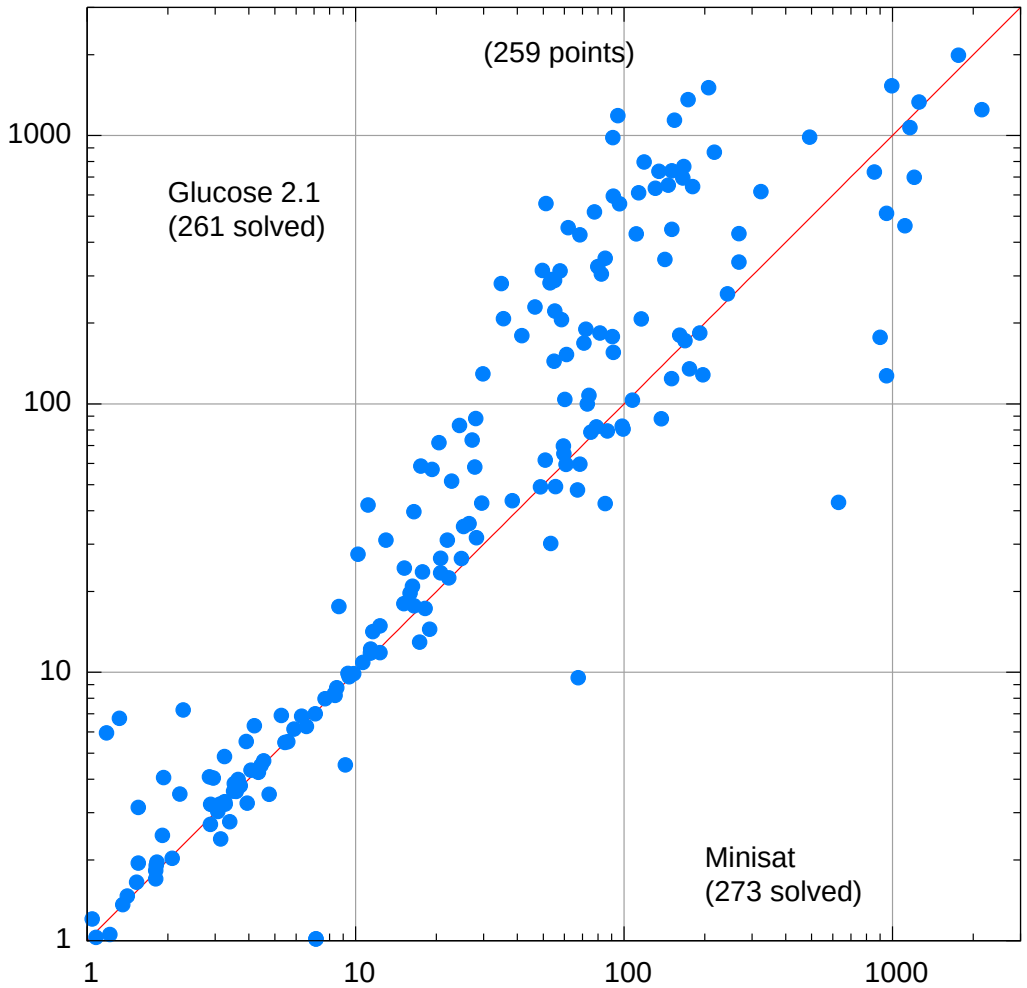


GLUCOSE and MUSER

Test set

- 300 instances from the SAT competition 2011, MUS category
- timeout set to 2400 seconds
- MUSER is used with default options (destructive approach, model rotation)

