On Solving MaxSAT Through SAT

Carlos Ansótegui
DIEI, UdL, Lleida, Spain

María Luisa Bonet
LSI, UPC, Barcelona, Spain

Jordi Levy
IIIA, CSIC, Barcelona, Spain

POS'10, Edinburgh

Partial MaxSat

$$\left.\begin{array}{l} C_1 \\ \ldots \\ \ldots \\ \ldots \\ C_m \end{array}\right\} \textit{Soft Clauses}$$

$$\left.\begin{array}{l} C_{m+1} \\ \ldots \\ \ldots \\ \ldots \\ C_{m+m'} \end{array}\right\} \textit{Hard Clauses}$$

Partial MaxSat is the problem of finding an assignment such that
no **hard** clause is falsified and
the minimum number of **soft** clauses are falsified.

Weighted Partial MaxSat

$$\left.\begin{array}{rl} w_1 : & C_1 \\ & \dots \\ & \dots \\ & \dots \\ w_m : & C_m \end{array}\right\} \textit{Soft Clauses}$$

$$\left.\begin{array}{rl} \infty : & C_{m+1} \\ & \dots \\ & \dots \\ & \dots \\ \infty : & C_{m+m'} \end{array}\right\} \textit{Hard Clauses}$$

Weighted Partial MaxSat is the problem of finding an assignment that minimizes the cost of the falsified clauses.

Solving Partial MaxSat Through Satisfiability Testing

Basic approach:

$$\varphi_k : \quad \left.\begin{array}{l} C_1 \vee b_1 \\ \ldots \\ \ldots \\ \ldots \\ C_m \vee b_m \end{array}\right\} \textit{Soft Clauses}$$

$$\left.\begin{array}{l} C_{m+1} \\ \ldots \\ \ldots \\ \ldots \\ C_{m+m'} \\ CNF(\sum_{i=1}^{m} b_i \leq k) \end{array}\right\} \textit{Hard Clauses}$$

if φ_{k-1} is unsatisfiable and φ_k satisfiable, then k is the optimum.

MaxSat Solvers Based On Satisfiability Testing

Solvers at MaxSat Evaluations 2008 and 2009:

- Partial and Weighted Partial MaxSat:
  - **SAT4Java Maxsat** [D. L. Berre]
  - **Msu1.2** (implementation of FU&MALIK algorithm) [J. Marques-Silva, V. Manquinho and J. Planes]
  - **Msu4.0** [J. Marques-Silva and J. Planes]
  - **MsUnCore** (implementation of FU&MALIK algorithm) [J. Marques-Silva, V. Manquinho and J. Planes]
  - **WBO** (uses a pseudo-Boolean solver) [V. Manquinho, J. Marques-Silva and J. Planes]

Our solvers and technical contributions:

- A Partial MaxSat algorithm, **PM2**, and the proof of its correctness
- A Weighted version of the FU&MALIK algorithm, **WPM1** together with its proof of correctness
- A weighted version of **PM2**, **WPM2**, and its proof of correctness.

The PM2 algorithm

input: $\varphi = \{C_1, \ldots, C_m, C_{m+1}, \ldots, C_{m+m'}\}$

$\varphi_w := \{C_1 \vee b_1, \ldots, C_m \vee b_m, C_{m+1}, \ldots, C_{m+m'}\}$ — Protect all soft clauses

$cost := 0$ — Optimal

$L := \emptyset$ — Set of Cores

while *true* **do**

  $(st, \varphi_c) := SAT(\varphi_w \cup CNF(\sum_{i=1}^{m} b_i \leq cost))$ — Call to SAT solver with at most cardinality

  if $st = SAT$ **then return** $cost$

  remove the hard clauses from φ_c

  if $\varphi_c = \emptyset$ **then return** UNSAT

  $B := \emptyset$ — Blocking variables of the core

  for each $C_i \vee b_i \in \varphi_c$ **do**

    $B := B \cup \{b_i\}$

  $L := L \cup \{B\}$

  $k := |\{\psi \in L \mid \psi \subseteq B\}|$ — Num. of cores contained in B

  $\varphi_w := \varphi_w \cup CNF(\sum_{b \in B} b \geq\ )$ — Add at least cardinality constraint

