Parallel Search for Maximum Satisfiability

Ruben Martins

IST, Technical University of Lisbon, Portugal

September 9, 2013

What is Boolean Satisfiability?

CNF Formula:

$$\bar{x}_2 \vee \bar{x}_1 \qquad x_2 \vee \bar{x}_3 \qquad x_1$$

$$x_3 \qquad x_2 \vee \bar{x}_1 \qquad \bar{x}_3 \vee x_1$$

- Boolean Satisfiability (SAT):
  - Decide if the formula is satisfiable or unsatisfiable

What is Boolean Satisfiability?

CNF Formula:

$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	x_1
x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable

What is Boolean Satisfiability?

CNF Formula:

$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	x_1
x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable
- How many clauses can we satisfy?

What is Maximum Satisfiability?

CNF Formula:

$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	x_1
x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Maximum Satisfiability (MaxSAT):
  - Find an assignment that maximizes (minimizes) number of satisfied (unsatisfied) clauses

What is Maximum Satisfiability?

CNF Formula:

$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	x_1
x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- An optimal solution would be:
  - $\nu = \{x_1 = 1, x_2 = 1, x_3 = 1\}$
- This assignment unsatisfies only 1 clause

MaxSAT Problems

- MaxSAT:
  - All clauses are soft
  - Minimize number of unsatisfied soft clauses
- Partial MaxSAT:
  - Clauses are soft or hard
  - Hard clauses must be satisfied
  - Minimize number of unsatisfied soft clauses
- Weighted Partial MaxSAT:
  - Clauses are soft or hard
  - Weights associated with soft clauses
  - Minimize sum of weights of unsatisfied soft clauses

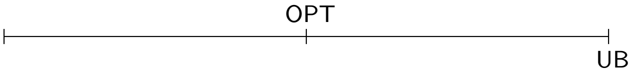
Why is MaxSAT Important?

- Many real-world applications can be encoded to MaxSAT:
  - Software package upgradeability:
    - Eclipse platform uses MaxSAT for managing the plugins dependencies
  - Error localization in C code
  - Debugging of hardware designs
  - Haplotyping with pedigrees
  - Reasoning over Biological Networks
  - Course timetabling
  - Combinatorial auctions
  - ...
- MaxSAT algorithms are effective for solving real-word problems

Outline

- MaxSAT Algorithms:
  - Linear search algorithms
  - Unsatisfiability-based algorithms
- Parallel MaxSAT:
  - Parallel algorithms
  - Deterministic approaches
  - Clause sharing heuristics
- Sequential MaxSAT:
  - Partitioning-based algorithms

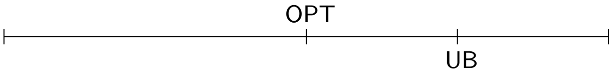
Linear Search Algorithms



- Optimum solution (OPT):
  - Assignment with **minimum** cost
- Upper Bound (UB) value:
  - Cost **greater than or equal** to OPT

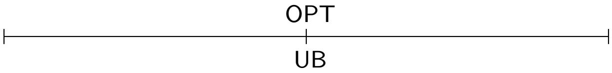
- Linear search algorithms:
  - Refine UB value until OPT is found

Linear Search Algorithms



- Optimum solution (OPT):
  - Assignment with **minimum** cost
- Upper Bound (UB) value:
  - Cost **greater than or equal** to OPT
- Linear search algorithms:
  - Refine UB value until OPT is found

Linear Search Algorithms



- Optimum solution (OPT):
  - Assignment with **minimum** cost
- Upper Bound (UB) value:
  - Cost **greater than or equal** to OPT

- Linear search algorithms:
  - Refine UB value until OPT is found

Linear Search Algorithms

Partial MaxSAT Formula:

φ_h (Hard):		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s (Soft):	x_1	x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

Linear Search Algorithms

Partial MaxSAT Formula:

$$\varphi_h : \quad \bar{x}_2 \vee \bar{x}_1 \qquad x_2 \vee \bar{x}_3$$

$$\varphi_s : \quad x_1 \vee \boxed{r_1} \qquad x_3 \vee \boxed{r_2} \qquad x_2 \vee \bar{x}_1 \vee \boxed{r_3} \qquad \bar{x}_3 \vee x_1 \vee \boxed{r_4}$$

