Learned Clause Minimization in Parallel SAT Solvers

Pragmatics of SAT 2019

Marc Hartung, Florian Schintke
1. Background
2. Parallel Clause Minimization
3. Experiments
4. Conclusion
Background
## (Learned) Clause Minimization in SC18

<table>
<thead>
<tr>
<th>Solver</th>
<th>Author</th>
<th>CM/LMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapleLCMDistChronoBT</td>
<td>Ryvchin et al.</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_LCM_Scavel_fix2</td>
<td>Xu et al.</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_CM</td>
<td>Luo et al.</td>
<td>✔</td>
</tr>
<tr>
<td>cms55-main-all4fixed</td>
<td>M. Soos</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_CM_ordUIP</td>
<td>Luo et al.</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_CM_Dist</td>
<td>Luo et al.</td>
<td>✔</td>
</tr>
<tr>
<td>cms55-main-all4fixed</td>
<td>M. Soos</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_CM_ordUIP+</td>
<td>Luo et al.</td>
<td>✔</td>
</tr>
<tr>
<td>Maple_LCM_Scavel_200_fix2</td>
<td>Xu et al.</td>
<td>✔</td>
</tr>
<tr>
<td>cms55-main-all4fixed</td>
<td>M. Soos</td>
<td>✔</td>
</tr>
</tbody>
</table>

### Success was not transferred to parallel

#### Top10 Main Track

<table>
<thead>
<tr>
<th>Solver</th>
<th>Author</th>
<th>CM/LMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm5s55-main-all4fixed</td>
<td>M. Soos</td>
<td>✔</td>
</tr>
</tbody>
</table>

#### Top10 Parallel Track

<table>
<thead>
<tr>
<th>Solver</th>
<th>Author</th>
<th>CM/LMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm5s55-parallel, 12 core</td>
<td>M. Soos</td>
<td>✔</td>
</tr>
<tr>
<td>cbpenelope</td>
<td>T. Sonobe</td>
<td>✘</td>
</tr>
<tr>
<td>ccpenelope</td>
<td>T. Sonobe</td>
<td>✘</td>
</tr>
<tr>
<td>syrup, 24 threads</td>
<td>Audemard et al.</td>
<td>✔</td>
</tr>
<tr>
<td>penelope_MDLC</td>
<td>Konan Tchinda et al.</td>
<td>✘</td>
</tr>
<tr>
<td>treengeling</td>
<td>A. Biere</td>
<td>✘</td>
</tr>
<tr>
<td>scalope</td>
<td>Konan Tchinda et al.</td>
<td>✘</td>
</tr>
</tbody>
</table>

---

Success was not transferred to parallel.
(Learned) Clause Minimization (LCM)

- Applied at decision level zero

After propagating $\neg l_1, \neg l_2, \ldots, \neg l_i$:

**Case 1:**
- $l_j$ propagated to true
- $C$ replaced by $l_1 \lor l_2 \lor \ldots \lor l_i \lor l_j$

**Case 2:**
- $l_i$ propagated to false
- $C$ replaced by $l_1 \lor l_2 \lor \ldots \lor l_i$

**Case 3:**
- Conflict detected
- $C$ replaced by $l_1 \lor l_2 \lor \ldots \lor l_i$

- In this presentation: Minimization ≡ Distillation/Vivification
LMC Approach [3]

Only apply CM to (in future) kept learned clauses

• Each clause minimized only once
• Reduction heuristic specifies which are kept
• Reduction example Glucose:

- Low LBD, higher activity → keep and minimize
- High LBD, lower activity → remove

Minimization triggered after a restart or decision tree is stashed
Parallel Clause Minimization
Heterogeneous vs. Homogenous

Heterogeneous minimization approach

Dedicate individual threads to minimization

- Examples:
  - CDCL solvers + One minimization thread [4][5]
  - Only part of solvers use minimization [6]

- Problems:
  - Not trivial for many cores
  - Introduces load balancing problem
  - Adds more magic parameters

- Finding good parameters expensive

→ Discarded for future work
Heterogeneous vs. Homogenous

Homogenous minimization approach

- All solvers use same minimization approach
- Example: Minimize export clauses [7]
- Problems:
  - Balance minimization and BCP
  - How and if minimizations should be shared
PCM – Private Clause Minimization

- Directly apply LCM approach
- Export and CM are independent
- No intentional sharing of minimizations
- Using lazy export policy: Minimized clauses might be shared

Implementation:
- LBD (≤ 5) cut
- Original version (no LBD cut) decreased performance
- Lazy export policy (two times used)
LPCM – Linked Private Clause Minimization

- Shared clauses are linked
- Minimizations shared via link

Implementation:
- LBD (≤ 5) cut
  - Clause header contains pointer to memory chunk
  - If minimized, chunk contains new clause
ECM – Export Clause Minimization

- Enforce minimization before export
- Already used in TopoSAT2[7]

Implementation:
- Lazy export policy (two times used)
- LBD ($\leq 3$ or $\leq 4$) and length ($\leq 30$) cut
- Marked clauses are protected during reduction
Experiments
Test Set and Environment

- SAT competition ‘16 application track, ’17 and ‘18 main track
- On Intel Xeon Phi 7250, 68 cores at 1.4 GHz with 96 GB RAM
- Maximum walltime of 15000 seconds
- Maximal 34 threads per solver
- Restrictions due to CPU frequency, cache and main memory
Vivification Overhead

