SAT 2021 Barcelona, July 5-9th 2021

# Deep Cooperation of CDCL and Local Search for SAT

Shaowei Cai*, Xindi Zhang

Institute of Software, Chinese Academy of Sciences

{caisw, zhangxd}@ios.ac.cn

# Outline

- Introduction
- Main Ideas
  - A Novel Framework of hybrid solvers
  - Phase Resetting with Local Search Assignments
  - Branching with Conflict Frequency in Local Search
- Experiments

# Outline

- Introduction
- Main Ideas
  - A Novel Framework of hybrid solvers
  - Phase Resetting with Local Search Assignments
  - Branching with Conflict Frequency in Local Search
- Experiments

# Introduction - SAT

SAT: Given a propositional formula $\varphi$, test whether there is an assignment to the variables that makes $\varphi$ true.

e.g., a CNF formula

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (x_2 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4)$$

Hard

- The first problem that is proved to be NP-Complete [S. Cook, 1971]
- ETH says 3-SAT cannot be solved in $2^{o(n)}$ time, and SETH says $k$-SAT needs roughly $2^n$ time for large k.

Important Applications

- EDA
- Software verification
- Automatic Theorem Proving
- cryptography
- …

Improve the efficiency of SAT Solving

# Introduction - CDCL

- The most popular approach: CDCL, since 1996 (evolved from DPLL)

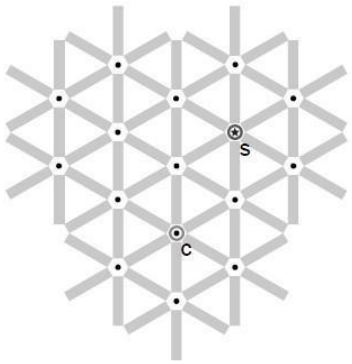
**Algorithm 1**: Typical CDCL algorithm: CDCL(F, $\alpha$)

```
1  dl ← 0;                    //decision level
2  if UnitPropagation(F,α)==CONFLICT then return UNSAT
3  while ∃ unassigned variables do
       /* PickBranchVar picks a variable to assign and
          picks the respective value                    */
4      (x, v) ← PickBranchVar(F, α);
5      dl ← dl + 1;
6      α ← α ∪ {(x, v)};
7      if UnitPropagation(F,α)==CONFLICT then
8          bl ← ConflictAnalysis(F, α);
9          if bl < 0 then
10             return UNSAT;
11         else
12             BackTrack(F, α, bl);
13             dl ← bl;
14 return SAT;
```

- clause learning
- Lazy data structures
- Restarting
- branching heuristics
  - Pick a variable
  - Pick the respective phase
- …

# Introduction - SLS

- The other important paradigm: stochastic local search (SLS), since 1992
  - a main incomplete method biased towards the satisfiable side.
  - Begin with a complete assignment and iteratively modify the assignment




# Introduction – Challenge on hybrid solving

Ten Challenges in Propositional Reasoning and Search

Bart Selman, Henry Kautz, and David McAllester
AT&T Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
{selman, kautz, dmac}@research.att.com
http://www. research, att.com/~selman/challenge

**Challenge 7**: Demonstrate the successful combination of stochastic search and systematic search techniques, by the creation of a new algorithm that outperforms the best previous examples of both approaches.

---**AAAI 1997**, Bart Selman, Henry Kautz and David McAllester

# Introduction – Related works

- Use a local search solver as the main body solver.
  - hybridGM, SATHYS
  - GapSAT: use CDCL as preprocessor before local search
- DPLL/CDCL as the main body solver
  - HINOTOS: local search finds subformulas for CDCL to solve
  - WalkSatz: calls WalkSAT at each node of a DPLL solver Satz.
  - CaDiCaL and Kissat: a local search solver is called when the solver resets the saved phases and is used only once immediately after the local search process
- Sequential call local search and CDCL
  - Sparrow2Riss, CCAnr+glucose, SGSeq

# Introduction – Related works

- Use a local search solver as the main body solver.
  - hybridGM, SATHYS
  - GapSAT: use CDCL as preprocessor before local search
- DPLL/CDCL as the main body solver
  - HINOTOS: local search finds subformulas for CDCL to solve

Previous works did not lead to hybrid solvers essentially better than CDCL solvers on application instances.