- Relax all soft clauses
- Relaxation variables:
  - $V_R = \{r_1, r_2, r_3, r_4\}$
  - If a soft clause ω_i is **unsatisfied**, then $r_i = 1$
  - If a soft clause ω_i is **satisfied**, then $r_i = 0$

Linear Search Algorithms

Partial MaxSAT Formula:

φ_h :		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s :	$x_1 \vee r_1$	$x_3 \vee r_2$	$x_2 \vee \bar{x}_1 \vee r_3$	$\bar{x}_3 \vee x_1 \vee r_4$

$V_R = \{r_1, r_2, r_3, r_4\}$

- Formula is satisfiable
  - $\nu = \{x_1 = 1, x_2 = 0, x_3 = 0, r_1 = 0, r_2 = 1, r_3 = 1, r_4 = 0\}$
- **Goal:** Minimize the number of relaxation variables assigned to 1

Linear Search Algorithms

Partial MaxSAT Formula:

φ_h :		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s :	$x_1 \vee r_1$	$x_3 \vee r_2$	$x_2 \vee \bar{x}_1 \vee r_3$	$\bar{x}_3 \vee x_1 \vee r_4$

$\mu = 2 \qquad V_R = \{r_1, r_2, r_3, r_4\}$

- r_2 and r_3 were assigned truth value 1:
  - Current solution unsatisfies 2 soft clauses
- Can less than 2 soft clauses be unsatisfied?

Linear Search Algorithms

Partial MaxSAT Formula:

$$\varphi_h : \quad \bar{x}_2 \vee \bar{x}_1 \quad x_2 \vee \bar{x}_3 \quad \text{CNF}(\textstyle\sum_{r_i \in V_R} r_i \leq 1)$$

$$\varphi_s : \quad x_1 \vee r_1 \quad x_3 \vee r_2 \quad x_2 \vee \bar{x}_1 \vee r_3 \quad \bar{x}_3 \vee x_1 \vee r_4$$

$$\mu = 2 \qquad V_R = \{r_1, r_2, r_3, r_4\}$$

- Add cardinality constraint to refine UB value:
  - $\text{CNF}(r_1 + r_2 + r_3 + r_4 \leq 1)$

Linear Search Algorithms

Partial MaxSAT Formula:

$$\varphi_h : \quad \bar{x}_2 \vee \bar{x}_1 \qquad x_2 \vee \bar{x}_3 \qquad \mathrm{CNF}(\textstyle\sum_{r_i \in V_R} r_i \leq 1)$$

$$\varphi_s : \quad x_1 \vee r_1 \qquad x_3 \vee r_2 \qquad x_2 \vee \bar{x}_1 \vee r_3 \qquad \bar{x}_3 \vee x_1 \vee r_4$$

$$\mu = 2 \qquad V_R = \{r_1, r_2, r_3, r_4\}$$

- Formula is unsatisfiable:
  - There are no solutions that unsatisfy 1 or less soft clauses

Linear Search Algorithms

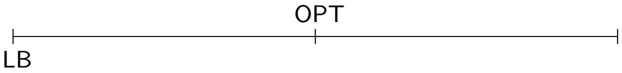
Partial MaxSAT Formula:

φ_h:		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s:	x_1	x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

$\mu = 2 \qquad V_R = \{r_1, r_2, r_3, r_4\}$

- **Optimal solution**:
  - $\nu = \{x_1 = 1, x_2 = 0, x_3 = 0\}$

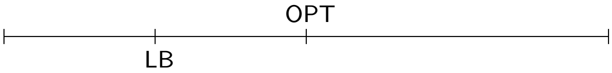
Unsatisfiability-based Algorithms



- Lower Bound (LB) value:
  - Cost **smaller than or equal** to OPT

- Unsatisfiability-based algorithms:
  - Use unsatisfiable cores to refine LB value until OPT is found