**SAT**

- Propagation Overhead:
  - \((L)PCM \approx 10\%\)
  - ECM3 \approx 1\%
  - ECM4 \approx 4\%
  - on average

- Minimization success correlates with overhead:
  - \((L)PCM \approx 40\%\)
  - ECM3 \approx 6\%
  - ECM4 \approx 32\%
  - on average

**UNSAT**
Syrup Runtime SAT

PCM increases SAT performance

Improvement to Syrup small for easy instances

Solved instances:
- Syrup: 333
- PCM: 343
- LPCM: 331
- ECM3: 326
- ECM4: 302
Syrup Runtime UNSAT

- ECM increases overall UNSAT performance
- PCM, LPCM and ECM3 improve performance
Parallel CM Solver

TopoSAT2 – ECM

- Glucose 3.0 based ECM solver
- Direct clause export
- Copies of clauses are minimized and exported
  → Minimizations are not used by minimizing solver

Sticky – LPCM, ECM

- Glucose 4.0 based solver with physical clause sharing
- No copy-sharing of clauses, only references are shared
- Adapted lazy clause sharing heuristic
SAT Competition Results

Results SC’16 (application track), SC’17, SC’18 (main track)

- Overall increase through nearly every CM approach
- Syrup-PCM nearly closed gap to Toposat2
- LPCM and ECM3 decrease SAT but increase solved UNSAT instances more
- TopoSAT2-ECM3 decrease:
  - No lazy export $\rightarrow$ missing activity filter for export $\rightarrow$ higher overhead
  - Minimizations not inserted in minimizing solver

<table>
<thead>
<tr>
<th>Solver</th>
<th>SAT</th>
<th>UNSAT</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syrup</td>
<td>333</td>
<td>347</td>
<td>680</td>
</tr>
<tr>
<td>Syrup-PCM</td>
<td><strong>343</strong></td>
<td>355</td>
<td><strong>698</strong></td>
</tr>
<tr>
<td>Syrup-LPCM</td>
<td>331</td>
<td>355</td>
<td>686</td>
</tr>
<tr>
<td>Syrup-ECM3</td>
<td>326</td>
<td><strong>367</strong></td>
<td>693</td>
</tr>
<tr>
<td>Syrup-ECM4</td>
<td>303</td>
<td>320</td>
<td>623</td>
</tr>
<tr>
<td>Sticky</td>
<td>303</td>
<td>307</td>
<td>610</td>
</tr>
<tr>
<td>Sticky-LPCM</td>
<td>298</td>
<td><strong>333</strong></td>
<td>631</td>
</tr>
<tr>
<td>Sticky-ECM3</td>
<td>296</td>
<td><strong>333</strong></td>
<td>629</td>
</tr>
<tr>
<td>TopoSAT2</td>
<td><strong>357</strong></td>
<td>344</td>
<td><strong>701</strong></td>
</tr>
<tr>
<td>TopoSAT2-ECM3</td>
<td>353</td>
<td>326</td>
<td>679</td>
</tr>
</tbody>
</table>
## SAT Competition Results

### Single Competition Results

<table>
<thead>
<tr>
<th>Solver</th>
<th>SAT’16A</th>
<th>SAT’17</th>
<th>SAT’18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAT</td>
<td>UNSAT</td>
<td>ALL</td>
</tr>
<tr>
<td>Syrup</td>
<td>77</td>
<td>113</td>
<td>190</td>
</tr>
<tr>
<td>Syrup-PCM</td>
<td>78</td>
<td>115</td>
<td>193</td>
</tr>
<tr>
<td>Syrup-LPCM</td>
<td>77</td>
<td>116</td>
<td>193</td>
</tr>
<tr>
<td>Syrup-ECM3</td>
<td>76</td>
<td>122</td>
<td>198</td>
</tr>
<tr>
<td>Syrup-ECM4</td>
<td>72</td>
<td>108</td>
<td>180</td>
</tr>
<tr>
<td>Sticky</td>
<td>63</td>
<td>93</td>
<td>156</td>
</tr>
<tr>
<td>Sticky-LPCM</td>
<td>68</td>
<td>104</td>
<td>172</td>
</tr>
<tr>
<td>Sticky-ECM3</td>
<td>66</td>
<td>102</td>
<td>168</td>
</tr>
<tr>
<td>TopoSAT2</td>
<td>80</td>
<td>116</td>
<td>196</td>
</tr>
<tr>
<td>TopoSAT2-ECM3</td>
<td>75</td>
<td>109</td>
<td>184</td>
</tr>
</tbody>
</table>

Syrup-PCM wins on SC’18 application track benchmarks

Syrup-ECM3 wins on SC’16 application track benchmarks

No real improvement on SC’17 benchmarks
Conclusion

- Homogeneous CM applicable for parallel solvers
  → Approaches solved 6 – 21 additional instances
- Sharing minimizations via link has no advantage
  → LPCM fewer solved instances than PCM
- More restrictive clause selection than in serial
  → ECM4 and TopoSAT2-ECM slow down
  → PCM/LPCM only succeed with LBD cut
- Prioritize:
  - Activity-based selection for SAT (PCM)
  - LBD-based selection for UNSAT (ECM)
References


Questions?

Marc Hartung
Parallel and Distributed Computing
Zuse Institute Berlin (ZIB)
hartung@zib.de