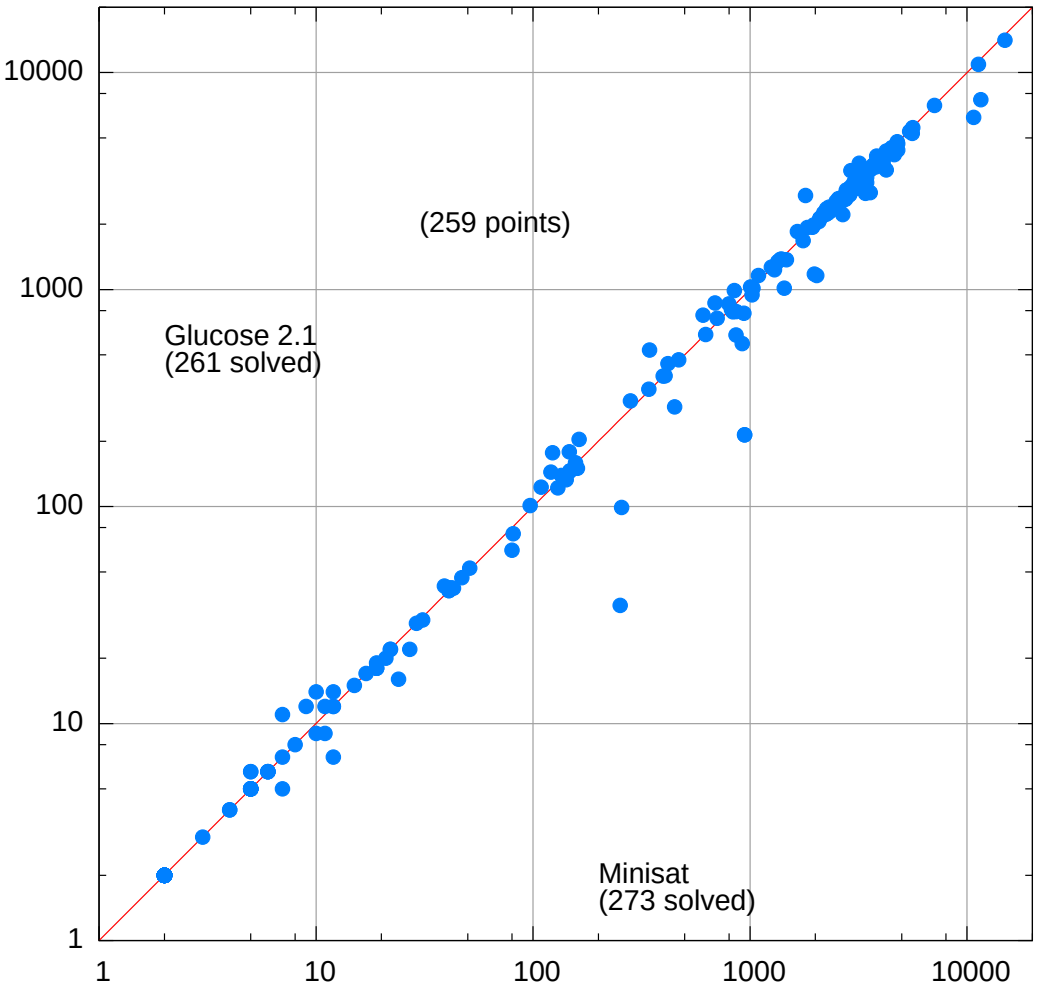
A first Attempt



Disappointing results

Trying to explain these bad results

Disappointing results



Disappointing results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls (not shown in the paper)
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