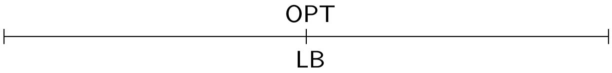
Unsatisfiability-based Algorithms



- Lower Bound (LB) value:
  - Cost **smaller than or equal** to OPT

- Unsatisfiability-based algorithms:
  - Use unsatisfiable cores to refine LB value until OPT is found

Unsatisfiability-based Algorithms



- Lower Bound (LB) value:
  - Cost **smaller than or equal** to OPT

- Unsatisfiability-based algorithms:
  - Use unsatisfiable cores to refine LB value until OPT is found

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h (Hard):		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s (Soft):	x_1	x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h:		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s:	x_1	x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h:		$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	
φ_s:	x_1	x_3	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable
- Identify an unsatisfiable core

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h: $\bar{x}_2 \vee \bar{x}_1$ $\quad x_2 \vee \bar{x}_3$ $\quad \boxed{\text{CNF}(r_1 + r_2 \leq 1)}$

φ_s: $x_1 \vee \boxed{r_1}$ $\quad x_3 \vee \boxed{r_2}$ $\quad x_2 \vee \bar{x}_1$ $\quad \bar{x}_3 \vee x_1$

- Relax unsatisfiable core:
  - Add relaxation variables
  - Add at-most-one constraint

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h:	$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	$\text{CNF}(r_1 + r_2 \leq 1)$	
φ_s:	$x_1 \vee r_1$	$x_3 \vee r_2$	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h:	$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	$\text{CNF}(r_1 + r_2 \leq 1)$	
φ_s:	$x_1 \vee r_1$	$x_3 \vee r_2$	$x_2 \vee \bar{x}_1$	$\bar{x}_3 \vee x_1$

- Formula is unsatisfiable
- Identify an unsatisfiable core

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

$$\varphi_h: \quad \bar{x}_2 \vee \bar{x}_1 \qquad x_2 \vee \bar{x}_3 \qquad \text{CNF}(r_1 + r_2 \leq 1) \quad \boxed{\text{CNF}(r_3 + \ldots + r_6 \leq 1)}$$

$$\varphi_s: \quad x_1 \vee r_1 \vee \boxed{r_3} \quad x_3 \vee r_2 \vee \boxed{r_4} \qquad x_2 \vee \bar{x}_1 \vee \boxed{r_5} \qquad \bar{x}_3 \vee x_1 \vee \boxed{r_6}$$

- Relax unsatisfiable core:
  - Add relaxation variables
  - Add at-most-one constraint

Unsatisfiability-based Algorithms

Partial MaxSAT Formula:

φ_h:	$\bar{x}_2 \vee \bar{x}_1$	$x_2 \vee \bar{x}_3$	$\text{CNF}(r_1 + r_2 \leq 1)$	$\text{CNF}(r_3 + \ldots + r_6 \leq 1)$
φ_s:	$x_1 \vee r_1 \vee r_3$	$x_3 \vee r_2 \vee r_4$	$x_2 \vee \bar{x}_1 \vee r_5$	$\bar{x}_3 \vee x_1 \vee r_6$

- Formula is satisfiable
- **Optimal solution**:
  - $\nu = \{x_1 = 1, x_2 = 0, x_3 = 0\}$

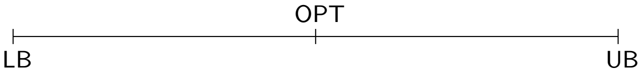
Outline

- MaxSAT Algorithms:
    - Linear search algorithms
    - Unsatisfiability-based algorithms
- Parallel MaxSAT:
    - Parallel algorithms
    - Deterministic approaches
    - Clause sharing heuristics
- Sequential MaxSAT:
    - Partitioning-based algorithms

Why Parallel MaxSAT?

