Improving Unsatisfiability-based Algorithms for Boolean Optimization

Vasco Manquinho Ruben Martins Inês Lynce

IST/INESC-ID, Technical University of Lisbon, Portugal

SAT 2010, Edinburgh

Motivation

- Increasing interest in generalizations of SAT
- SAT techniques extended for MaxSAT, PBO and WBO
- Unsatisfiability-based algorithms have been proposed for Boolean Optimization problems
  - very effective for several classes of instances
  - can perform poorly on instances that are easy for classical approaches
- Integration of procedures in a unique Boolean optimization framework

Outline

- Background
  - MaxSAT, PBO and WBO
- Algorithmic Solutions
  - Classical Approaches
  - Unsatisfiability-based approaches
- Improving Unsatisfiability-based algorithms
  - PBO as preprocessing
  - Constraint Branching
- Experimental Results
- Conclusions

Maximum Satisfiability (MaxSAT)

MaxSAT Problem

Given a CNF formula φ, find an assignment to problem variables that maximizes the number of satisfied clauses in φ (or minimizes the number of unsatisfied clauses).

Partial MaxSAT Problem

Given a conjunction of two CNF formulas φ_h and φ_s, find an assignment to problem variables that satisfies all hard clauses (φ_h) and maximizes the number of satisfied soft clauses (φ_s).

Maximum Satisfiability (MaxSAT)

Weighted CNF Formula

- set of weighted clauses
- weighted clause: pair (ω, c) where ω is a clause and $c \in \mathbb{N}$ is a positive cost of unsatisfying ω

Weighted MaxSAT Problem

Given a weighted CNF formula $\varphi_{s,c}$, find an assignment to problem variables that minimizes the total cost of unsatisfied clauses.

Weighted Partial MaxSAT Problem

Given a weighted CNF formula $\varphi_{s,c}$ and a classical CNF formula φ_h, find an assignment to problem variables that satisfies all hard clauses (φ_h) and minimizes the total cost of unsatisfied soft clauses in $\varphi_{s,c}$.

Pseudo-Boolean Optimization (PBO)

Pseudo-Boolean Optimization

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{j=1}^{n} c_j \cdot x_j \\
\text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij} \cdot l_j \geq b_i, \\
& l_j \in \{x_j, \overline{x}_j\}, x_j \in \{0, 1\}, \\
& a_{ij}, b_i, c_j \in \mathbb{N}_0^+
\end{array}
$$

Weighted Boolean Optimization (WBO)

WBO Formula

Weighted Boolean Optimization formula is composed of two pseudo-Boolean constraint sets (φ_h, φ_s):

- φ_h: set of hard pseudo-Boolean constraints
- φ_s: set of soft weighted pseudo-Boolean constraints
- Soft pseudo-Boolean constraint (ω, c):
  - ω: pseudo-Boolean constraint
  - there is an integer weight c representing the cost of not satisfying ω

WBO Problem

Given a WBO formula, find an assignment to problem variables that satisfies all hard constraints (φ_h) and minimizes the total cost of unsatisfied soft constraints (φ_s).

WBO (Example)

Weighted Boolean Optimization instance

$$\begin{aligned}\varphi_h &= \{x_1 + x_2 + x_3 \geq 2, \quad 2\overline{x}_1 + \overline{x}_2 + x_3 \geq 2\} \\ \varphi_s &= \{(x_1 + \overline{x}_2 \geq 1, 2), \quad (\overline{x}_1 + \overline{x}_3 \geq 1, 3)\}\end{aligned}$$

- Assignments that satisfy all hard constraints:

(1) $x_1 = x_3 = 1$; $x_2 = 0$; $\sum c_i = 3$
(2) $x_1 = 0$; $x_2 = x_3 = 1$; $\sum c_i = 2$ (solution)

Encode MaxSAT as WBO

- For each hard clause $(l_1 \vee l_2 \vee \cdots \vee l_k)$
  - define a hard PB constraint as $l_1 + l_2 + \cdots + l_k \geq 1$
- For each weighted soft clause (ω, c) where $\omega = (l_1 \vee l_2 \vee \cdots \vee l_k)$
  - define a soft PB constraint as $l_1 + l_2 + \cdots + l_k \geq 1$ with weight c

Encode MaxSAT as WBO (Example)