the local search process

- Sequential call local search and CDCL
  - Sparrow2Riss, CCAnr+glucose, SGSeq

# Outline

- Introduction
- Main Ideas
  - A Novel Framework of hybrid solvers
  - Phase Resetting with Local Search Assignments
  - Branching with Conflict Frequency in Local Search
- Experiments

# Idea 1: Exploring Promising Branches by Local Search

Identify which branches deserve exploration

$\frac{|\alpha|}{|V|} > p$ and there is no conflict under $\alpha$. $p = 0.4$

$\frac{|\alpha|}{|\alpha\_max|} > q$ and there is no conflict under $\alpha$. $q = 0.6$

The cutoff of each Local Search process: certain amount of memory accesses ($5 \times 10^7$)

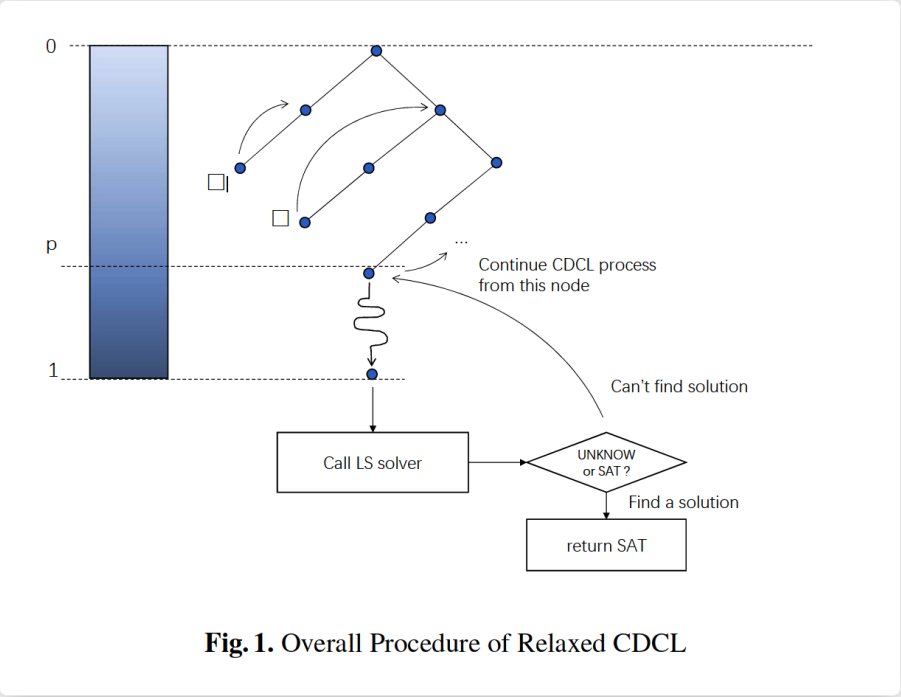



**Fig. 1.** Overall Procedure of Relaxed CDCL

# Idea 2: Phase Resetting with Local Search Assignments

- Phase selection is an important component of a CDCL solver.
- Most modern CDCL solvers utilize the phase saving heuristic (Pipatsrisawat & Darwiche, SAT 2007).

- Our idea:
  - After each time the CDCL is restarted, resets the saved phases of all variables with assignments produced by local search.

**Table 1.** Probability of different phases in our phase resetting mechanism

| Phase Name | $\alpha\_max\_LS[x]$ | $\alpha\_latest\_LS[x]$ | $\alpha\_best\_LS[x]$ | no change |
|---|---|---|---|---|
| Probability | 20% | 65% | 5% | 10% |

- $\alpha\_max\_LS$ and $\alpha\_best\_LS$ serve for the aim to maximize the depth of the branch
- $\alpha\_latest\_LS$ adds diversification

# Idea 3: Branching with Conflict Frequency in Local Search

- CDCL is a powerful framework owing largely to the utilization of the conflict information
- branching heuristics aim to promote conflicts.
- Can information from SLS be used to enhance branching heuristics to promote conflicts?

Our idea:

$ls\_confl\_freq$ (x) = #(steps in which x appears in unsatisfied clauses) / #total_local_search_steps

multiply $ls\_confl\_freq(\mathrm{x})$ with 100 , resulting $ls\_confl\_num(\mathrm{x})$.

**LS Enhanced VSIDS**: for each variable $x$, its activity is increased by $ls\_confl\_num(x)$

**LS Enhanced LRB**: for each variable $x$, the number of learnt clause during its period $I$ is creased by $ls\_confl\_num(x)$.