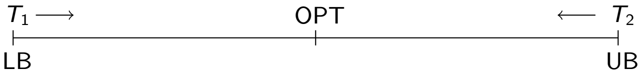
- Multicore processors are now predominant
- Several parallel SAT solvers have emerged:
  - Search space splitting
  - Portfolio
- Parallel approaches for MaxSAT are just starting
- This thesis presents the **first** parallel MaxSAT algorithms

Parallel MaxSAT (2 threads) [RCRA'11, AI Comm.'12]



- Linear search algorithms:
  - UB search
- Unsatisfiability-based algorithms:
  - LB search

Parallel MaxSAT (2 threads) [RCRA'11, AI Comm.'12]

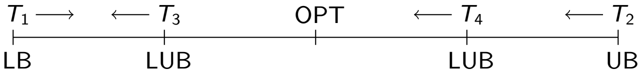


- Linear search algorithms:
  - UB search
- Unsatisfiability-based algorithms:
  - LB search
- Parallel search:
  - Search on LB and UB of the optimal solution
  - Exchange information

Parallel MaxSAT (n threads) [RCRA'11, ICTAI'11, AI Comm.'12]

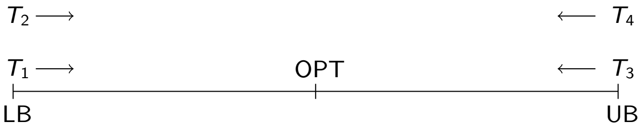
- Splitting approach:
  - Search on different values of the upper bound
- Portfolio approach:
  - Multiple threads perform lower and upper bound search

Parallel MaxSAT (Splitting) [RCRA'11, AI Comm.'12]



- Local Upper Bound (LUB):
  - Cost between LB and UB
- Splitting approach:
  - 1 thread searches on the LB (T_1)
  - 1 thread searches on the UB (T_2)
  - Remaining threads search on LUB (T_3, T_4)

Parallel MaxSAT (Portfolio) [ICTAI'11, AI Comm.'12]



- Portfolio approach:
  - Half threads search on the LB (T_1, T_2)
  - Half threads search on the UB (T_3, T_4)
  - Diversification of the search using **different** cardinality constraints

Parallel MaxSAT (Experimental Results)

- Benchmarks: 497 industrial partial MaxSAT instances
- Timeout: 1,800 seconds
- Solvers:
  - WBO (sequential MaxSAT solver - 1 run):
    - Uses a linear search algorithm for 10% of the time limit
    - If no solution is found changes to unsatisfiability-based algorithm
  - PWBO (parallel MaxSAT solver - 10 runs):
    - 2 threads, PWBO-T2
    - 4 and 8 threads, PWBO-S (splitting approach)
    - 4 and 8 threads, PWBO-P (portfolio approach)

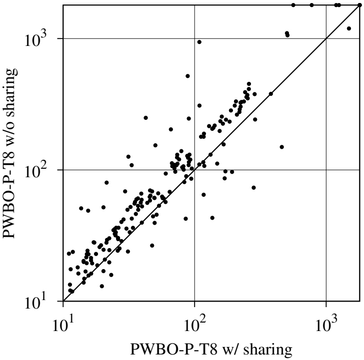
Parallel MaxSAT (Experimental Results)

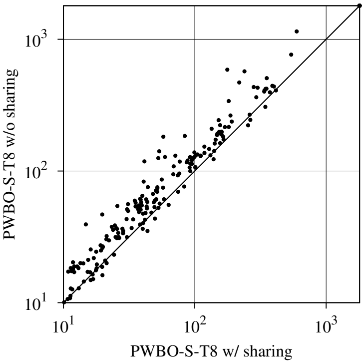
- Number of instances solved by each solver and speedup of PWBO:

	#Solved	Speedup
WBO	317	1.00
PWBO-T2	398	2.69
PWBO-P-T4	399	3.92
PWBO-S-T4	399	4.16
PWBO-S-T8	399	4.83
PWBO-P-T8	403	5.19

Parallel MaxSAT (Experimental Results)

- Impact of sharing learned clauses:





Deterministic Parallel MaxSAT [RCRA'12, AI Comm.'13]

- PWBO exhibits non-deterministic behavior:
  - Different runs of the solver may find different solutions
- 504 partial industrial benchmarks (10 runs):