  $cost := cost + 1$

The PM2 algorithm

input: $\varphi = \{C_1, \ldots, C_m, C_{m+1}, \ldots, C_{m+m'}\}$
$\varphi_w := \{C_1 \vee b_1, \ldots, C_m \vee b_m, C_{m+1}, \ldots, C_{m+m'}\}$ — Protect all soft clauses
$cost := 0$ — Optimal
$L := \emptyset$ — Set of Cores

while *true* **do**

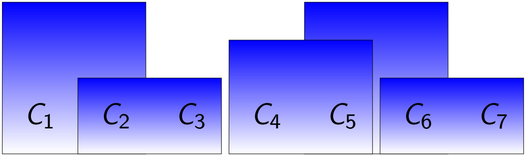
$(st, \varphi_c) := SAT(\varphi_w \cup CNF(\sum_{i=1}^{m} b_i \leq cost))$ — Call to SAT solver with at most cardinality

if $st = SAT$ **then return** $cost$
remove the hard clauses from φ_c
if $\varphi_c = \emptyset$ **then return** UNSAT
$B := \emptyset$ — Blocking variables of the core
for each $C_i \vee b_i \in \varphi_c$ **do**
$B := B \cup \{b_i\}$
$L := L \cup \{B\}$
$k := |\{\psi \in L \mid \psi \subseteq B\}|$ — Num. of cores contained in B
$\varphi_w := \varphi_w \cup CNF(\sum_{b \in B} b \geq 1)$ — Add at least cardinality constraint
$cost := cost + 1$

The PM2 algorithm

input: $\varphi = \{C_1, \ldots, C_m, C_{m+1}, \ldots, C_{m+m'}\}$
$\varphi_w := \{C_1 \vee b_1, \ldots, C_m \vee b_m, C_{m+1}, \ldots, C_{m+m'}\}$ Protect all soft clauses
$cost := 0$ Optimal
$L := \emptyset$ Set of Cores
while *true* **do**
$(st, \varphi_c) := SAT(\varphi_w \cup CNF(\sum_{i=1}^{m} b_i \leq cost))$ Call to SAT solver with at most cardinality
if $st = SAT$ **then return** *cost*
remove the hard clauses from φ_c
if $\varphi_c = \emptyset$ **then return** UNSAT
$B := \emptyset$ Blocking variables of the core
for each $C_i \vee b_i \in \varphi_c$ **do**
$B := B \cup \{b_i\}$
$L := L \cup \{B\}$
$k := |\{\psi \in L \mid \psi \subseteq B\}|$ Num. of cores contained in B
$\varphi_w := \varphi_w \cup CNF(\sum_{b \in B} b \geq k)$ Add at least cardinality constraint
$cost := cost + 1$

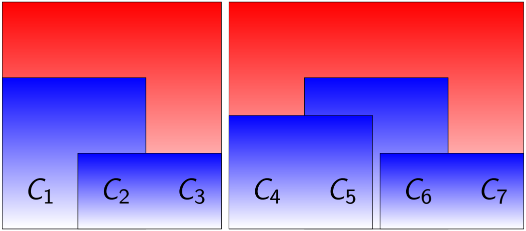
Covers



Definition (covers)

Given a set of cores L, we say that B_2 is a *cover* of L,
if it is a **nonempty minimal set** such that,
for every $B_1 \in L$, if $B_1 \cap B_2 \neq \emptyset$, then $B_1 \subseteq B_2$

Covers



Definition (covers)