# Outline

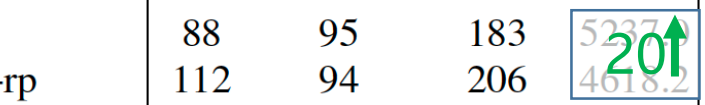
- Introduction
- Main Ideas
  - A Novel Framework of hybrid solvers
  - Phase Resetting with Local Search Assignments
  - Branching with Conflict Frequency in Local Search
- Experiments

# Experiments - Preliminaries

- Base solver:
  - glucose (v4.2.1)
  - MapleLCMDistChronoBT-DL (v2.1)
  - Kissat_sat (2414b6d)
  - CCAnr
- Benchmarks:
  - The latest four SAT Competitions/Race (2017-2020)
  - US Federal Communication Commission(FCC) 10000 instances

# Experiments – Results on SC benchmarks

| solver | #SAT | #UNSAT | #Solved | PAR2 | #SAT | #UNSAT | #Solved | PAR2 |
|---|---|---|---|---|---|---|---|---|
| | SC2017(351) | | | | SC2018(400) | | | |
| glucose_4.2.1 | 83 | 101 | 184 | 5220.0 | 95 | 95 | 190 | 5745.9 |
| glucose+rx | 88 | 95 | 183 | 5237.0 | 113 | 95 | 208 | 5282.4 |
| glucose+rx+rp | 112 | 94 | 206 | 4618.2 | 141 | 87 | 228 | 4698.3 |
| glucose+rx+rp+cf | 110 | 94 | 204 | 4668.5 | 150 | 91 | 241 | 4438.2 |
| Maple-DL-v2.1 | 101 | 113 | 214 | 4531.0 | 133 | 102 | 235 | 4533.9 |
| Maple-DL+rx | 101 | 112 | 213 | 4592.3 | 149 | 101 | 250 | 4148.4 |
| Maple-DL+rx+rp | 111 | 103 | 214 | 4447.1 | 158 | 93 | 251 | 4447.1 |
| Maple-DL+rx+rp+cf | 116 | 107 | 223 | 4139.4 | 162 | 97 | 259 | 3927.6 |
| Kissat_sat | 115 | 114 | 229 | 3943.5 | 167 | 98 | 265 | 3786.4 |
| Kissat_sat+cf | 113 | 113 | 226 | 4001.0 | 178 | 104 | 282 | 3409.4 |
| CCAnr | 13 | N/A | 13 | 9629.9 | 56 | N/A | 56 | 8622.0 |
| | SC2019(400) | | | | SC2020(400) | | | |
| glucose_4.2.1 | 118 | 86 | 204 | 5437.6 | 68 | 91 | 159 | 6494.6 |
| glucose+rx | 120 | 84 | 204 | 5443.0 | 93 | 88 | 181 | 6018.4 |
| glucose+rx+rp | 134 | 85 | 219 | 5096.3 | 130 | 85 | 215 | 5125.7 |
| glucose+rx+rp+cf | 140 | 85 | 225 | 4923.6 | 134 | 87 | 221 | 4977.9 |
| Maple-DL-v2.1 | 143 | 97 | 240 | 4601.8 | 86 | 104 | 190 | 5835.7 |
| Maple-DL+rx | 146 | 93 | 239 | 4602.1 | 121 | 105 | 226 | 4977.8 |
| Maple-DL+rx+rp | 155 | 94 | 249 | 4416.3 | 142 | 99 | 241 | 4589.2 |
| Maple-DL+rx+rp+cf | 154 | 95 | 249 | 4377.4 | 151 | 106 | 257 | 4171.1 |
| Kissat_sat | 159 | 88 | 247 | 4293.2 | 146 | 114 | 260 | 4048.8 |
| Kissat_sat+cf | 162 | 90 | 252 | 4211.7 | 157 | 113 | 270 | 3896.8 |
| CCAnr | 13 | N/A | 13 | 9678.3 | 45 | N/A | 45 | 8978.7 |



