Conflict-Based Selection of Branching Rules in SAT Algorithms

Marc Herbstritt

Bernd Becker

Institute of Computer Science
Albert-Ludwigs-Universität
Freiburg im Breisgau

Where do we come from?

- South-west Germany
- Freiburg
- Black Forest
- near Rhein river
- near France and Suisse

What is our interest in SAT?

- Research focus on VLSI topics
  - Testing
  - Verification
  - Logic Synthesis
  - Routing
- Basic data structures and algorithms:
  - Decision Diagrams (BDDs,OKFDDs,K*BMDs,...)
  - SAT (general SAT, structural SAT, ...)
  - ...

Overview

- Introduction
- SAT Applications
- SAT Algorithm
- Branching Rules
- Adaptive Framework
- Experimental Results
- Conclusions

SAT Algorithms: New Features

- Intelligent Branching Rules
- Preprocessing
- Conflict analysis techniques
  - Non-chronological Backtracking
  - Conflict Learning
- Restarts
- Algorithm Portfolio

Branching Rules: Comparison

Branching Rule	Time	Aborts
Böhm	1817,45	8
MOM	1428,04	7
OS-JW	807,82	4
TS-JW	911,28	4
DLCS	746,3	3
DLIS	409,14	1
RDLIS	439,16	1,1
RAND	1431,85	5,7

Conclusion: DLIS gets best results

Observation: But still instance specific differences

- ➡ no general best-of-all branching rule
- ➡ variable selection in DP is NP-/coNP-hard (Liberatore, 2000)

Conflict Analysis (4/4)

Assume (v_{11}=1) @ DL 7:

→ v_{12}=0 due to c_1

→ v_{16}=1 due to c_2

→ v_2=0 due to c_3

→ v_{10}=0 due to c_4

→ $\mathbf{v_1}$=1 due to c_5

→ $\mathbf{v_3}$=1 due to c_6

→ $\mathbf{v_5}$=0 due to c_7

→ $\mathbf{v_{18}}$=1 due to c_8

→ conflict at c_9 due to v_{18}

- 1UIP scheme stops at R_4
- v_{10} last literal from DL 7 in R_4
- next „lower" in R_4: v_{19}=0 @ DL 3
- R_4 triggers v_{10}=1 @ DL 3
- Nonchronological backtracking to DL 3