#Solved	Avg. #Models	Avg. Δ run time
405	7.52	20.77%

Deterministic Parallel MaxSAT [RCRA'12, AI Comm.'13]

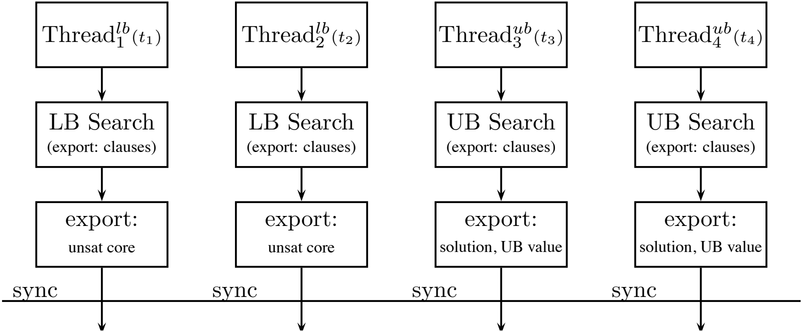
- PWBO exhibits non-deterministic behavior:
  - Different runs of the solver may find different solutions
- 504 partial industrial benchmarks (10 runs):

#Solved	Avg. #Models	Avg. Δ run time
405	7.52	20.77%

- **Solution:** import information at fixed points during the search

Deterministic Parallel MaxSAT [RCRA'12, AI Comm.'13]

- Deterministic version of PWBO-P:



- The definition of synchronization points must be deterministic:
  - Synchronize after k conflicts (period)

Deterministic Parallel MaxSAT [RCRA'12, AI Comm.'13]

Different kinds of synchronization:

- Standard synchronization:
  - Lower bound search syncs at each core
- Period synchronization:
  - Lower bound search syncs at each period
- Dynamic synchronization:
  - Dynamically adjust the size of the period

Deterministic Parallel MaxSAT (Results)

- Comparison between non-deterministic and deterministic solvers:

Solver	#Solved	Speedup
Non-Deterministic	405	1.00
Standard	400	0.77
Period	400	0.88
Dynamic	401	0.90

- Performance of deterministic solvers are comparable to performance of non-deterministic solver

Clause Sharing Heuristics [LION'12, AI Comm.'13]

- Sharing learned clauses improves the performance of the solver
- Not all learned clauses should be shared

Clause Sharing Heuristics [LION'12, AI Comm.'13]

- Sharing learned clauses improves the performance of the solver
- Not all learned clauses should be shared
- **Question:** which learned clauses should be shared?

Clause Sharing Heuristics [LION'12, AI Comm.'13]

- Static:
  - Learned clauses are shared within a given cutoff
- Dynamic:
  - Dynamic heuristics adjust the cutoff during the search
- Freezing:
  - Shared clauses are frozen until expected to be useful

Clause Sharing Heuristics (Experimental Results)

- Benchmarks: 504 industrial partial MaxSAT instances
- Portfolio version of PWBO with 4 threads:
  - Fair evaluation: dynamic deterministic version of PWBO was used
- Solvers:
  - Static heuristics: LBD 5, Size 8, Size 32
  - Dynamic heuristic: starts with a cutoff of size 8
  - Freezing heuristic: uses a cutoff of size 32

Clause Sharing Heuristics (Experimental Results)

- Comparison between clause sharing heuristics:

	Avg. #Clauses	Avg. Size	#Solved	Speedup
No Sharing	-	-	400	1.00
Random	40,686.10	99.57	400	1.12
LBD 5	20,822.01	12.66	401	1.25
Size 8	16,903.33	5.41	401	1.28
Size 32	48,687.91	13.42	401	1.24
Dynamic	28,496.23	8.57	401	1.38
Freezing	31,827.38	10.93	402	1.37

Outline

- MaxSAT Algorithms:
  - Linear search algorithms
  - Unsatisfiability-based algorithms
- Parallel MaxSAT:
  - Parallel algorithms
  - Deterministic approaches
  - Clause sharing heuristics
- Sequential MaxSAT:
  - Partitioning-based algorithms

