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

<table>
<thead>
<tr>
<th>Branching Rule</th>
<th>Time</th>
<th>Aborts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Böhm</td>
<td>1817,45</td>
<td>8</td>
</tr>
<tr>
<td>MOM</td>
<td>1428,04</td>
<td>7</td>
</tr>
<tr>
<td>OS-J W</td>
<td>807,82</td>
<td>4</td>
</tr>
<tr>
<td>TS-J W</td>
<td>911,28</td>
<td>4</td>
</tr>
<tr>
<td>DLCS</td>
<td>746,3</td>
<td>3</td>
</tr>
<tr>
<td>DLIS</td>
<td>409,14</td>
<td>1</td>
</tr>
<tr>
<td>RDLIS</td>
<td>439,16</td>
<td>1,1</td>
</tr>
<tr>
<td>RAND</td>
<td>1431,85</td>
<td>5,7</td>
</tr>
</tbody>
</table>

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

\[ v_{12} = 0 \text{ due to } c_1 \]
\[ v_{16} = 1 \text{ due to } c_2 \]
\[ v_2 = 0 \text{ due to } c_3 \]
\[ v_{10} = 0 \text{ due to } c_4 \]
\[ v_1 = 1 \text{ due to } c_5 \]
\[ v_3 = 1 \text{ due to } c_6 \]
\[ v_5 = 0 \text{ due to } c_7 \]
\[ v_{18} = 1 \text{ due to } c_8 \]
\[ \text{conflict at } c_9 \text{ due to } v_{18} \]

\[ v_{12} = 0 \text{ due to } c_1 \]
\[ v_{16} = 1 \text{ due to } c_2 \]
\[ v_2 = 0 \text{ due to } c_3 \]
\[ v_{10} = 0 \text{ due to } c_4 \]
\[ v_1 = 1 \text{ due to } c_5 \]
\[ v_3 = 1 \text{ due to } c_6 \]
\[ v_5 = 0 \text{ due to } c_7 \]
\[ v_{18} = 1 \text{ due to } c_8 \]
\[ \text{conflict at } c_9 \text{ due to } v_{18} \]

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

\[
\begin{align*}
\text{Res}(v_1,R_3,c_5) &= (-v_8, +v_{10}, +v_{17}, +v_{19}) & [R_4] \\
\text{Res}(v_3,R_2,c_6) &= (-v_1, +v_{10}, +v_{17}, +v_{19}) & [R_3] \\
\text{Res}(v_5,R_1,c_7) &= (-v_1, -v_3, +v_{10}, +v_{17}, +v_{19}) & [R_2] \\
\text{Res}(v_{18},c_9,c_8) &= (-v_1, -v_3, +v_5, +v_{17}, +v_{19}) & [R_1]
\end{align*}
\]
Adaptive Framework

SAT Algorithm

DETECTION

Selection

Modification

CONFLICT ANALYSIS

Pool of Branching Rules

Prob
Adaptive Framework

Features of our approach:

- Set of Branching Rules: \( B = \{\rho_1, \ldots, \rho_t\} \)
- Attach preference value \( \text{Pref}(\rho_i) \) where
  \[
  0 \leq \text{Pref}(\rho_i) \leq 1 \\
  \sum \text{Pref}(\rho_i) = 1
  \]
- Branching Rule selection methods
- Conflict-based adaption of preference values
Selection Methods

3 selection methods
(known from theory of Genetic Algorithms):

- **Roulette-Wheel (RW):**
  \[ \text{Prob}(\boldsymbol{\rho}) = \text{Pref}(\boldsymbol{\rho}) \]

- **Linear Ranking (LR):**
  \[ \text{Prob}(\boldsymbol{\rho}) = \text{Rank}(\boldsymbol{\rho}, B) \cdot \frac{2}{(n \cdot (n+1))} \]

- **k-Tournament (2T):**
  - select randomly \( k \) elements from \( B, B_k \subset B \)
  - select \( \rho_{sel} \in B_k \) with maximum preference value
  \[ \text{Pref}(\rho_{sel}) = \max_{\rho \in B_k} (\text{Pref}(\rho)) \]
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?
Definition (Individual Averaged $\#C/\#V$ Ratio):
For SAT instance $I$, set at the beginning

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

During search, after each conflict, update

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

Now:
If

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

→ relatively less constrained
→ punishing mode

else

→ relatively more constrained
→ reward mode
Adaption of Preferences (3/5)