Resolution ↑

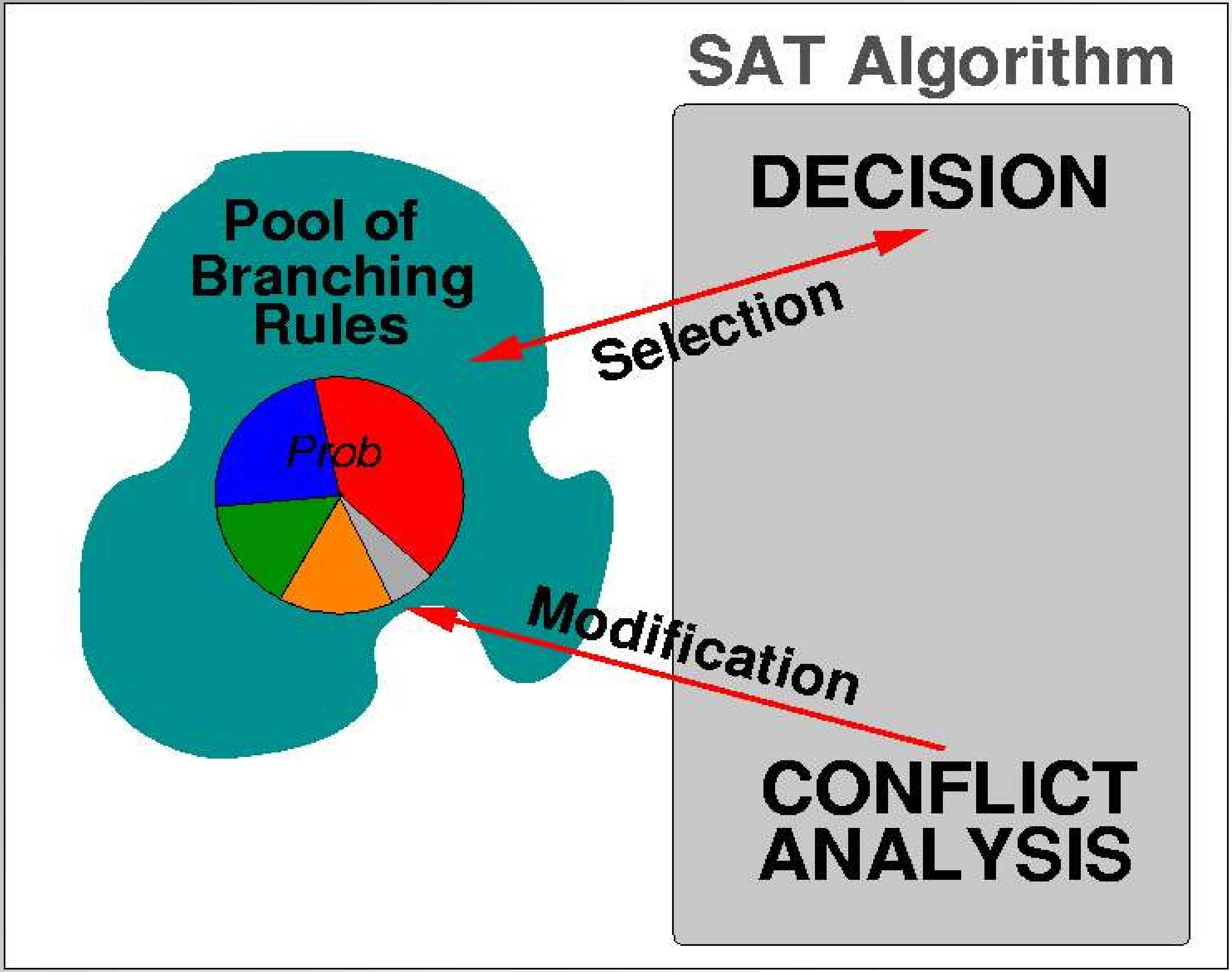
$Res(\mathbf{v_1},R_3,c_5)=(-v_8,+v_{10},+v_{17},+v_{19})$ [R_4]

$Res(\mathbf{v_3},R_2,c_6)=(-v_1,+v_{10},+v_{17},+v_{19})$ [R_3]

$Res(\mathbf{v_5},R_1,c_7)=(-v_1,-v_3,+v_{10},+v_{17},+v_{19})$ [R_2]

$Res(\mathbf{v_{18}},c_9,c_8)=(-v_1,-v_3,+v_5,+v_{17},+v_{19})$ [R_1]

Adaptive Framework



Adaptive Framework

Features of our approach:

- Set of Branching Rules: $B = \{\rho_1, \ldots, \rho_t\}$
- Attach preference value $Pref(\rho_i)$ where

$$0 \leq Pref(\rho_i) \leq 1$$

$$\sum Pref(\rho_i) = 1$$

- Branching Rule selection methods
- Conflict-based adaption of preference values

Selection Methods

3 selection methods

(known from theory of Genetic Algorithms):

- Roulette-Wheel (RW):
  - $Prob(\rho) = Pref(\rho)$
- Linear Ranking (LR):
  - $Prob(\rho) = Rank(\rho, B) \cdot \frac{2}{(n \cdot (n+1))}$
- k-Tournament (2T):
  - select randomly k elements from $B, B_k \subset B$
  - select $\rho_{sel} \in B_k$ with maximum preference value
  - $Pref(\rho_{sel}) = max_{\rho \in B_k}(Pref(\rho))$

Adaption of Preferences (1/5)

Observation

Conflicts are
- mandatory in unsatisfiable SAT instances to reduce search costs
- unessential in satisfiable SAT instances since search path without conflicts exists

Problem

How to determine solvability of SAT instance?