Given a set of cores L, we say that B_2 is a *cover* of L, if it is a **nonempty minimal set** such that, for every $B_1 \in L$, if $B_1 \cap B_2 \neq \emptyset$, then $B_1 \subseteq B_2$

input: $\varphi = \{C_1, \ldots, C_m, C_{m+1}, \ldots, C_{m+m'}\}$

$\varphi_w := \{C_1 \vee b_1, \ldots, C_m \vee b_m, C_{m+1}, \ldots, C_{m+m'}\}$ Protect all soft clauses

$cost := 0$ Optimal

$L := \emptyset;\ AL := \emptyset;$ Set of cores and atleast

$AM := \{b_1 \leq 0, \ldots, b_m \leq 0\}$ Set of atmosts

while *true* **do**

  $(st, \varphi_c) := SAT(\varphi_w \cup AL \cup AM)$ Call to the SAT solver

  if $st = SAT$ **then return** *cost*

  remove the hard clauses from φ_c

  if $\varphi_c = \emptyset$ **then return** UNSAT

  $B := \{b_i \mid C_i \vee b_i \in \varphi_c\}$

  $L := L \cup \{B\}$

  $k := |\{B' \in L \mid B' \subseteq B\}|$ Num. of cores contained in A

  $AL := AL \cup CNF(\sum_{b \in B} b \geq k)$ Add new at-least cardinality constraint

  $AM := \emptyset$ Recalculate in each iteration

  for each $B \in coversof(L)$ **do**

    $k := |\{B' \in L \mid B' \subseteq B\}|$ Num. of cores contained in the cover B

    $AM := AM \cup CNF(\sum_{b \in B} b \leq k)$ Add new at-most cardinality constraint

  $cost := cost + 1$

Example: PM2.1

1:	$x_1 \vee b_1$
1:	$x_2 \vee b_2$
1:	$x_3 \vee b_3$
1:	$x_4 \vee b_4$
∞:	$\overline{x_1} \vee \overline{x_2}$
∞:	$\overline{x_1} \vee \overline{x_3}$
∞:	$\overline{x_1} \vee \overline{x_4}$
∞:	$\overline{x_2} \vee \overline{x_3}$
∞:	$\overline{x_2} \vee \overline{x_4}$
∞:	$\overline{x_3} \vee \overline{x_4}$
∞:	$b_1 \leq 0$
∞:	$b_2 \leq 0$
∞:	$b_3 \leq 0$
∞:	$b_4 \leq 0$

Example: PM2.1

1:	$x_1 \vee b_1$
1:	$x_2 \vee b_2$
1:	$x_3 \vee b_3$
1:	$x_4 \vee b_4$
∞:	$\overline{x_1} \vee \overline{x_2}$
∞:	$\overline{x_1} \vee \overline{x_3}$
∞:	$\overline{x_1} \vee \overline{x_4}$
∞:	$\overline{x_2} \vee \overline{x_3}$
∞:	$\overline{x_2} \vee \overline{x_4}$
∞:	$\overline{x_3} \vee \overline{x_4}$
∞:	$b_1 \leq 0$
∞:	$b_2 \leq 0$
∞:	$b_3 \leq 0$
∞:	$b_4 \leq 0$

1:	$x_1 \vee b_1$
1:	$x_2 \vee b_2$
1:	$x_3 \vee b_3$
1:	$x_4 \vee b_4$
∞:	$\overline{x_1} \vee \overline{x_2}$
∞:	$\overline{x_1} \vee \overline{x_3}$
∞:	$\overline{x_1} \vee \overline{x_4}$
∞:	$\overline{x_2} \vee \overline{x_3}$
∞:	$\overline{x_2} \vee \overline{x_4}$
∞:	$\overline{x_3} \vee \overline{x_4}$
∞:	$b_1 + b_2 \geq 1$
∞:	$b_1 + b_2 \leq 1$
∞:	$b_3 \leq 0$
∞:	$b_4 \leq 0$