| | Analysis for SAT | | | | Analysis for UNSAT | |
|---|---|---|---|---|---|---|
| solver | #by LS | #SAT_bonus | #LS_call | LS_time(%) | #LS_call | LS_time(%) |
| | | | SC2017(351) | | | |
| glucose+rx | 20 | 11 | 24.28 | 21.66 | 16.36 | 5.52 |
| glucose+rx+rp | 10 | 33 | 17.77 | 18.46 | 14.33 | 4.86 |
| glucose+rx+rp+cf | 17 | 29 | 22.7 | 22.19 | 15.3 | 5.81 |
| Maple+rx | 16 | 9 | 13.86 | 7.52 | 11.18 | 2.03 |
| Maple+rx+rp | 11 | 15 | 9.63 | 10.43 | 6.54 | 2.36 |
| Maple+rx+rp+cf | 6 | 16 | 12.59 | 7.49 | 8.59 | 2.12 |
| | | | SC2018(400) | | | |
| glucose+rx | 50 | 4 | 11.27 | 20.66 | 29.62 | 4.94 |
| glucose+rx+rp | 47 | 31 | 9.46 | 18.4 | 21.66 | 5.64 |
| glucose+rx+rp+cf | 53 | 36 | 11.43 | 20.28 | 20.62 | 6.64 |
| Maple+rx | 52 | 7 | 4.8 | 13.02 | 11.69 | 2.81 |
| Maple+rx+rp | 56 | 13 | 4.84 | 15.21 | 8.7 | 3.04 |
| Maple+rx+rp+cf | 51 | 18 | 6.52 | 12.53 | 15.62 | 2.94 |
| | | | SC2019(400) | | | |
| glucose+rx | 14 | 8 | 26.46 | 10.79 | 17.42 | 6.39 |
| glucose+rx+rp | 10 | 26 | 22.68 | 8.67 | 14.59 | 5.14 |
| glucose+rx+rp+cf | 11 | 26 | 20.39 | 11.82 | 15.51 | 5.95 |
| Maple+rx | 14 | 7 | 12.66 | 2.67 | 12.94 | 1.98 |
| Maple+rx+rp | 9 | 14 | 8.6 | 3.17 | 16.59 | 2.53 |
| Maple+rx+rp+cf | 12 | 15 | 11.21 | 3.05 | 17.23 | 2.22 |
| | | | SC2020(400) | | | |
| glucose+rx | 30 | 9 | 14.94 | 11.75 | 14.67 | 10.27 |
| glucose+rx+rp | 23 | 37 | 13.17 | 10.79 | 9.4 | 9.71 |
| glucose+rx+rp+cf | 23 | 37 | 12.78 | 11.67 | 10.52 | 10.28 |
| Maple+rx | 19 | 13 | 14.21 | 6.69 | 10.24 | 5.25 |
| Maple+rx+rp | 30 | 29 | 8.53 | 6.62 | 11.7 | 6.18 |
| Maple+rx+rp+cf | 23 | 36 | 10.95 | 6.05 | 14.17 | 5.42 |

# Experiments – Results on FCC benchmark

**Table 3.** Comparing with state-of-the-art solvers on FCC. glucose+ is short for glucose+rx+rp+cf, and malple+ is short for Maple-DL+rx+rp+cf.

| Benchmark | glucose | glucose+ | Maple | Maple+ | kissat_sat | kissat_sat+cf | CCAnr |
|---|---|---|---|---|---|---|---|
| | #SAT | #SAT | #SAT | #SAT | #SAT | #SAT | #SAT |
| | #UNSAT | #UNSAT | #UNSAT | #UNSAT | #UNSAT | #UNSAT | #UNSAT |
| | #Solved | #Solved | #Solved | #Solved | #Solved | #Solved | #Solved |
| | PAR2 | PAR2 | PAR2 | PAR2 | PAR2 | PAR2 | PAR2 |
| FCC (10000) | 7330 | 8075 | 8084 | 8759 | 8192 | 8214 | 7853 |
| | 187 | 197 | 215 | 218 | 207 | 211 | 0 |
| | 7517 | 8272 | 8299 | 8977 | 8399 | 8425 | 7853 |
| | 2555.85 | 1850.58 | 1867.13 | 1243.66 | 1760.55 | 1734.61 | 2215.35 |

# Conclusion

As far as we know, this is the first work that meets the standard of the challenge 7 *"Demonstrate the successful combination of stochastic search and systematic search techniques, by the creation of a new algorithm that outperforms the best previous examples of both approaches."* on standard application benchmarks.

# Thank you!
# Any question?