Adaption of Preferences (2/5)

Definition (Individual Averaged #C/#V Ratio):

For SAT instance I, set at the beginning

$$AR(I) = \frac{NoOfClauses(I)}{NoOfVariables(I)}$$

During search, after each conflict, update

$$AR_{new}(I) = \frac{1}{2} \cdot \left(AR_{old} + \frac{NoUnresolvedClauses(I)}{NoFreeVariables(I)} \right)$$

Now:

If $\frac{NoUnresolvedClauses(I)}{NoFreeVariables(I)} \leq AR_{old}(I):$

- → relatively less constrained
- → punishing mode

else

- → relatively more constrained
- → reward mode

Adaption of Preferences (3/5)

Definition (Conflict-triggering branching rule):
BR $\rho \in \mathcal{B}$ triggers a conflict iff
(1) A conflict occurs on decision level d
(2) Non-chronological backtracking backtracks to d´
(3) ρ was applied at decision level d´

Keep 2 counters for each branching rule ρ :

Used(ρ) = number of applications of ρ
Trigger(ρ) = number of conflicts triggered by ρ

Adaption of Preferences (4/5)

Now we can dynamically adapt preferences when ρ triggered a conflict:

$$Update(\rho) = 1 + (-1)^{mode} \cdot \frac{Trigger(\rho)}{Used(\rho)}$$

$$Pref_{new}(\rho) = Update(\rho) \cdot Pref_{old}(\rho)$$

($mode=1$ in punishing mode, $mode=0$ in reward mode)

➡ preference is decreased in punishing mode
➡ preference is increased in reward mode

Adaption of Preferences (5/5)

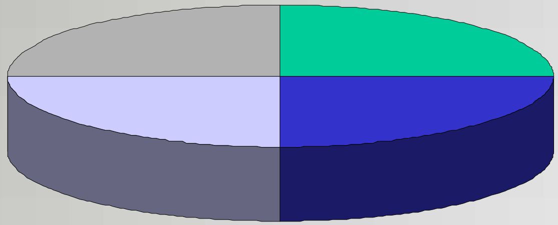
What else must be done?

- Difference distribution after update of preference
  - → uniform/weighted distribution
- Suitable initialization values
  - → Ranking of single-branching rule experiments wrt Time, #Aborts, both
  - → `Time-Rank, Abort-Rank, Time-Abort-Rank`
  - → we restrict to `Abort-Rank` (minimizing aborts has highest priority, gave best results!)

Difference Distribution

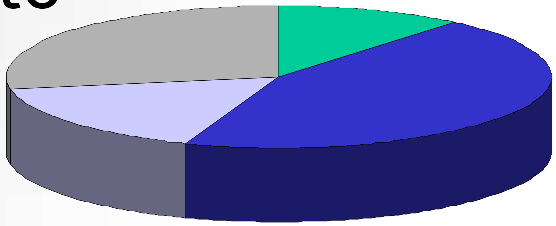
After update of preference value, difference between old and new preference must be distributed to the other branching rules!

- Uniform:



  each gets the same portion
- Weighted:

