Approximation Strategies for Incomplete MaxSAT

Saurabh Joshi[1] Prateek Kumar[1] Ruben Martins[2] Sukrut Rao[1]

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

[1]IIT Hyderabad [2]Carnegie Mellon University

SAT+SMT School 2019, IIT Bombay

8th December 2019

MaxSAT

$$\begin{aligned} (x_1 \vee x_2) \wedge \\ (\neg x_1 \vee x_2) \wedge \\ (x_1 \vee \neg x_2) \wedge \\ (\neg x_1 \vee \neg x_2) \end{aligned}$$

MaxSAT

$$\begin{aligned} (x_1 \vee x_2)\wedge \\ (\neg x_1 \vee x_2)\wedge \\ (x_1 \vee \neg x_2)\wedge \\ (\neg x_1 \vee \neg x_2) \end{aligned}$$

Unsat

MaxSAT

$$(x_1 \vee x_2) \wedge \quad \text{Unsat}$$
$$(\neg x_1 \vee x_2) \wedge$$
$$(x_1 \vee \neg x_2) \wedge$$
$$(\neg x_1 \vee \neg x_2)$$

$$(x_1 \vee x_2 \vee r_1) \wedge$$
$$(\neg x_1 \vee x_2 \vee r_2) \wedge$$
$$(x_1 \vee \neg x_2 \vee r_3) \wedge$$
$$(\neg x_1 \vee \neg x_2 \vee r_4)$$

MaxSAT

$$\begin{aligned}(x_1 \lor x_2)\land \\ (\neg x_1 \lor x_2)\land \\ (x_1 \lor \neg x_2)\land \\ (\neg x_1 \lor \neg x_2)\end{aligned}$$

Unsat

$$\begin{aligned}(x_1 \lor x_2 \lor r_1)\land \\ (\neg x_1 \lor x_2 \lor r_2)\land \\ (x_1 \lor \neg x_2 \lor r_3)\land \\ (\neg x_1 \lor \neg x_2 \lor r_4)\end{aligned}$$

$(\sum r_i) \leq k$

Cardinality Constraint

Minimize k (MaxSAT)

MaxSAT

$$\begin{array}{llr} w_1 & (x_1 \vee x_2)\wedge & \text{Unsat} \\ w_2 & (\neg x_1 \vee x_2)\wedge & \\ w_3 & (x_1 \vee \neg x_2)\wedge & \\ w_4 & (\neg x_1 \vee \neg x_2) & \end{array}$$

$$\begin{array}{rl} (x_1 \vee x_2 \vee r_1)\wedge & (\sum r_i) \leq k \\ (\neg x_1 \vee x_2 \vee r_2)\wedge & \text{Cardinality Constraint} \\ (x_1 \vee \neg x_2 \vee r_3)\wedge & \\ (\neg x_1 \vee \neg x_2 \vee r_4) & \end{array}$$

Minimize k (Weighted MaxSAT)

MaxSAT

$$\begin{array}{lrl} w_1 & (x_1 \vee x_2)\wedge & \text{Unsat} \\ w_2 & (\neg x_1 \vee x_2)\wedge & \\ w_3 & (x_1 \vee \neg x_2)\wedge & \\ w_4 & (\neg x_1 \vee \neg x_2) & \end{array}$$

$$\begin{array}{rl} (x_1 \vee x_2 \vee r_1)\wedge & \\ (\neg x_1 \vee x_2 \vee r_2)\wedge & \\ (x_1 \vee \neg x_2 \vee r_3)\wedge & (\sum w_i \cdot r_i) \leq k \\ (\neg x_1 \vee \neg x_2 \vee r_4) & \text{PB Constraint} \end{array}$$

Minimize k (Weighted MaxSAT)

Motivation for MaxSAT

- Operations Research
- Logistics
- Resource Allocation
- Computational Biology
- Fault Localization
- ... and many more

Motivation for MaxSAT

- Operations Research
- Logistics
- Resource Allocation
- Computational Biology
- Fault Localization
- ... and many more

For many applications it may be desirable to find a good solution (even if suboptimal) very quickly.

Motivation for MaxSAT

- Operations Research
- Logistics
- Resource Allocation
- Computational Biology
- Fault Localization
- ... and many more

For many applications it may be desirable to find a good solution (even if suboptimal) very quickly. That's where incomplete solvers come into play!

