УНИВЕРСИТЕТ ИТМО

Задача максимальной выполнимости (MaxSAT). Ядра невыполнимости.

Чухарев Константин // Университет ИТМО

Санкт-Петербург, 2020

Сириус
Образовательный центр

ITMO UNIVERSITY
Saint Petersburg, Russia

Maximum Satisfiability

Chukharev Konstantin // ITMO University

This logo should also
be in English...

Сириус
Образовательный центр

Saint Petersburg, 2020

Definitions (again)

Assignment: $\nu : X \to \{0, u, 1\} \qquad 0 < u < 1$

Clause (of CNF) is:

satisfied, if *at least one* literal is assigned 1,
unsatisfied, if *all* its literals are assigned 0,
unresolved, otherwise.

In case you forgot what clause looks like:

$$x \vee y \vee z$$

CNF formula φ is:

satisfied, if *all* of its clauses are satisfied,
unsatisfied, if *at least one* clause is unsatisfied,
unresolved, otherwise.

SAT problem

MaxSAT Problem

Given: a set of clauses F (CNF formula)

Task: find ν, s.t. $\sum_{C \in F} \nu(C) \to \max$

Find an assignment that satisfies a maximum number of clauses.

Flavours of MaxSAT

- Partial MaxSAT
  - Clauses may be “hard” (infinite weight) and “soft” (finite weight)
  - All hard clauses must be satisfied

- Weighted MaxSAT
  - Each clause C has an associated weight w_C
  - $\sum\limits_{C \in F} w_C \nu(C) \to \max$
- Weighted Partial MaxSAT
  - “Partial”: hard clauses
  - “Weighted”: soft clauses with weights

Algorithms for MaxSAT

- ↳ Convert to PBO (Pseudo-Boolean Optimization)
- ↳ Convert to IP (Integer Programming)
- ↳ Branch-and-Bound
- ↳ Unsatisfiability-based (“core-guided”)
- ↳ Hybrid
- ↳ [[many more, discovered annually]]

MaxSAT $\rightsquigarrow$ PBO [1/3]

- Clause:
  - $L_1 \vee \dots \vee L_n \iff L_1 + \dots + L_n \geq 1$
- Cardinality constraints:
  - $\text{AtLeastK}(\{L_1, \dots, L_n\}, k) \iff L_1 + \dots + L_n \geq k$
- Pseudo-Boolean constraints:
  - $\sum_i w_i L_i \geq k$
  - Example: $2L_1 + 3L_2 - L_3 \geq 2$

PBO? [2/3]

- Pseudo-Boolean Satisfaction (PBS)
  - ↳ SAT for pseudo-boolean constraints
- Pseudo-Boolean Optimization
  - ↳ PBS + objective function
  - ↳ 0-1 integer programming (ILP)

MaxSAT $\rightsquigarrow$ PBO [3/3]

MaxSAT:

$$\left.\begin{array}{l} x_1 \lor \neg x_2 \lor x_4 \\ \neg x_1 \lor \neg x_2 \lor x_3 \end{array}\right\} \text{hard clauses}$$

$$\left.\begin{array}{l} [8]\ \neg x_2 \lor \neg x_4 \\ [4]\ \neg x_3 \lor x_2 \\ [3]\ x_1 \lor x_3 \\ [2]\ x_6 \end{array}\right\} \text{soft clauses [with weights]}$$

0-1 ILP:

Minimize:

$$8r_3 + 4r_4 + 3r_5 + 2\neg x_6$$

Subject to:

$$x_1 + \neg x_2 + x_4 \geq 1$$

$$\neg x_1 + \neg x_2 + x_3 \geq 1$$

$$r_3 + \neg x_2 + \neg x_4 \geq 1$$

$$r_4 + \neg x_3 + x_2 \geq 1$$

$$r_5 + x_1 + x_3 \geq 1$$

relaxation variables

Algorithms for PBS/PBO

- PBS
  - Extended SAT-solvers, e.g. MiniSat+
- PBO
  - SAT-based approach
  - Branch-and-Bound
  - UNSAT-based approach

SAT-based approach

$\nu \leftarrow \emptyset$

$\mathrm{UB} \leftarrow \infty$

while true:

  if P is SAT:

    $\nu \leftarrow \mathrm{getAssignment}()$

    $\mathrm{UB} \leftarrow \sum_{i=1}^{n} w_i \nu(C_i)$

    $P \leftarrow P \wedge \left(\sum_{i=1}^{n} w_i \nu(C_i) < \mathrm{UB} \right)$

  else:

    return (ν, UB)

Iterative search for optimal UB:

- Linear search (simplest, least effective)
- Binary search?
- SAT to UNSAT (good in some cases)