Disappointing results

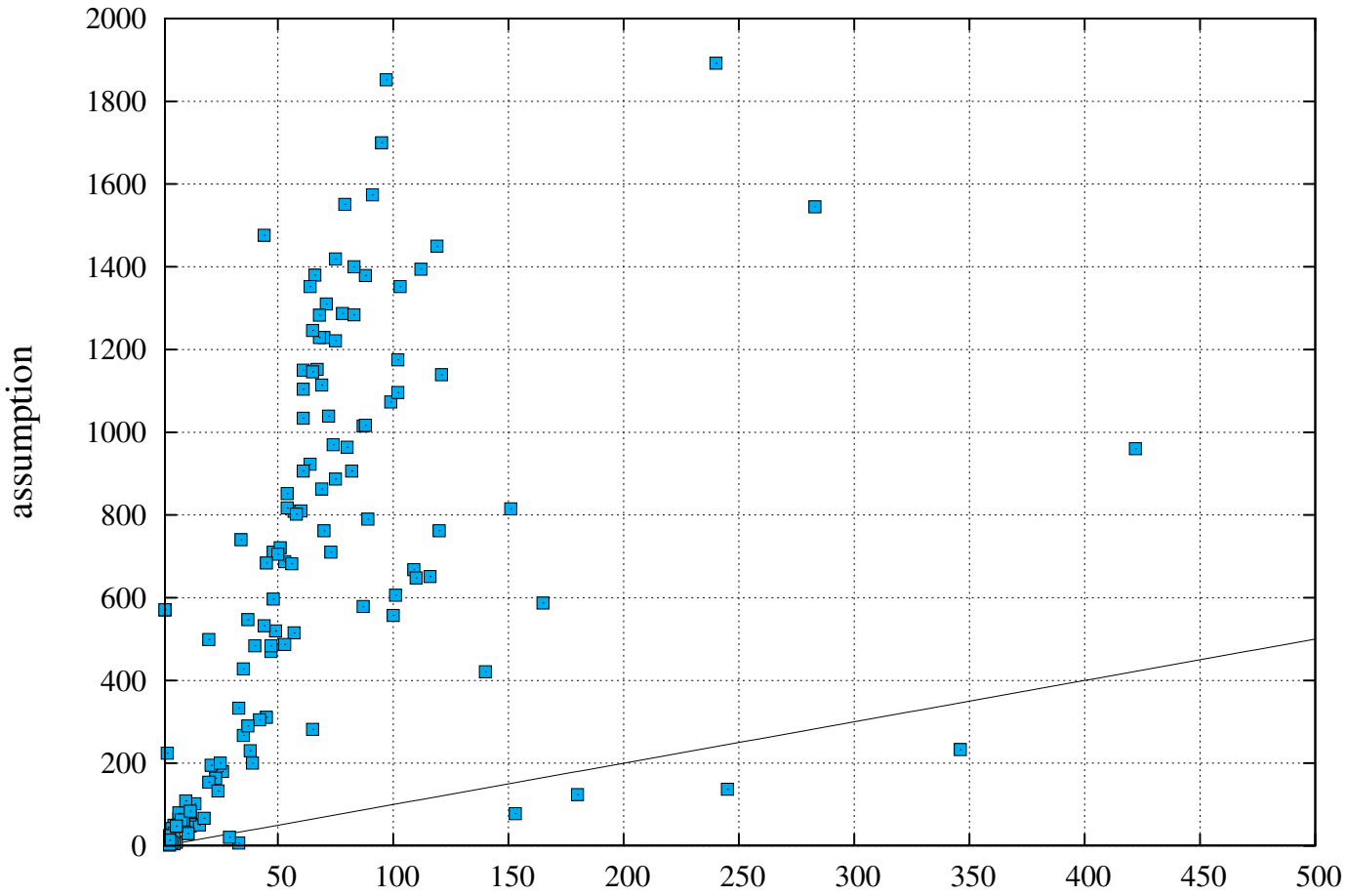
Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls (not shown in the paper)
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level

Disappointing results

- Each point represents an instance
- x-axis is the average number of initial variables in learnt clauses
- y-axis is the average number of selector variables in learnt clauses



Disappointing results

		LBD				
			size		LBD	
Instance	#C	time	avg	max	avg	max
fdmus_b21_96	8541	29	1145	5980	1095	5945
longmult6	8853	46	694	3104	672	3013
dump_vc950	360419	110	522	36309	498	35873
g7n	70492	190	1098	16338	1049	16268

- LBD looks like size
- Clauses are very long

Disappointing results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls (not shown in the paper)
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- **The LBD of a clause looks like its size !**