  each gets a portion proportional to its own preference value



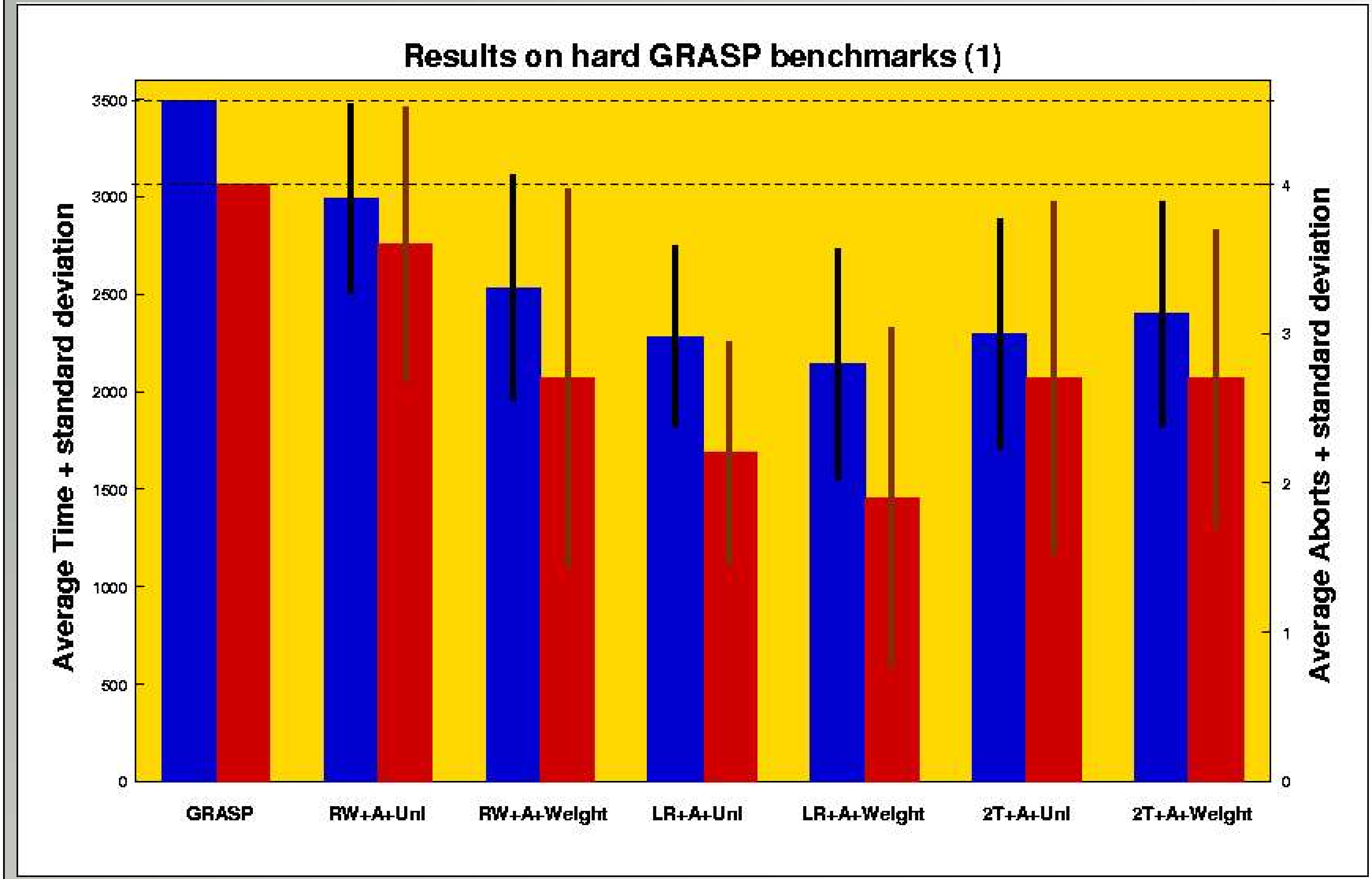
Experiments (1): Benchmarks

Name	# var	# clauses	status
bw_large.c	3016	50457	sat
bw_large.d	6325	131973	sat
e0ddr2-19-by-5-1	19500	103887	sat
e0ddr2-19-by-5-4	19500	104527	sat
enddr2-10-by-5-1	20700	111567	sat
enddr2-10-by-5-8	21000	113729	sat
ewddr2-10-by-5-1	21800	118607	sat
ewddr2-10-by-5-8	22500	123329	sat
hfo3.010.1	215	920	sat
hfo3.022.1	215	920	sat
hfo3.027.1	215	920	sat
qg5-10	1000	43636	unsat
qg7-11	1331	49534	unsat

Experimental Results (1)

Solver	Time		Aborts	
	Avg.	Std. Dev.	Avg.	Std. Dev.
GRASP-DLIS	3492		4	
RW+Abort+Uni	2989	488	3,60	0,92
RW+Abort+weight	2531	581	2,70	1,27
LR+Abort+uni	2281	467	2,20	0,75
LR+Abort+weight	2139	594	1,90	1,14
2T+Abort+uni	2294	594	2,70	1,19
2T+Abort+weight	2398	580	2,70	1,00

Experimental Results (1)