Example: PM2.1

$1: \quad x_1 \vee b_1$
$1: \quad x_2 \vee b_2$
$1: \quad x_3 \vee b_3$
$1: \quad x_4 \vee b_4$
$\infty: \quad \overline{x_1} \vee \overline{x_2}$
$\infty: \quad \overline{x_1} \vee \overline{x_3}$
$\infty: \quad \overline{x_1} \vee \overline{x_4}$
$\infty: \quad \overline{x_2} \vee \overline{x_3}$
$\infty: \quad \overline{x_2} \vee \overline{x_4}$
$\infty: \quad \overline{x_3} \vee \overline{x_4}$
$\infty: \quad b_1 \leq 0$
$\infty: \quad b_2 \leq 0$
$\infty: \quad b_3 \leq 0$
$\infty: \quad b_4 \leq 0$

$1: \quad x_1 \vee b_1$
$1: \quad x_2 \vee b_2$
$1: \quad x_3 \vee b_3$
$1: \quad x_4 \vee b_4$
$\infty: \quad \overline{x_1} \vee \overline{x_2}$
$\infty: \quad \overline{x_1} \vee \overline{x_3}$
$\infty: \quad \overline{x_1} \vee \overline{x_4}$
$\infty: \quad \overline{x_2} \vee \overline{x_3}$
$\infty: \quad \overline{x_2} \vee \overline{x_4}$
$\infty: \quad \overline{x_3} \vee \overline{x_4}$
$\infty: \quad b_1 + b_2 \geq 1$
$\infty: \quad b_1 + b_2 \leq 1$
$\infty: \quad b_3 \leq 0$
$\infty: \quad b_4 \leq 0$

$1: \quad x_1 \vee b_1$
$1: \quad x_2 \vee b_2$
$1: \quad x_3 \vee b_3$
$1: \quad x_4 \vee b_4$
$\infty: \quad \overline{x_1} \vee \overline{x_2}$
$\infty: \quad \overline{x_1} \vee \overline{x_3}$
$\infty: \quad \overline{x_1} \vee \overline{x_4}$
$\infty: \quad \overline{x_2} \vee \overline{x_3}$
$\infty: \quad \overline{x_2} \vee \overline{x_4}$
$\infty: \quad \overline{x_3} \vee \overline{x_4}$
$\infty: \quad b_1 + b_2 \geq 1$
$\infty: \quad b_3 + b_4 \geq 1$
$\infty: \quad b_1 + b_2 \leq 1$
$\infty: \quad b_3 + b_4 \leq 1$

Example: PM2.1

1:	$x_1 \vee b_1$	1:	$x_1 \vee b_1$	1:	$x_1 \vee b_1$	1:	$x_1 \vee b_1$
1:	$x_2 \vee b_2$	1:	$x_2 \vee b_2$	1:	$x_2 \vee b_2$	1:	$x_2 \vee b_2$
1:	$x_3 \vee b_3$	1:	$x_3 \vee b_3$	1:	$x_3 \vee b_3$	1:	$x_3 \vee b_3$
1:	$x_4 \vee b_4$	1:	$x_4 \vee b_4$	1:	$x_4 \vee b_4$	1:	$x_4 \vee b_4$
∞:	$\overline{x_1} \vee \overline{x_2}$	∞:	$\overline{x_1} \vee \overline{x_2}$	∞:	$\overline{x_1} \vee \overline{x_2}$	∞:	$\overline{x_1} \vee \overline{x_2}$
∞:	$\overline{x_1} \vee \overline{x_3}$	∞:	$\overline{x_1} \vee \overline{x_3}$	∞:	$\overline{x_1} \vee \overline{x_3}$	∞:	$\overline{x_1} \vee \overline{x_3}$
∞:	$\overline{x_1} \vee \overline{x_4}$	∞:	$\overline{x_1} \vee \overline{x_4}$	∞:	$\overline{x_1} \vee \overline{x_4}$	∞:	$\overline{x_1} \vee \overline{x_4}$
∞:	$\overline{x_2} \vee \overline{x_3}$	∞:	$\overline{x_2} \vee \overline{x_3}$	∞:	$\overline{x_2} \vee \overline{x_3}$	∞:	$\overline{x_2} \vee \overline{x_3}$
∞:	$\overline{x_2} \vee \overline{x_4}$	∞:	$\overline{x_2} \vee \overline{x_4}$	∞:	$\overline{x_2} \vee \overline{x_4}$	∞:	$\overline{x_2} \vee \overline{x_4}$
∞:	$\overline{x_3} \vee \overline{x_4}$	∞:	$\overline{x_3} \vee \overline{x_4}$	∞:	$\overline{x_3} \vee \overline{x_4}$	∞:	$\overline{x_3} \vee \overline{x_4}$
∞:	$b_1 \leq 0$	∞:	$b_1 + b_2 \geq 1$	∞:	$b_1 + b_2 \geq 1$	∞:	$b_1 + b_2 \geq 1$
∞:	$b_2 \leq 0$	∞:	$b_1 + b_2 \leq 1$	∞:	$b_3 + b_4 \geq 1$	∞:	$b_3 + b_4 \geq 1$
∞:	$b_3 \leq 0$	∞:	$b_3 \leq 0$	∞:	$b_1 + b_2 \leq 1$	∞:	$b_1 + b_2 + b_3 + b_4 \geq 3$
∞:	$b_4 \leq 0$	∞:	$b_4 \leq 0$	∞:	$b_3 + b_4 \leq 1$	∞:	$b_1 + b_2 + b_3 + b_4 \leq 3$