Motivation for MaxSAT

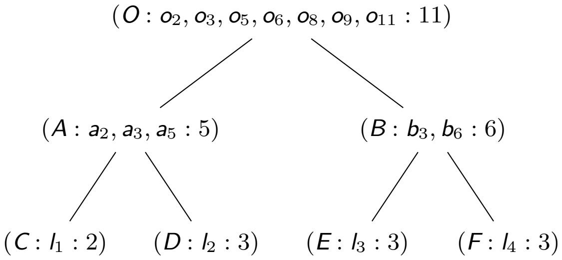
- Operations Research
- Logistics
- Resource Allocation
- Computational Biology
- Fault Localization
- ... and many more

For many applications it may be desirable to find a good solution (even if suboptimal) very quickly. That's where incomplete solvers come into play!

Our contributions

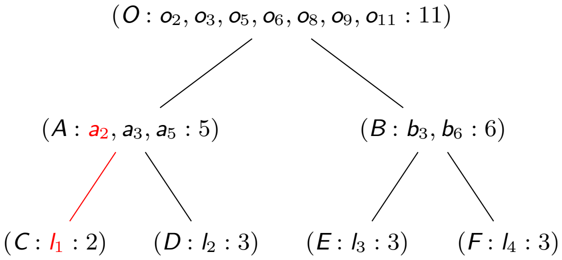
- Weight relaxation based approximation
- Subproblem minimization based approximation

GTE for Pseudo-Boolean Constraints



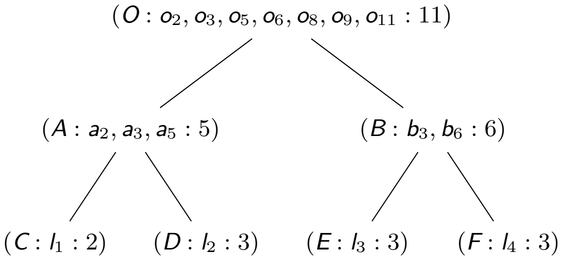
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$

GTE for Pseudo-Boolean Constraints



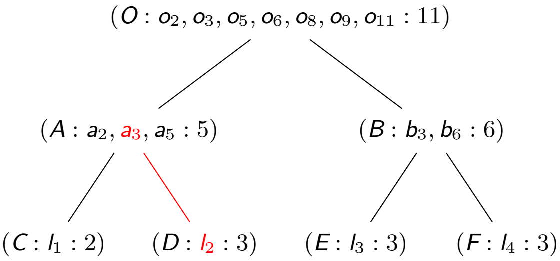
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2)$

GTE for Pseudo-Boolean Constraints



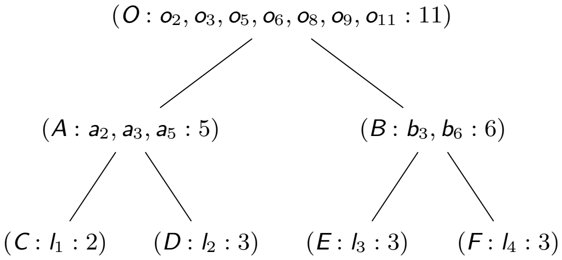
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2)$

GTE for Pseudo-Boolean Constraints



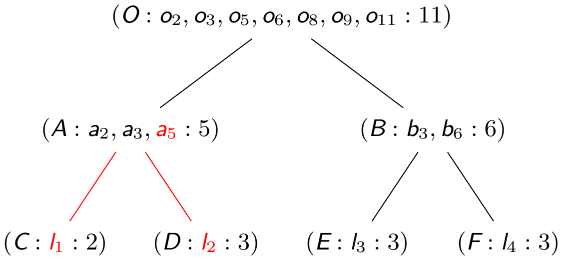
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3)$

GTE for Pseudo-Boolean Constraints



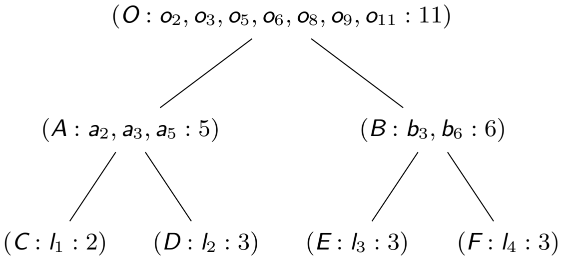
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3)$