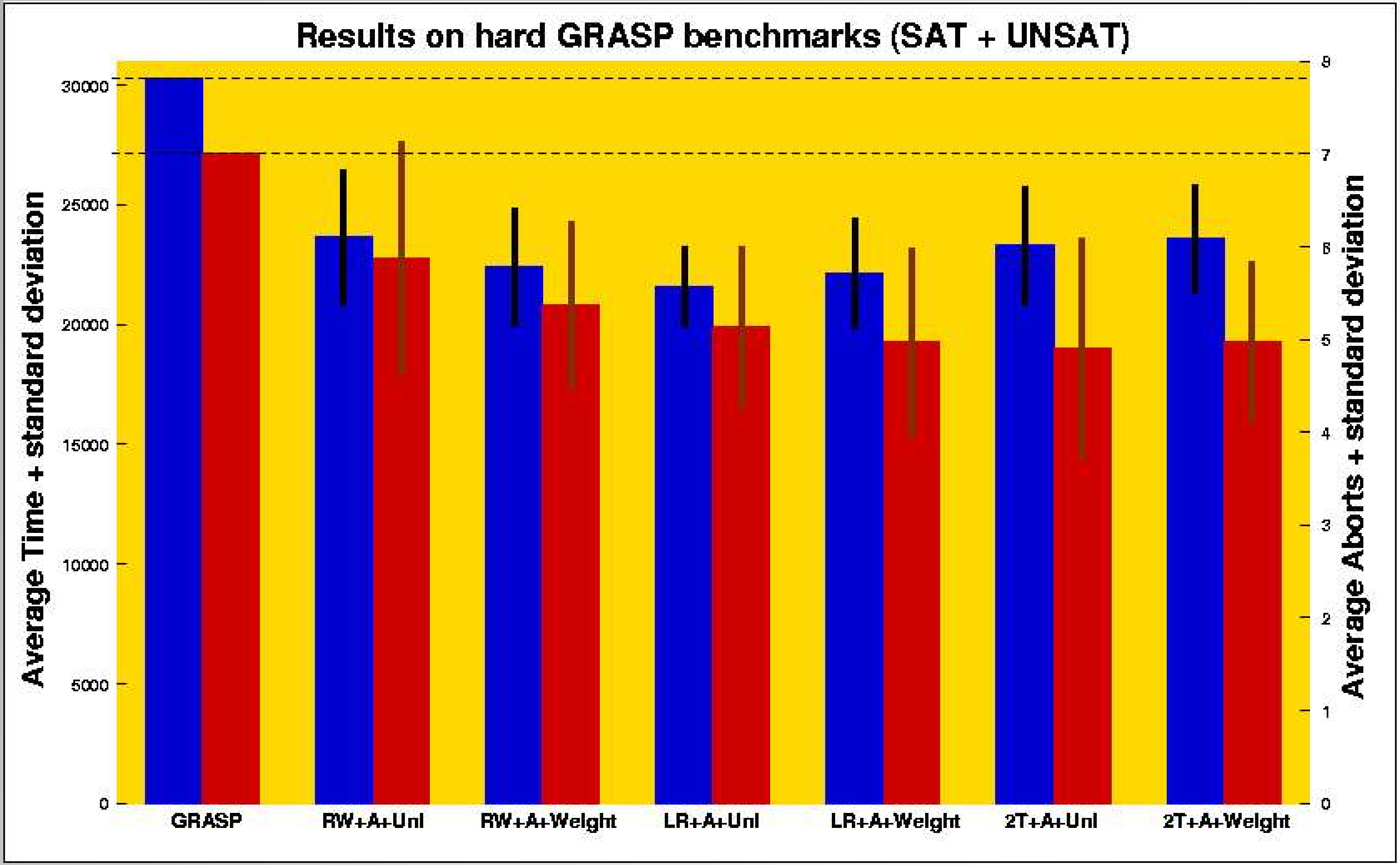
Experiments (2): Benchmarks

Name	#var	#clauses	status	GRASP
4blocks	758	47820	sat	1524.62
bw_large.d	6325	131973	sat	**3002.47**
c7552_bug	7559	20109	sat	94.16
hfo5.032.1	55	1163	sat	2826.72
hfo6.018.1	40	1745	sat	**3000.82**
hfo6.020.1	40	1745	sat	**3000.70**
barrel5	1407	5383	unsat	950.36
hfo3.002.0	215	920	unsat	**3000.78**
hfo3.015.0	215	920	unsat	**3000.50**
hfo3.035.0	215	920	unsat	**3000.45**
hfo3.039.0	215	920	unsat	**3000.65**
qg5-10	1000	43636	unsat	1011.89
qg6-11	1331	49204	unsat	1595.54
qg7-11	1331	49534	unsat	1278.77