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

Keep 2 counters for each branching rule $\rho$:

- $\text{Used}(\rho) =$ number of applications of $\rho$
- $\text{Trigger}(\rho) =$ number of conflicts triggered by $\rho$
Now we can dynamically adapt preferences when $\rho$ triggered a conflict:

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

\[
Pref_{\text{new}}(\rho) = Update(\rho) \cdot Pref_{\text{old}}(\rho)
\]

($\text{mode}=1$ in punishing mode, $\text{mode}=0$ 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

<table>
<thead>
<tr>
<th>Name</th>
<th># var</th>
<th># clauses</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>bw_large.c</td>
<td>3016</td>
<td>50457</td>
<td>sat</td>
</tr>
<tr>
<td>bw_large.d</td>
<td>6325</td>
<td>131973</td>
<td>sat</td>
</tr>
<tr>
<td>e0ddr2-19-by-5-1</td>
<td>19500</td>
<td>103887</td>
<td>sat</td>
</tr>
<tr>
<td>e0ddr2-19-by-5-4</td>
<td>19500</td>
<td>104527</td>
<td>sat</td>
</tr>
<tr>
<td>enddr2-10-by-5-1</td>
<td>20700</td>
<td>111567</td>
<td>sat</td>
</tr>
<tr>
<td>enddr2-10-by-5-8</td>
<td>21000</td>
<td>113729</td>
<td>sat</td>
</tr>
<tr>
<td>ewddr2-10-by-5-1</td>
<td>21800</td>
<td>118607</td>
<td>sat</td>
</tr>
<tr>
<td>ewddr2-10-by-5-8</td>
<td>22500</td>
<td>123329</td>
<td>sat</td>
</tr>
<tr>
<td>hfo3.010.1</td>
<td>215</td>
<td>920</td>
<td>sat</td>
</tr>
<tr>
<td>hfo3.022.1</td>
<td>215</td>
<td>920</td>
<td>sat</td>
</tr>
<tr>
<td>hfo3.027.1</td>
<td>215</td>
<td>920</td>
<td>sat</td>
</tr>
<tr>
<td>qg5-10</td>
<td>1000</td>
<td>43636</td>
<td>unsat</td>
</tr>
<tr>
<td>qg7-11</td>
<td>1331</td>
<td>49534</td>
<td>unsat</td>
</tr>
</tbody>
</table>
## Experimental Results (1)

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time</th>
<th>Aborts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRASP-DLIS</td>
<td>3492</td>
<td></td>
</tr>
<tr>
<td>RW + Abort + Uni</td>
<td>2989</td>
<td>488</td>
</tr>
<tr>
<td>RW + Abort + weight</td>
<td>2531</td>
<td>581</td>
</tr>
<tr>
<td>LR + Abort + uni</td>
<td>2281</td>
<td>467</td>
</tr>
<tr>
<td>LR + Abort + weight</td>
<td>2139</td>
<td>594</td>
</tr>
<tr>
<td>2T + Abort + uni</td>
<td>2294</td>
<td>594</td>
</tr>
<tr>
<td>2T + Abort + weight</td>
<td>2398</td>
<td>580</td>
</tr>
</tbody>
</table>
Experimental Results (1)

Results on hard GRASP benchmarks (1)
## Experiments (2): Benchmarks

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

Machine: Intel Xeon 2Ghz, 2GB RAM

Time limit: 3000sec
Experimental Results (2)

Results on hard GRASP benchmarks (SAT + UNSAT)
Experimental Results (2)

Results on hard GRASP benchmarks (only SAT)
Experimental Results (2)

Results on hard GRASP benchmarks (only UNSAT)

Average Time + standard deviation

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Time + Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRASP</td>
<td>~18000</td>
</tr>
<tr>
<td>RW+A+Unl</td>
<td>~14000</td>
</tr>
<tr>
<td>RW+A+Weight</td>
<td>~12000</td>
</tr>
<tr>
<td>LR+A+Unl</td>
<td>~10000</td>
</tr>
<tr>
<td>LR+A+Weight</td>
<td>~8000</td>
</tr>
<tr>
<td>2T+A+Unl</td>
<td>~6000</td>
</tr>
<tr>
<td>2T+A+Weight</td>
<td>~4000</td>
</tr>
</tbody>
</table>
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, ...).