Conflict-Based Selection of Branching Rules in SAT Algorithms

Marc Herbstritt
Bernd Becker

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

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

Problem: SAT
Given: Boolean formula $\phi$ in clausal form (CNF) over variables $x_1, \ldots, x_n$
Question: Exists assignment $(v(x_1), \ldots, v(x_n))$ such that $\phi$ is satisfied?

Definition (Satisifiability)
- literal $l$ is sat iff $[l \equiv x \land v(x) = 1] \lor [l \equiv \neg x \land v(x) = 0]$
- clause $c = (l_1 \lor \ldots \lor l_k)$ is sat iff at least one literal $l_i$ is sat
- CNF $\varphi = (c_1 \land \ldots \land c_m)$ is sat iff all clauses $c_i$ are sat
- CNF $\phi$ is unsat iff no satisfiable assignment exists

Further notation: clause $c$ is unresolved iff partial assignment doesn't make $c$ sat or unsat
Applications: CEC

CEC: Combinational Equivalence Checking
  - Miter construction: $SPEC \oplus IMPL$
  - Based on efficient gate-to-CNF transformation

\[
x = A \neg D(w_1, \ldots, w_j)
\]
\[
\Rightarrow \quad \left[ \prod_{i=1}^{j} (w_i \lor \neg x) \right] \land \left[ (\sum_{i=1}^{j} \neg w_i) \lor x \right]
\]

\[
x = O \neg R(w_1, \ldots, w_j)
\]
\[
\Rightarrow \quad \left[ \prod_{i=1}^{j} (\neg w_i \lor x) \right] \land \left[ (\sum_{i=1}^{j} w_i) \lor \neg x \right]
\]

\[
x = N \neg T(w)
\]
\[
\Rightarrow \quad (x \lor w)(\neg x \lor \neg w)
\]
Applications: BMC

BMC: Bounded Model Checking

- Property Checking for bounded time frames
- Based on unrolling sequential circuit
- Formula $\mathcal{AG}p$ becomes

$$\varphi = I_0 \land \prod_{i=0}^{k-1} \rho(i, i+1) \land (\sum_{i=0}^{k-1} \neg P_i)$$

where

- $I_0$ is initial state
- $\rho(i, i+1)$ is transition between cycle $i$ and $i+1$
- $P_i$ is property in cycle $i$

- $\varphi$ is satisfiable iff reachable state exists in cycle $i$ which contradicts $P_i$
SAT Algorithm: Davis-Putnam

\[
\text{Davis-Putnam}(\varphi) \{ \\
\text{if } \varphi \text{ is empty return satisfiable} \\
\text{if } \epsilon \in \varphi \text{ return not satisfiable} \\
\text{if } \exists c_j \in \varphi \text{ with } s(c_j) = 1(c_j = \{l\}) \\
\quad \text{then} \\
\quad \text{/* unit propagation */} \\
\quad \text{satisfy } l \text{ and simplify } \varphi \text{ to } \varphi_l \\
\quad \text{return } \text{Davis-Putnam}(\varphi_l) \\
\text{else} \\
\text{/* branching rule */} \\
\text{select unassigned variable } x_i \text{ and an assignment } v(x_i) = a \\
\text{simplify } \varphi \text{ to } \varphi_{x_i} \\
\text{if } \text{Davis-Putnam}(\varphi_{x_i}) = \text{satisfiable} \\
\quad \text{then return satisfiable} \\
\quad \text{else} \\
\quad \text{change assignment of } x_i \text{ to } v(x_i) = \neg a \\
\quad \text{simplify } \varphi \text{ to } \varphi_{\neg x_i} \\
\text{return } \text{Davis-Putnam}(\varphi_{\neg x_i}) \\
\} \]
SAT Algorithms: New Features

- Intelligent Branching Rules
- Preprocessing
- Conflict analysis techniques
  - Non-chronological Backtracking
  - Conflict Learning
- Restarts
- Algorithm Portfolio
SAT Algorithms: Solver