Disappointing results

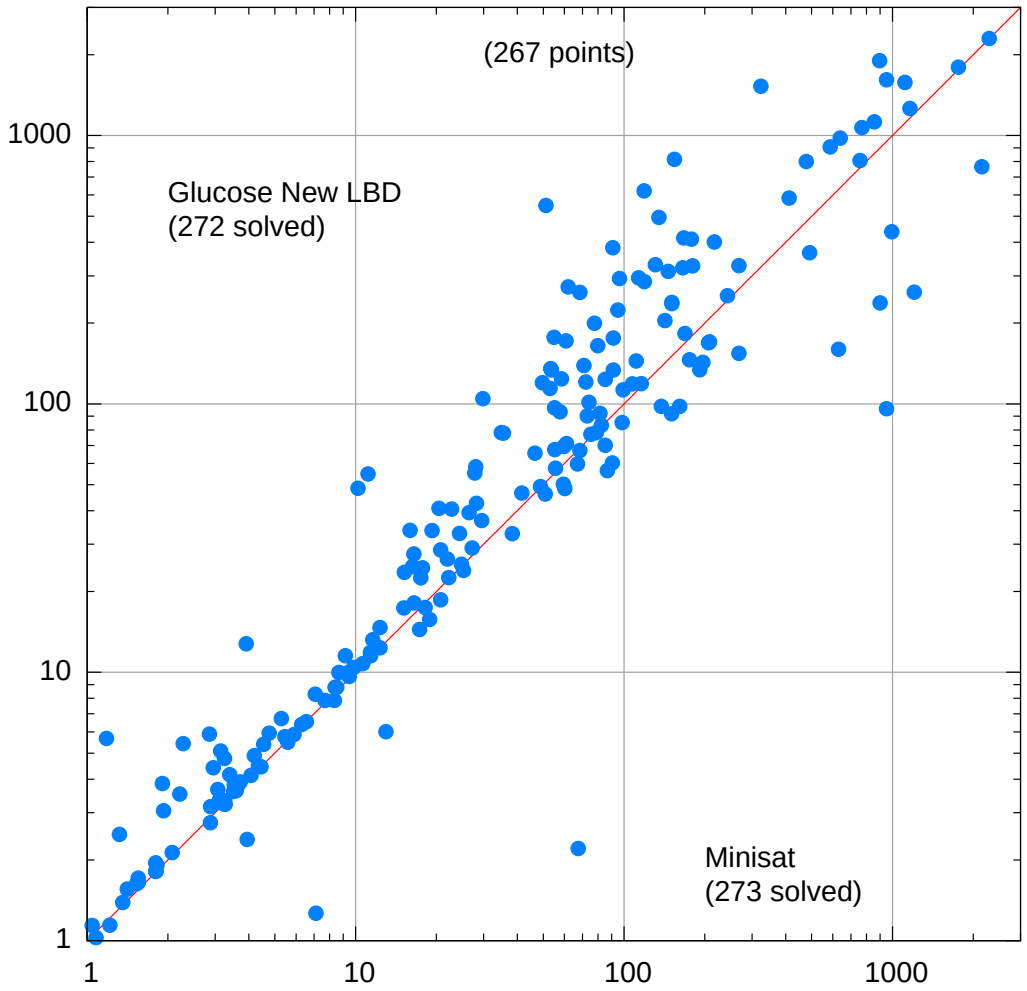
Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls (not shown in the paper)
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- **The LBD of a clause looks like its size !**

Refine LBD : Do not take into account selectors

A second attempt



New LBD

		LBD					New LBD				
			size		LBD			size		LBD	
Instance	#C	time	avg	max	avg	max	time	avg	max	avg	max
fdmus_b21_96	8541	29	1145	5980	1095	5945	**11**	972	6391	**8**	71
longmult6	8853	46	694	3104	672	3013	**14**	627	2997	**11**	61
dump_vc950	360419	110	522	36309	498	35873	**67**	1048	36491	**8**	307
g7n	70492	190	1098	16338	1049	16268	**75**	1729	17840	**27**	160

LBD matters

- However, results need to be improve

Clauses are too long

Many algorithms have to traverse clauses

- Dynamic computing of LBD (useful but costly)
- Conflict analysis
- Unit propagation
- Deleting satisfiable clauses

Clauses are too long

Many algorithms have to traverse clauses

- Dynamic computing of LBD (useful but costly)
  - → Store the number of selectors in the clause
  - → Stop when all initial literals have been tested
- Conflict analysis
- Unit propagation
- Deleting satisfiable clauses

Clauses are too long

Many algorithms have to traverse clauses

- Dynamic computing of LBD (useful but costly)
  - → Store the number of selectors in the clause
  - → Stop when all initial literals have been tested
- Conflict analysis
  - → Force initial literals to be placed at the beginning
- Unit propagation
- Deleting satisfiable clauses