Weighted Partial MaxSAT instance

$$\begin{aligned} \varphi_h &= \{x_1 \vee x_2 \vee \bar{x}_3, \quad \bar{x}_2 \vee x_3, \quad \bar{x}_1 \vee x_3\} \\ \varphi_s &= \{(\bar{x}_3, 5), \quad (x_1 \vee x_2, 3), \quad (x_1 \vee x_3, 2)\} \end{aligned}$$

Corresponding WBO instance

$$\begin{aligned} \varphi_h &= \{x_1 + x_2 + \bar{x}_3 \geq 1, \quad \bar{x}_2 + x_3 \geq 1, \quad \bar{x}_1 + x_3 \geq 1\} \\ \varphi_s &= \{(\bar{x}_3 \geq 1, 5), \quad (x_1 + x_2 \geq 1, 3), \quad (x_1 + x_3 \geq 1, 2)\} \end{aligned}$$

Encode PBO as WBO

- For each pseudo-Boolean constraint $\sum_{j=1}^{n} a_{ij} l_j \geq b_i$
  - add this PB constraint to the set of hard PB constraints
- For each term $c_j \cdot x_j$ in the objective function
  - add a weighted soft PB constraint of the form $((\overline{x}_j \geq 1), c_j)$

Encode PBO as WBO (Example)

Pseudo-Boolean Optimization instance

$$\begin{aligned} \text{minimize} \quad & 4x_1 + 2x_2 + x_3 \\ \text{subject to} \quad & 2x_1 + 3x_2 + 5x_3 \geq 5 \\ & \bar{x}_1 + \bar{x}_2 \geq 1 \\ & x_1 + x_2 + x_3 \geq 2 \end{aligned}$$

Corresponding WBO instance

$$\begin{aligned} \varphi_h &= \{2x_1 + 3x_2 + 5x_3 \geq 5, \quad \bar{x}_1 + \bar{x}_2 \geq 1, \quad x_1 + x_2 + x_3 \geq 2\} \\ \varphi_s &= \{(\bar{x}_1 \geq 1, 4), \quad (\bar{x}_2 \geq 1, 2), \quad (\bar{x}_3 \geq 1, 1)\} \end{aligned}$$

Algorithmic Solutions (Classical Approaches)

- Branch and bound:

  e.g. `MaxSatz`, `MiniMaxSAT`

- Iteration of the upper bound:

  e.g. `Pueblo`, `minisat+`

- Conversions from one Boolean formalism to another:

  e.g. `minisat+`, `SAT4J MS`

Unsatisfiability-based MaxSAT

Original algorithm proposed by Fu&Malik [SAT 2006]:

(1) Identify unsatisfiable sub-formula of an UNSAT formula
  - SAT solver able to generate an UNSAT core

(2) For each unsatisfiable sub-formula φ_C:
  - Relax all soft clauses in φ_C by adding a new relaxation variable to each clause
  - Add a new constraint such that at most 1 relaxation variable is assigned value 1

(3) When the resulting CNF formula is SAT, the solver terminates

(4) Otherwise, go back to 1

Unsatisfiability-based MaxSAT

$\varphi_W \leftarrow \varphi$
while (φ_W is UNSAT)
  do Let φ_C be an unsatisfiable sub-formula of φ_W
    $V_R \leftarrow \emptyset$
    for each soft clause $\omega \in \varphi_C$
      do $\omega_R \leftarrow \omega \cup \{r\}$
        $\varphi_W \leftarrow \varphi_W - \{\omega\} \cup \{\omega_R\}$
        $V_R \leftarrow V_R \cup \{r\}$
    $\varphi_R \leftarrow \text{CNF}(\sum_{r \in V_R} r = 1)$ ▷ Equals1 constraint
    $\varphi_W \leftarrow \varphi_W \cup \varphi_R$ ▷ Clauses in φ_R are declared hard
return $|\varphi| -$ number of relaxation variables assigned to 1

Unsatisfiability-based Weighted MaxSAT

$\varphi_W \leftarrow \varphi$
$cost_{lb} \leftarrow 0$
while (φ_W is UNSAT)
  do Let φ_C be an unsatisfiable sub-formula of φ_W
    $min_c \leftarrow \min_{\omega \in \varphi_C \wedge \neg hard(\omega)} cost(\omega)$
    $cost_{lb} \leftarrow cost_{lb} + min_c$
    $V_R \leftarrow \emptyset$
    for each soft clause $\omega \in \varphi_C$
      do $\omega_R \leftarrow \omega \cup \{r\}$
        $cost(\omega_R) \leftarrow min_c$
        if $cost(\omega) > min_c$
          then $\varphi_W \leftarrow \varphi_W \cup \{\omega_R\}$
            $cost(\omega) \leftarrow cost(\omega) - min_c$
          else $\varphi_W \leftarrow \varphi_W - \{\omega\} \cup \{\omega_R\}$
        $V_R \leftarrow V_R \cup \{r\}$
    $\varphi_W \leftarrow \varphi_W \cup \mathrm{CNF}(\sum_{r \in V_R} r = 1)$
return $cost_{lb}$