Improving Sequential MaxSAT [ECAI'12, SAT'13]

- Unsatisfiability-based algorithms are very effective
- Performance is related with unsatisfiable cores given by SAT solver:
  - Some unsatisfiable cores may be unnecessarily large
  - **Solution:** Partitioning of the soft clauses

Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

$$\boxed{\gamma_1} \quad \boxed{\gamma_2} \quad \boxed{\gamma_3}$$

Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:

- Relax unsatisfiable core

Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
   - Relax unsatisfiable core

(4) The formula is satisfiable:
   - If there are no more partitions:
     ▷ Optimum found
   - Otherwise, **go back to 2**

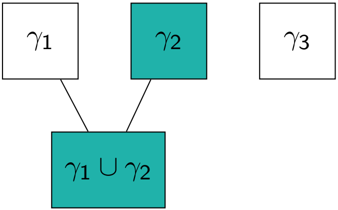
Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
   - Relax unsatisfiable core

(4) The formula is satisfiable:
   - If there are no more partitions:
     ▷ Optimum found
   - Otherwise, go back to 2



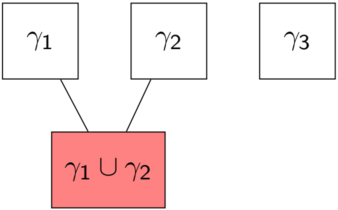
Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
- Relax unsatisfiable core

(4) The formula is satisfiable:
- If there are no more partitions:
  ▷ Optimum found
- Otherwise, go back to 2



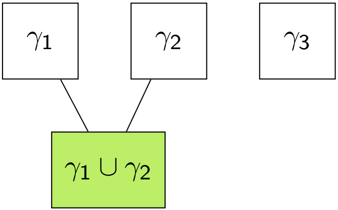
Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
- Relax unsatisfiable core

(4) The formula is satisfiable:
- If there are no more partitions:
  ▷ Optimum found
- Otherwise, **go back to 2**



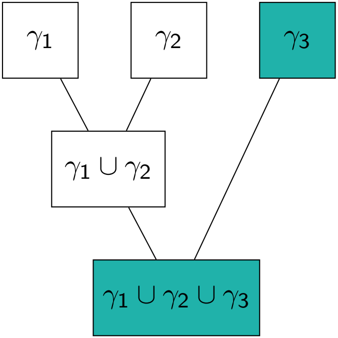
Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
  - Relax unsatisfiable core

(4) The formula is satisfiable:
  - If there are no more partitions:
    ▷ Optimum found
  - Otherwise, go back to 2



Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

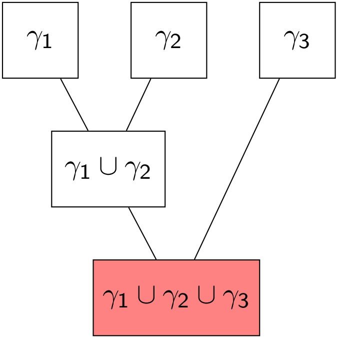
(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:

- Relax unsatisfiable core

(4) The formula is satisfiable:

- If there are no more partitions:
  ▷ Optimum found
- Otherwise, go back to 2



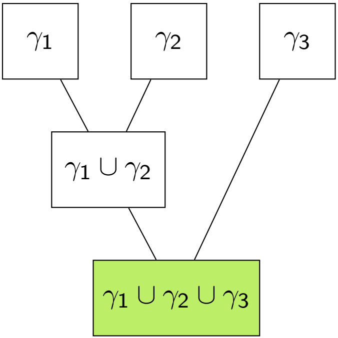
Partition-based MaxSAT algorithms [ECAI'12]

(1) Partition the soft clauses

(2) Add a new partition to the formula

(3) While the formula is unsatisfiable:
  - Relax unsatisfiable core

(4) The formula is satisfiable:
  - If there are no more partitions:
    ▷ **Optimum found**
  - Otherwise, go back to 2



MaxSAT Partitioning [ECAI'12, SAT'13]