Clauses are too long

Many algorithms have to traverse clauses

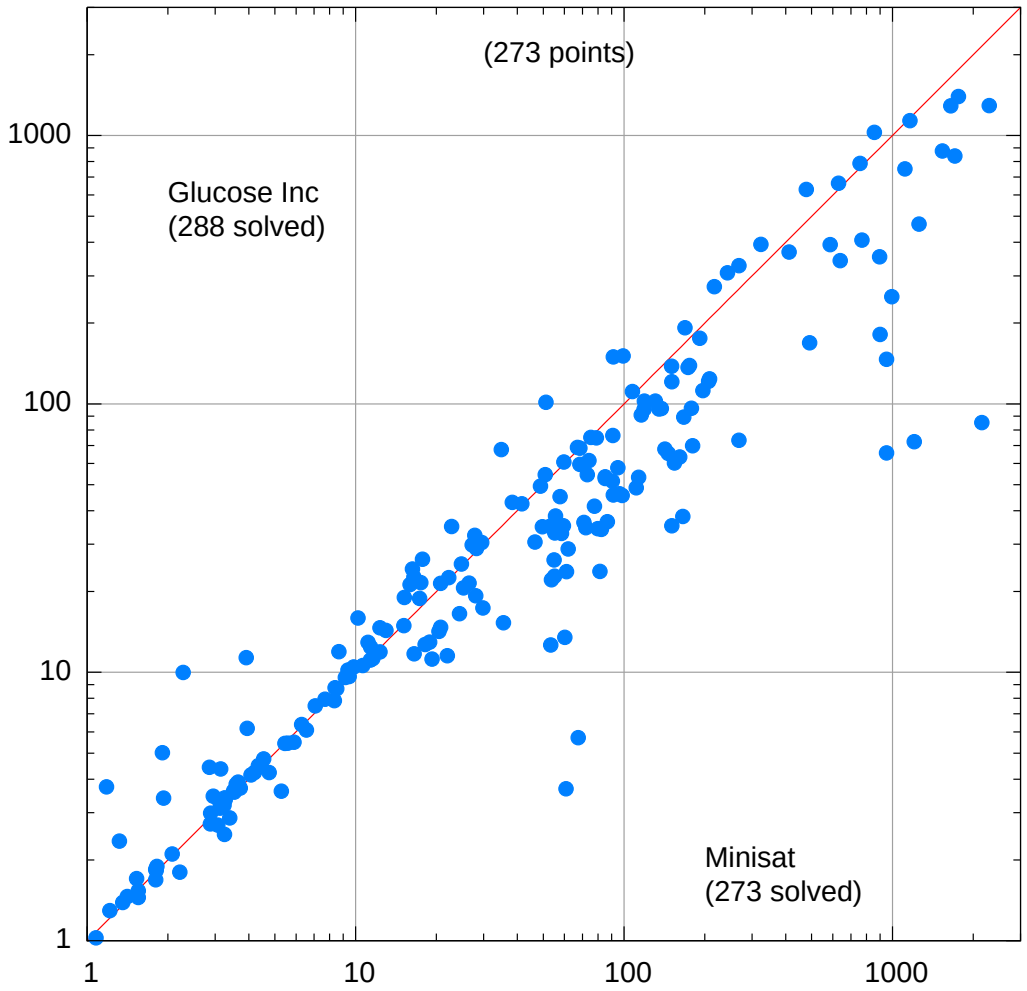
- Dynamic computing of LBD (useful but costly)
  - → Store the number of selectors in the clause
  - → Stop when all initial literals have been tested
- Conflict analysis
  - → Force initial literals to be placed at the beginning
- Unit propagation
  - → Look for a non selector literal or a satisfied one
  - → Push selectors at the end of the clause
- Deleting satisfiable clauses