Machine: Intel Xeon 2Ghz, 2GB RAM

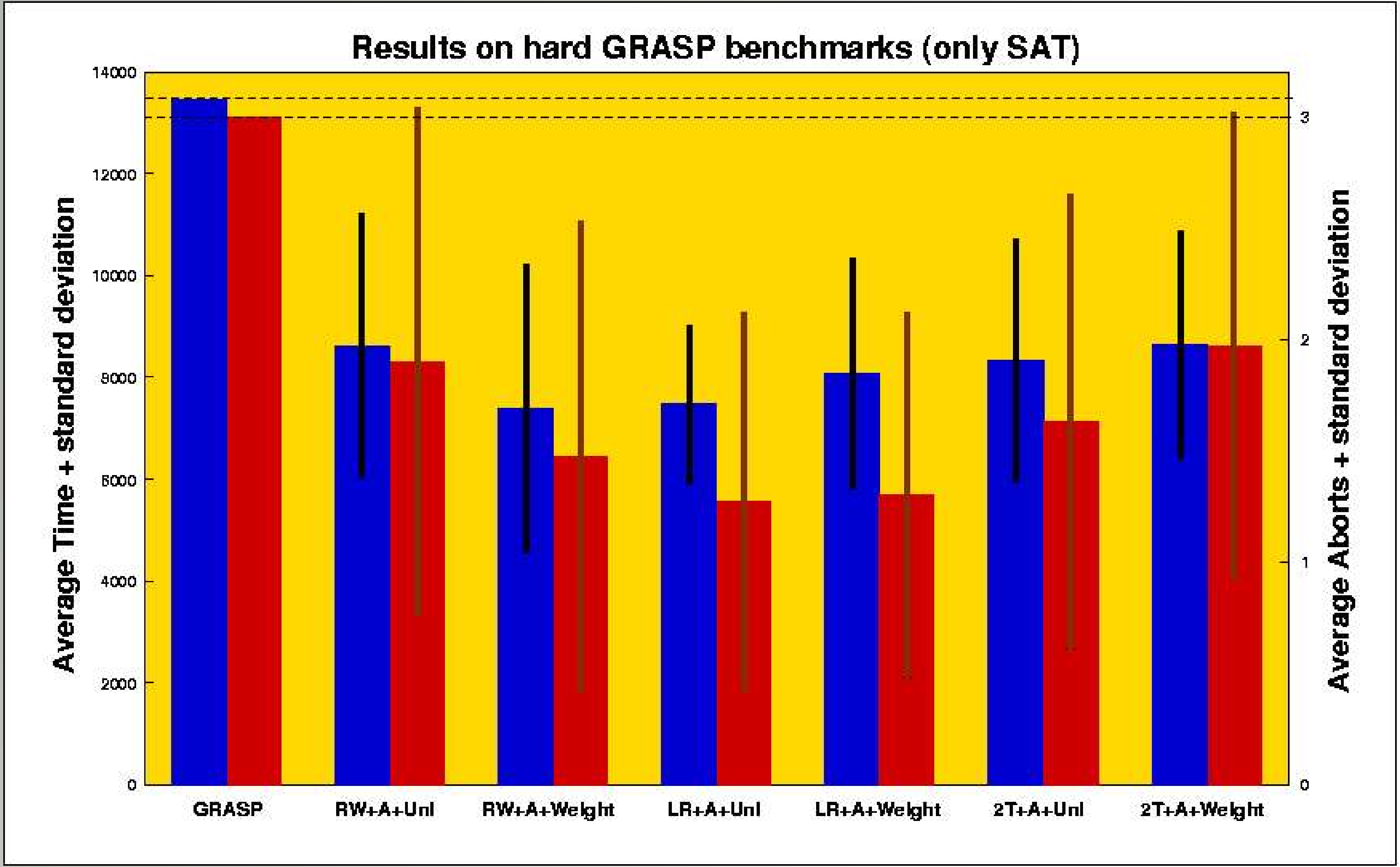
Time limit: 3000sec

Experimental Results (2)