Branch-and-Bound

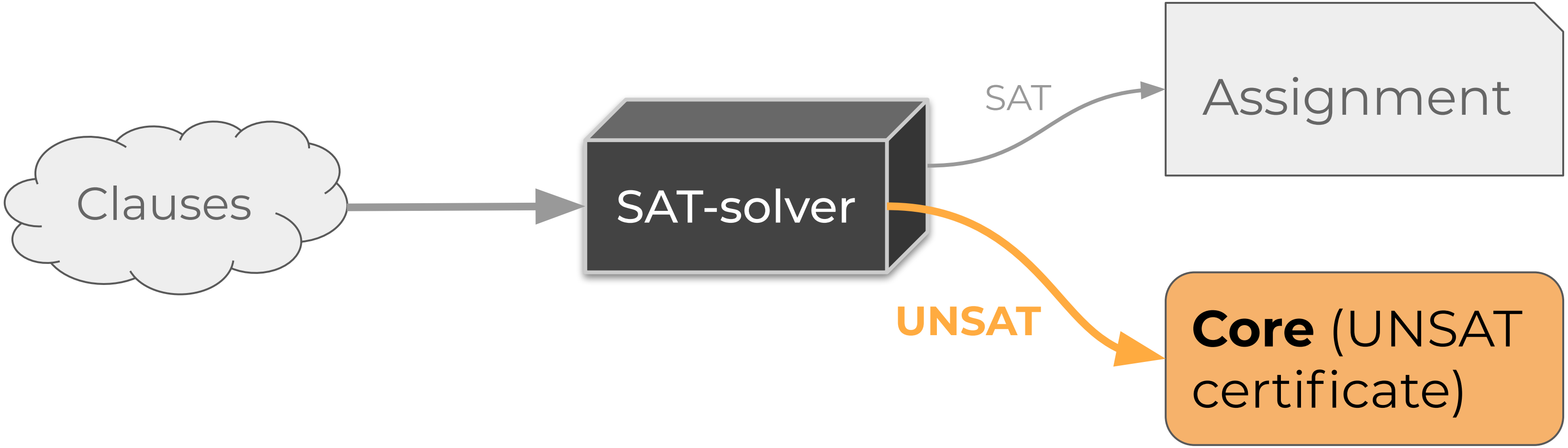
Not today

UNSAT-based approach

↳ Also called “core-guided”.

… Core?

(Not) Solving via SAT Solvers



Unsat Core, MUS, MCS, HS and other scary terms

- **MUS** – Minimal Unsatisfiable Subset $\mathcal{M} \models \bot \wedge \forall_{\mathcal{M}' \subset \mathcal{M}} \mathcal{M}' \models \top$

$$\underline{(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2)} \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

MUS = Core

- **MCS** – Minimal Correction Subset $\mathcal{F} \setminus \mathcal{C} \models \top \wedge \forall_{\mathcal{C}' \subset \mathcal{C}} \mathcal{F} \setminus \mathcal{C}' \models \bot$

$$\underline{(\neg x_1 \vee \neg x_2)} \wedge (x_1) \wedge (x_2) \wedge \underline{(\neg x_3 \vee \neg x_4)} \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- Every **MUS** is a *minimal hitting set* of a set of all **MCS**es and vice-versa.

Duality

Efficiently extract the smallest MUSes

Hitting Set

Given a collection of sets Σ, the (*minimum*) **hitting set** is the (smallest) set *S* which intersects (hits) every set in Σ.

Equivalent to the (*minimum*) **vertex cover** problem for graphs.

$$\begin{aligned}\Sigma = \{&\{1,\underline{2}\},\{1,\underline{5}\},\{\underline{2},3\},\{\underline{2},\underline{6}\},\{\underline{2},7\},\{3,\underline{4}\},\{3,\underline{6}\},\\ &\{\underline{4},10\},\{\underline{4},\underline{9}\},\{\underline{5},7\},\{\underline{6},8\},\{\underline{6},\underline{9}\},\{7,8\},\{8,\underline{9}\}\}\end{aligned}$$

$$S = \{2,4,5,6,9\}$$

UNSAT-based (core-guided) algorithm

1. Convert all SOFT clauses to HARD
2. Try to solve: if SAT, goto 5
3. Relax unsatisfiable subset (core)
4. Repeat (goto 2)
5. First SAT is the optimal solution, done!

$\phi_w \leftarrow \phi$

while ϕ_w is UNSAT:

$\quad \phi_C \leftarrow \text{getCore}(\phi_w)$

$\quad V_R \leftarrow \emptyset$

$\quad$ **foreach** soft clause $\omega \in \phi_C$:

$\quad\quad \omega_R \leftarrow \omega \cup \{r\}$

$\quad\quad \phi_w \leftarrow (\phi_w \setminus \{\omega\}) \cup \{\omega_R\}$

$\quad\quad V_R \leftarrow V_R \cup \{r\}$

$\quad \phi_w \leftarrow \phi_w \cup \{\sum_{r \in V_R} r \leq 1\}$

See also: MSU3, Fu-Malik, WPM2, Eva, RC2

Enough theory.

Demo time

WCNF (Weighted CNF)

$$x_1 \lor \neg x_2 \lor x_4$$
$$\neg x_1 \lor \neg x_2 \lor x_3$$
$$[8]\ \neg x_2 \lor \neg x_4$$
$$[4]\ \neg x_3 \lor x_2$$
$$[3]\ x_1 \lor x_3$$
$$[2]\ x_6$$

```
p wcnf 5 6 999
999 1 -2 4 0
999 -1 -2 3 0
8 -2 -4 0
4 -3 2 0
3 1 3 0
2 5 0
```

Example: Minimum Vertex Cover

Graph 1: https://bit.ly/34cYFPf

Graph 2: https://bit.ly/3hkxLJ2

Sample solution: https://bit.ly/3hhwUsg

Thanks for your attention.

kchukharev@itmo.ru

Lipen

ITsMOre than a UNIVERSITY