- GRASP (Marques-Silva&Sakallah, 1996)
- rel_sat (Bayardo&Schrag, 1997)
- SATO (H.Zhang, 1997)
- Satz (Li, 1997)
- Chaff (Moskewicz&Madigan&Zhao&L.Zhang&Malik, 2001)
- BerkMin (Goldberg, 2002)
Branching Rules: Overview

- Böhm
- Maximum Occurences on Clauses of Minimum Size (MOM)
- Literal Count Heuristics:
  - Dynamic Largest Combined Sum (DLCS)
  - Dynamic Largest Individual Sum (DLIS)
  - Randomized DLIS (RDLIS)
- Random Selection (RAND)
Branching Rule: MOM

Maximum Occurences on Clauses of Minimum Size (MOM):

- Selects variable $x$ that maximizes
  \[
  \left[ f(x) + f(\neg x) \right] \cdot 2^k + f(x) \cdot f(\neg x)
  \]
  where $f(l)$ is number of occurrences of literal $l$ in the smallest unresolved clauses

- Gives preference to variables occurring frequently as positive and/or negative literal in many small clauses

- MOM is used in SAT solver SATZ (Li&Anbulagan 1997) using $k=10$
Branching Rule: DLIS

Dynamic Largest Individual Sum (DLIS):

- For a variable $x$, compute
  
  $$UC_p(x) = \{ c \mid A \ x \in c \land c \ is \ unresolved \}$$
  
  $$UC_n(x) = \{ c \mid A \neg x \in c \land c \ is \ unresolved \}$$

- Selects variable $x$ where
  
  $$x = \arg \max_y (UC_p(y), UC_n(y))$$

- Gives preference to variables occurring often either positive or negative

- Used in SAT solver GRASP
  
  (Marques-Silva & Sakallah, 1997)
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-JW</td>
<td>807.82</td>
<td>4</td>
</tr>
<tr>
<td>TS-JW</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
Conflict Analysis (1/4)

- Reasons (unit clauses) are stored for each implication
- Backward traversal of implication graph
  - implicitly stored in assignment stack
- Identification of UIPs (Unique Implication Point)
- Conflict Clause is generated and backtrack level derived from this clause
Initial clause database:

\[
c_1: (+v_6,-v_{11},-v_{12})
\]

\[
c_2: (-v_{11},+v_{13},+v_{16})
\]

\[
c_3: (-v_2,+v_{12},-v_{16})
\]

\[
c_4: (+v_2,-v_4,-v_{10})
\]

\[
c_5: (+v_1,-v_8,+v_{10})
\]

\[
c_6: (+v_3,+v_{10})
\]

\[
c_7: (-v_5,+v_{10})
\]

\[
c_8: (-v_1,-v_3,+v_5,+v_{17},+v_{18})
\]

\[
c_9: (-v_3,-v_{18},-v_{19})
\]

<table>
<thead>
<tr>
<th>DL</th>
<th>Var</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>v17</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>v19</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>v4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>v6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>v13</td>
<td>0</td>
</tr>
</tbody>
</table>

Now assume:

\[
c_1: (+v_6,-v_{11},-v_{12})
\]

\[
c_2: (-v_{11},+v_{13},+v_{16})
\]

\[
c_3: (-v_2,+v_{12},-v_{16})
\]

\[
c_4: (+v_2,-v_4,-v_{10})
\]

\[
c_5: (+v_1,-v_8,+v_{10})
\]

\[
c_6: (+v_3,+v_{10})
\]

\[
c_7: (-v_5,+v_{10})
\]

\[
c_8: (-v_1,-v_3,+v_5,+v_{17},+v_{18})
\]

\[
c_9: (-v_3,-v_{18},-v_{19})
\]
Conflict Analysis (3/4)