Weighted PM2

input: $\varphi = \{(C_1, w_1), \ldots, (C_m, w_m), (C_{m+1}, \infty), \ldots, (C_{m+m'}, \infty)\}$
$\varphi^e := \{C_1 \vee b_1, \ldots, C_m \vee b_m, C_{m+1}, \ldots, C_{m+m'}\}$ Protect all soft clauses
$SC := \{\{1\}, \{2\}, \ldots, \{m\}\}$ Set of Covers
$AL := \emptyset$ Set of at least constraints
$AM := \{w_1\, b_1 \leq 0, \ldots, w_m\, b_m \leq 0\}$ Set of at most constraints
while *true* **do**
  $(st, \varphi_c) := SAT(\varphi^e \cup CNF(AL \cup AM))$ Call to the SAT solver
  if $st = SAT$ **then return** $\sum\{k' \mid \sum_{i \in B'} w_i\, b_i \leq k' \in AM\}$
  $A := \{i \mid C_i \vee b_i \in \varphi_c\}$ Indexes of blocking vars. in the core
  if $A = \emptyset$ **then return** UNSAT
  $RC := \{B \in SC \mid B \cap A \neq \emptyset\}$ Covers to be removed
  $B := \bigcup_{B' \in RC} B'$ New cover
  $k := \text{newbound}(AL, B)$ New bound
  $SC := SC \setminus RC \cup \{B\}$ New set of Covers
  $AL := AL \cup \{\sum_{i \in B} w_i\, b_i \geq k\}$ Add new at-least cardinality constra
  $AM := AM \setminus \{\sum_{i \in B'} w_i\, b_i \leq k' \mid B' \in RC\} \cup \{\sum_{i \in B} w_i\, b_i \leq k\}$
  Actualize at-most constraints

Computation of the New Bound

$$\text{newbound}(AL, B) = \min\{k > \text{bound}(AL, B) \mid AL \cup \{\sum_{i \in B} w_i\, b_i = k\} \text{is SAT}\}$$

example

$$AL = \{\ 10\, b_1 + 4\, b_2 \geq 4,$$
$$8\, b_3 + 2\, b_4 \geq 2\}$$
$$B = \{1, 2, 3, 4\}$$

Then,

$$\text{bound}(AL, B) = 6$$
$$\text{newbound}(AL, B) = 12$$

Computation of the New Bound

$$\text{newbound}(AL, B) = \min\{k > \text{bound}(AL, B) \mid AL \cup \{\sum_{i \in B} w_i\, b_i = k\} \text{is SAT}\}$$

Computing newbound is NP-hard.