Unsatisfiability-based Weighted MaxSAT

Weighted MaxSAT instance

$$\begin{aligned}\varphi_h &= \{x_1 \vee x_2 \vee \overline{x}_3, \quad \overline{x}_2 \vee x_3, \quad \overline{x}_1 \vee x_3\} \\ \varphi_s &= \{(\overline{x}_3, 5), \quad (x_1 \vee x_2, 3), \quad (x_1 \vee x_3, 2)\}\end{aligned}$$

Unsatisfiable sub-formula:

$$\varphi_C = \{\overline{x}_2 \vee x_3, \quad \overline{x}_1 \vee x_3, \quad (\overline{x}_3, 5), \quad (x_1 \vee x_2, 3)\}$$

- $min_C = 3$
- Relax $(x_1 \vee x_2, 3)$ to $(r_1 \vee x_1 \vee x_2, 3)$
- Split $(\overline{x}_3, 5)$ into $(\overline{x}_3, 2)$ and $(r_2 \vee \overline{x}_3, 3)$
- Add $\mathrm{CNF}(r_1 + r_2 = 1)$ to φ_h

Unsatisfiability-based Weighted MaxSAT

Weighted MaxSAT instance

$$\begin{aligned}\varphi_h &= \{x_1 \vee x_2 \vee \overline{x}_3, \quad \overline{x}_2 \vee x_3, \quad \overline{x}_1 \vee x_3\} \\ \varphi_s &= \{(\overline{x}_3, 5), \quad (x_1 \vee x_2, 3), \quad (x_1 \vee x_3, 2)\}\end{aligned}$$

Results in a new formula:

$$\begin{aligned}\varphi_h &= \{x_1 \vee x_2 \vee \overline{x}_3, \quad \overline{x}_2 \vee x_3, \quad \overline{x}_1 \vee x_3, \quad \text{CNF}(r_1 + r_2 = 1)\} \\ \varphi_s &= \{(\overline{x}_3, 2), \quad (r_2 \vee \overline{x}_3, 3), \quad (r_1 \vee x_1 \vee x_2, 3), \quad (x_1 \vee x_3, 2)\}\end{aligned}$$

Algorithm for Weighted Boolean Optimization

- Follows the same approach as Unsatisfiability-Based Weighted MaxSAT algorithm
- Instead of SAT solver, uses Pseudo-Boolean solver enhanced with unsatisfiable sub-formula extraction
- Relaxation of pseudo-Boolean constraints $\sum a_j l_j \geq b$
  - $b \cdot r + \sum a_j l_j \geq b$
- No need to encode constraint $\sum_{r \in V_R} r = 1$ into CNF

Improving Unsatisfiability-based Algorithms

- Unsatisfiability-based algorithms search on the lower bound. Sometimes is better to search on the upper bound:
  (1) PBO as Preprocessing
- The number of relaxation variables grows significantly at each step:
  (2) Constraint Branching

Encode WBO as PBO

- For each hard PB constraint $\sum_{j=1}^{n} a_{ij} l_j \geq b_i$
  - add this PB constraint to the set of constraints
- For each weighted soft PB constraint $\sum_{j=1}^{n} a_{ij} l_j \geq b_i$ with cost c_j
  - define a PB constraint with a new relaxation variable r
    $b_i r + \sum_{j=1}^{n} a_{ij} l_j \geq b_i$
  - add $c_j \cdot r$ to the objective function

Encode WBO as PBO (Example)

Weighted Boolean Optimization instance

$$\begin{aligned} \varphi_h &= \{x_1 + x_2 + x_3 \geq 2, \quad 2\overline{x}_1 + \overline{x}_2 + x_3 \geq 2, \quad x_1 + x_4 \geq 1\} \\ \varphi_s &= \{(x_1 + \overline{x}_2 \geq 1, 2), \quad (\overline{x}_1 + \overline{x}_3 \geq 1, 3), \quad (\overline{x}_4 \geq 1, 4)\} \end{aligned}$$