- Weight-based partitioning:
  - Soft clauses with the same weight belong to the same partition
- Graph-based partitioning:
  - Hypergraph graph representation
  - Variable Incidence Graph (VIG) representation
  - Clause-Variable Incidence Graph (CVIG) representation

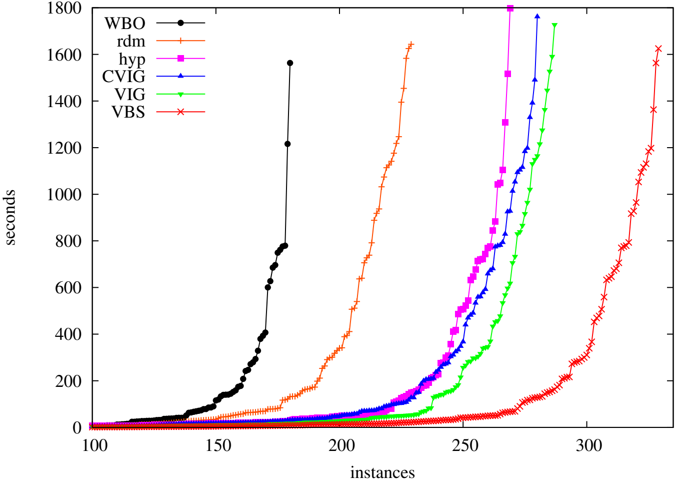
MaxSAT Partitioning (Results)

- Benchmarks:
  - 504 industrial partial MaxSAT instances
  - 598 weighted partial MaxSAT instances

- Solvers:
  - WBO
  - WEIGHT (Weight-based partitioning)
  - RDM (Random partitioning – 16 partitions)
  - HYP (Hypergraph partitioning – 16 partitions)
  - VIG (Community partitioning – Variable Incidence Graph)
  - CVIG (Community partitioning – Clause-Variable Incidence Graph)
  - VBS (Virtual Best Solver)

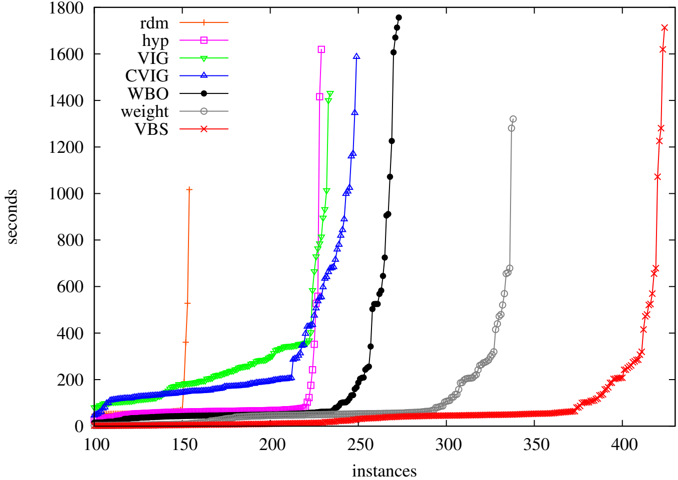
MaxSAT Partitioning (Results)

- Running times of solvers for industrial partial MaxSAT instances



MaxSAT Partitioning (Results)

- Running times of solvers for weighted partial MaxSAT instances



Conclusions

- PWBO first parallel MaxSAT solver for multicore architectures:
  - Winner of several tracks in the MaxSAT evaluations
  - Publicly available: `http://sat.inesc-id.pt/pwbo/`
- Deterministic parallel MaxSAT solvers have comparable performance to non-deterministic
- Sharing learned clauses boost the performance of the solver
- Partitioning-based techniques improves sequential MaxSAT

Publications

- International Journals, Conferences, Workshops:
  - 2013: AI Comm.'13*, SAT'13, RCRA'13
  - 2012: Constraints'12, AI Comm.'12, ECAI'12, LION'12, RCRA'12
  - 2011: ICTAI'11, RCRA'11
  - 2010: SAT'10, ICTAI'10
  - 2009: ModRef'09

$*$ Under review