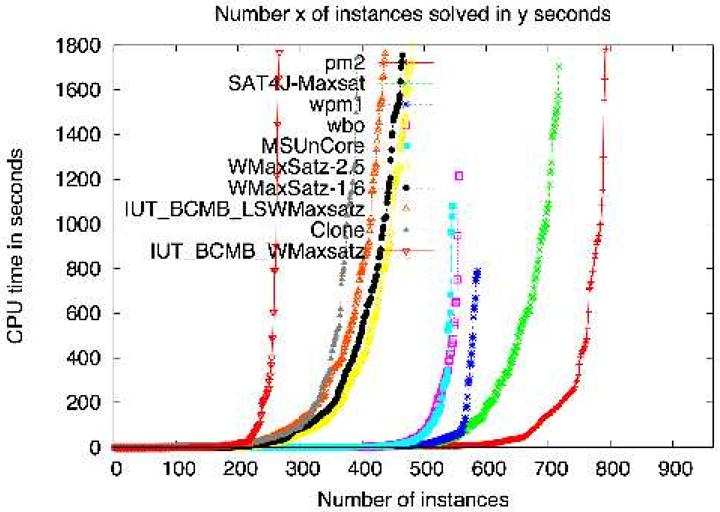
Computable as:

function newbound(AL, B)
$k :=$ bound(AL, B)
repeat
    $k = subsetsum(\{w_i \mid i \in B\}, k)$
until $SAT(CNF(AL \cup \{\sum_{i \in B} w_i\, b_i = k\}))$
return k

Experiments: Partial MaxSat Category

set	best08	WPM1	PM2.1	msu1.2	msu4.0	SAT4J
Crafted						
$\frac{Maxclique}{Random}$(96)	**2.4(96)**	50.4(1)	126(54)	-(0)	106(61)	114(52)
$\frac{Maxclique}{Structured}$(62)	**73(36)**	41.2(11)	62(12)	4.9(7)	105.2(13)	50.5(13)
$\frac{Maxone}{3SAT}$(80)	**0.46(80)**	16(46)	22(80)	52.7(40)	118.2(35)	96.6(31)
$\frac{Maxone}{Structured}$(60)	**10.1(60)**	0.69(2)	253(34)	122.7(2)	3.34(1)	**10.1(60)**
Industrial						
$\frac{Bcp}{fir}$(59)	49(46)	**32 (57)**	18(58)	49.2(46)	-(0)	13.3(10)
$\frac{Bcp}{hipp-yRa1}$(1183)	19(1111)	3(1122)	**13.5(1163)**	7.2(1105)	0.29(348)	12.2(1109)
$\frac{Bcp}{msp}$(148)	**49(104)**	15.5(26)	384.2(36)	4.9(25)	22.9(79)	8.8(93)
$\frac{Bcp}{mtg}$(215)	26(206)	5.8(170)	**10.5(214)**	17.5(164)	0.43(22)	57(196)
$\frac{Bcp}{syn}$(74)	**63(34)**	14.1(32)	**71.2(34)**	51.1(31)	105.2(11)	67.4(21)
$\frac{Pbo}{mqc-nencdr}$(128)	**167(115)**	80.4(50)	142(78)	50.3(54)	**167.5(115)**	180.6(102)
$\frac{Pbo}{mqc-nlogencdr}$(128)	**111(128)**	67.1(75)	140.3(97)	53(65)	**111(128)**	117.5(126)
$\frac{Pbo}{routing}$(15)	2.9(15)	**1(15)**	24.7(15)	2.9(15)	54.9(15)	26.4(9)

Industrial


Number x of instances solved in y seconds
CPU time in seconds
Number of instances
1800
1600
1400
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
700
800
900
pm2
SAT4J-Maxsat
wpm1
wbo
MSUnCore
WMaxSatz-2.5
WMaxSatz-1.6
IUT_BCMB_LSWMaxsatz
Clone
IUT_BCMB_WMaxsatz