@ decision level 6:
- $c_1: (+v_6, -v_{11}, -v_{12})$
- $c_2: (-v_{11}, +v_{13}, +v_{16})$
- $c_3: (-v_2, +v_{12}, -v_{16})$
- $c_4: (+v_2, -v_4, -v_{10})$
- $c_5: (+v_1, -v_8, +v_{10})$
- $c_6: (+v_3, +v_{10})$
- $c_7: (-v_5, +v_{10})$
- $c_8: (-v_1, -v_3, +v_5, +v_{17}, +v_{18})$
- $c_9: (-v_3, -v_{18}, -v_{19})$

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 $c_4$
- $v_1=1$ due to $c_5$
- $v_3=1$ due to $c_6$
- $v_5=0$ due to $c_7$
- $v_{18}=1$ due to $c_8$
- conflict at $c_9$ due to $v_{18}$
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 \( c_4 \)
- \( v_1 = 1 \) due to \( c_5 \)
- \( v_3 = 1 \) due to \( c_6 \)
- \( v_5 = 0 \) due to \( c_7 \)
- \( v_{18} = 1 \) due to \( c_8 \)
- conflict at \( c_9 \) due to \( v_{18} \)

Conflict Analysis (4/4)

- 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

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

Features of our approach:

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

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

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

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

- **k-Tournament (2T):**
  - select randomly \( k \) elements from \( B, B_k \subset B \)
  - select \( \rho_{\text{sel}} \in B_k \) with maximum preference value
  \[ \text{Pref}(\rho_{\text{sel}}) > \min_{\rho \in B_k} \left( \text{Pref}(\rho) \right) \]
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
\[
AR(I) = \frac{\text{NoOfClauses}(I)}{\text{NoOfVariables}(I)}
\]
During search, after each conflict, update \( AR(I) \)
\[
AR_{new}(I) = \frac{1}{2} \left( AR_{old} + \frac{\text{NoUnresolvedClauses}(I)}{\text{NoFreeVariables}(I)} \right)
\]

Now:
if \[
\frac{\text{NoUnresolvedClauses}(I)}{\text{NoFreeVariables}(I)} < AR_{old}(I)
\]
→ relatively less constrained
→ punishing mode
else
→ relatively more constrained
→ reward mode
Adaption of Preferences (3/5)

Definition (Conflict-triggering branching rule):
BR $\rho \in B$ 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 $\rho$:

Used($\rho$) = number of applications of $\rho$
Trigger($\rho$) = number of conflicts triggered by $\rho$
Adaption of Preferences (4/5)

Now we can dynamically adapt preferences when $\rho$ 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 \( \text{Pref}(\rho_i) \)
  - uniform/weighted distribution
- Suitable initialization values
  - Ranking of single-branching rule experiments wrt Time, #Aborts, both
    - Time-Rank, Abort-Rank, Time-Abort-Rank
### Experiments: 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

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time Average</th>
<th>Time Std. Deviation</th>
<th>Aborts Average</th>
<th>Aborts Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRASP-DLIS</td>
<td>3492</td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>RW+Abort+Uni</td>
<td>2989</td>
<td>488</td>
<td>3.60</td>
<td>0.92</td>
</tr>
<tr>
<td>RW+Abort+weight</td>
<td>2531</td>
<td>581</td>
<td>2.70</td>
<td>1.27</td>
</tr>
<tr>
<td>LR+Abort+uni</td>
<td>2281</td>
<td>467</td>
<td>2.20</td>
<td>0.75</td>
</tr>
<tr>
<td>LR+Abort+weight</td>
<td>2139</td>
<td>594</td>
<td>1.90</td>
<td>1.14</td>
</tr>
<tr>
<td>2T+Abort+uni</td>
<td>2294</td>
<td>594</td>
<td>2.70</td>
<td>1.19</td>
</tr>
<tr>
<td>2T+Abort+weight</td>
<td>2398</td>
<td>580</td>
<td>2.70</td>
<td>1.00</td>
</tr>
</tbody>
</table>
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, ...)

marc herbstritt
26 gi/itg/gmm 2003