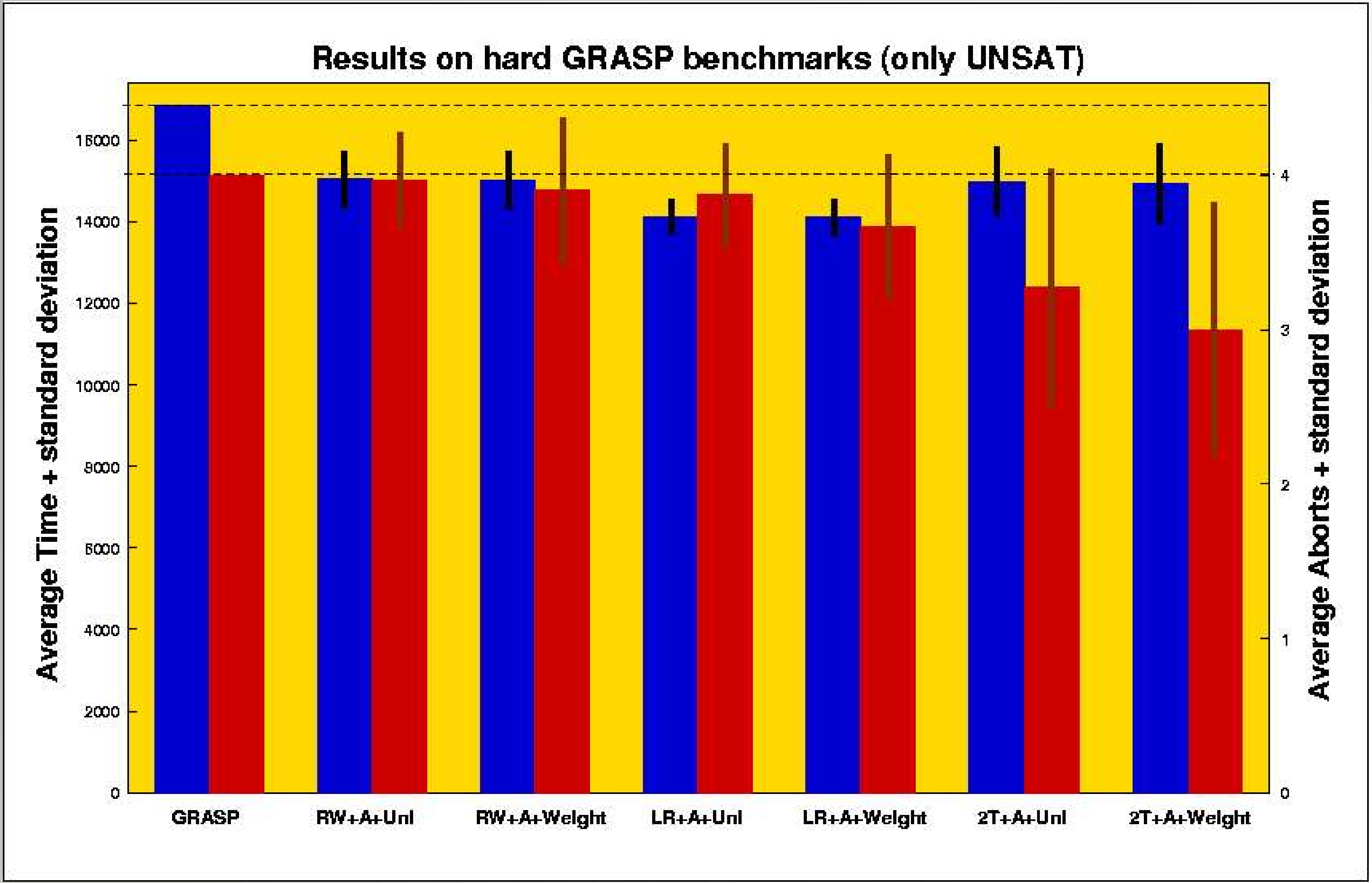


marc herbstritt sat2003

Experimental Results (2)



Experimental Results (2)



The times they are a changing!

- Project was initiated before Chaff
- Chaff changed a lot!
- What is the benefit of our work?
  - Solvability estimation is of high interest
  - Exploitation of several branching rules is helping
  - Seems to fit into portfolio concept and distributed computation

Future work

- Design of Chaff-like branching rules (see also BerkMin) with different flavors
- Robust solvability estimation measure
- Interaction between the SAT components
- ...

Conclusions

We presented

- an adaptive framework combining
  - multiple branching rules
  - information from conflict-analysis
- a definition to handle solvability status during SAT search

Experimental results show the feasibility.

Future work will target to transfer the framework to new class of SAT solvers (Chaff, ...)