Corresponding PBO instance

$$\begin{aligned} \text{minimize} \quad & 2r_1 + 3r_2 + 4r_3 \\ \text{subject to} \quad & x_1 + x_2 + x_3 \geq 2 \\ & 2\overline{x}_1 + \overline{x}_2 + x_3 \geq 2 \\ & x_1 + x_4 \geq 1 \\ & r_1 + x_1 + \overline{x}_2 \geq 1 \\ & r_2 + \overline{x}_1 + \overline{x}_3 \geq 1 \\ & r_3 + \overline{x}_4 \geq 1 \end{aligned}$$

PBO as Preprocessing

(1) Simplification techniques are used in the PBO formula:

- a generalization of `Hypre` for PB formulas is used

(2) The PBO formula is solved using tight limits:

- PB solver is used for 10% of the time limit
- If optimality is not proved, the formula is translated back to WBO
- Small learnt clauses are kept in the WBO formula as hard clauses

Using Constraint Branching

- Consider the following Equals1 constraint: $\sum_{i=1}^{k} r_i = 1$:
  - If r_i is assigned to 1, all other variables $r_j \neq r_i$ must be 0
  - However, if r_i is assigned to 0, no propagation occurs
- Assigning value 1 to any of these variables produces very different search trees

Using Constraint Branching

- Constraint Branching:
  - Instead of assigning one variables, half of the variables are assigned:
    $\omega_{c1} : \sum_{i=1}^{k/2} r_i = 0$
  - If $\varphi \cup \{\omega_{c1}\}$ is unsatisfiable then:
    - $\exists_i r_i = 1$, with $1 \leq i \leq \frac{k}{2}$
    - we can infer $\omega_{c2} : \sum_{i=k/2+1}^{k} r_i = 0$

Computing Cores with Constraint Branching

$\textsc{compute\_core}(\varphi)$

if (no large Equals1 constraint exist in φ)
    then $(\mathsf{st}, \varphi_C) \leftarrow \mathrm{PB}(\varphi)$
        return (st, φ_C)
    else Select a large Equals1 constraint ω from φ
        $k = \mathsf{size}(\omega)$
        $\omega_{c1} : \sum_{i=1}^{k/2} r_i = 0$
        $(\mathsf{st}, \varphi_{C1}) \leftarrow \textsc{compute\_core}(\varphi \cup \{\omega_{c1}\})$
        if $(\mathsf{st} = \textbf{SAT} \vee \omega_{c1} \notin \varphi_{C1})$
            then return $(\mathsf{st}, \varphi_{C1})$
            else $\omega_{c2} : \sum_{i=k/2+1}^{k} r_i = 0$
                $(\mathsf{st}, \varphi_{C2}) \leftarrow \textsc{compute\_core}(\varphi \cup \{\omega_{c2}\})$
                if $(\mathsf{st} = \textbf{SAT} \vee \omega_{c2} \notin \varphi_{C2})$
                    then return $(\mathsf{st}, \varphi_{C2})$
                    else return $(\mathsf{st}, \varphi_{C1} \cup \varphi_{C2})$

Experimental Results

- Industrial benchmark sets of the partial MaxSAT problem
- The most effective MaxSAT solvers from the MaxSAT evaluation of 2009 were considered: `MSUncore`, `SAT4J (MS)`, `pm2`
- Timeout: 1800 seconds
- Intel Xeon 5160 server with 3GB RAM

Experimental Results

- Solved Instances for Industrial Partial MaxSAT:

Benchmark set	#I	MSUncore	SAT4J (MS)	pm2	wbo1.0	wbo1.2
bcp-fir	59	49	10	58	40	47
bcp-hipp-yRa1	176	139	140	166	144	137
bcp-msp	148	121	95	93	26	95
bcp-mtg	215	173	196	215	181	207
bcp-syn	74	32	21	39	34	33
CircuitTraceCompaction	4	0	4	4	0	4
HaplotypeAssembly	6	5	0	5	5	5
pbo-mqc	256	119	250	217	131	210
pbo-routing	15	15	13	15	15	15
PROTEIN_INS	12	0	2	3	1	2
Total	965	553	731	815	577	755

Conclusions

- PBO solvers can be used as a preprocessing step such that:
  1) inference preprocessing techniques are used;
  2a) some problems are easily solved with a search on the upper bound;
  2b) restrict the search space by learning hard constraints.
- Constraint branching can improve the effectiveness of the solver
- Experimental results show that these techniques significantly improve the performance of `wbo`
- These results provide a strong stimulus for further integration of other Boolean optimization techniques