Clauses are too long

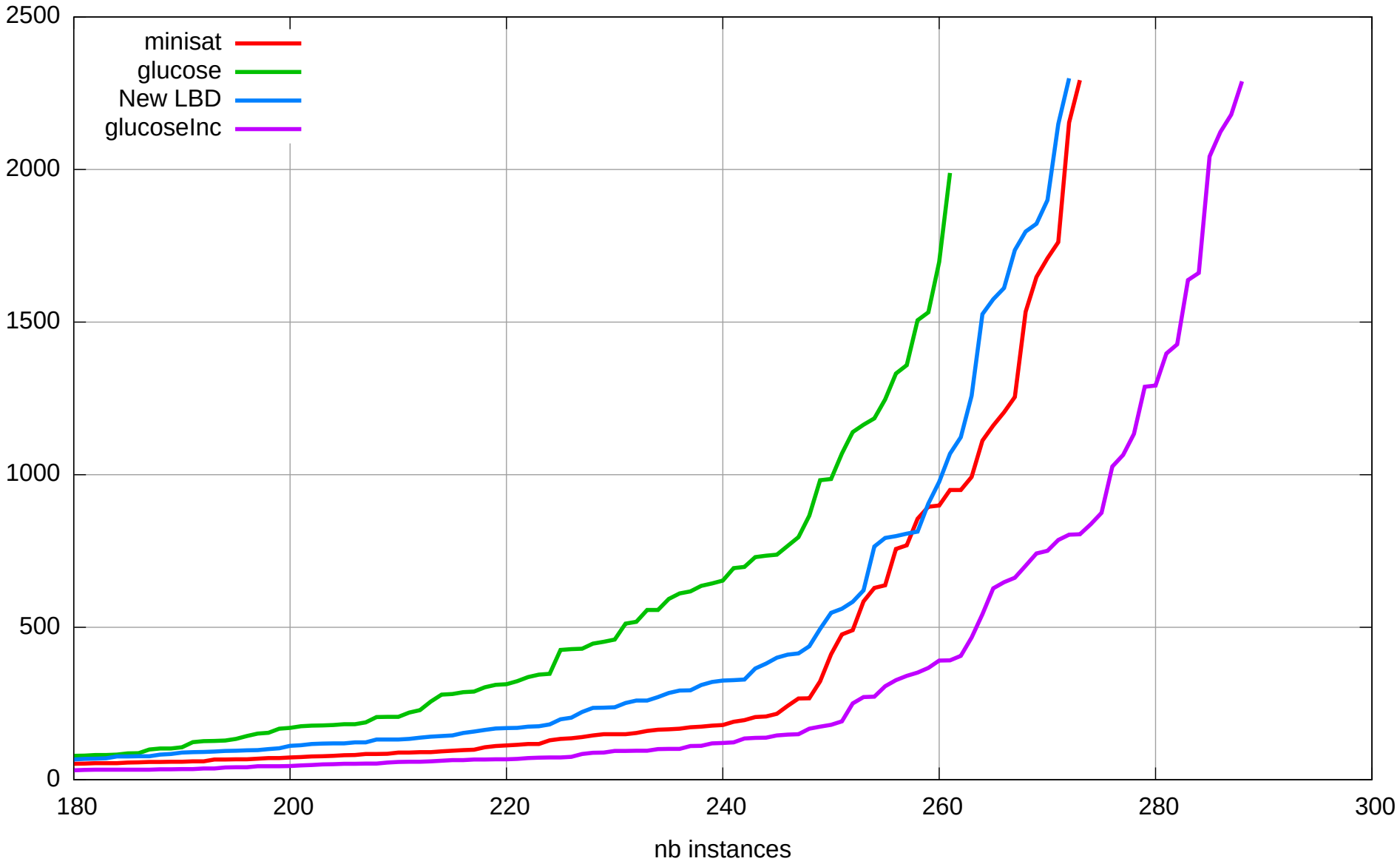
Many algorithms have to traverse clauses

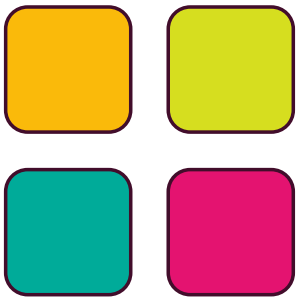
- Dynamic computing of LBD (useful but costly)
  - → Store the number of selectors in the clause
  - → Stop when all initial literals have been tested
- Conflict analysis
  - → Force initial literals to be placed at the beginning
- Unit propagation
  - → Look for a non selector literal or a satisfied one
  - → Push selectors at the end of the clause
- Deleting satisfiable clauses
  - → Take only watched literals into account

Third attempt



Final comparison





Conclusion

Conclusion

- Adapt GLUCOSE to deal with selectors and assumptions
  - Adapt the definition of LBD
  - Modify algorithms dealing with long clauses
- Application to MUS extraction (using MUSER)

- Modify heuristics to take into account the semantic of selectors
  - From a black box to a gray box SAT oracle
- Try other contexts : MAXSAT for example
- Suggested by Alexander Nadel : An incremental track in next competition...