GTE for Pseudo-Boolean Constraints



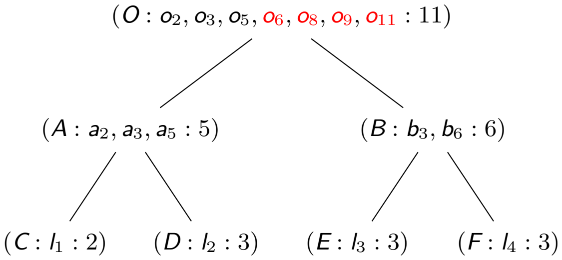
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5)$

GTE for Pseudo-Boolean Constraints



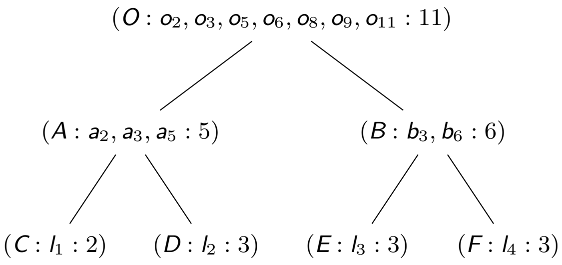
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5) \ldots$

GTE for Pseudo-Boolean Constraints



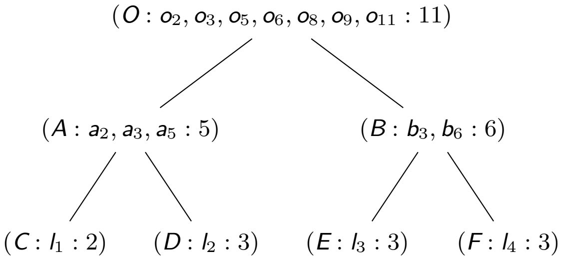
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4 \leq 5$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5) \ldots$
  $\neg o_6 \wedge \neg o_8 \wedge \neg o_9 \wedge \neg o_{11}$

GTE for Pseudo-Boolean Constraints



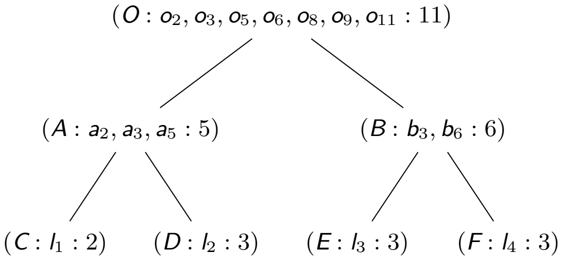
- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4 \leq 5$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5) \ldots$
  $\neg o_6 \wedge \neg o_8 \wedge \neg o_9 \wedge \neg o_{11}$
- Worst case exponential size (e.g., weights $1, 2, 4, 8, \ldots$)

GTE for Pseudo-Boolean Constraints



- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4 \leq 5$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5) \ldots$
  $\neg o_6 \wedge \neg o_8 \wedge \neg o_9 \wedge \neg o_{11}$
- Worst case exponential size (e.g., weights $1, 2, 4, 8, \ldots$)
- Polynomial size encoding when all the weights are same.

GTE for Pseudo-Boolean Constraints



- Encoding $2l_1 + 3l_2 + 3l_3 + 3l_4 \leq 5$
- $(\neg l_1 \vee a_2) \wedge (\neg l_2 \vee a_3) \wedge (\neg l_1 \vee \neg l_2 \vee a_5) \dots$
  $\neg o_6 \wedge \neg o_8 \wedge \neg o_9 \wedge \neg o_{11}$
- Worst case exponential size (e.g., weights $1, 2, 4, 8, \dots$)
- Polynomial size encoding when all the weights are same. This can be leveraged for incomplete solving.

Weight relaxation

- $m = 3$ are the number of clusters we want to form

$10 \quad 3 \quad 27 \quad 12 \quad 11 \quad 2 \quad 4 \quad 26 \quad 25$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order

$2 \quad 3 \quad 4 \quad 10 \quad 11 \quad 12 \quad 25 \quad 26 \quad 27$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster

$$\mid 2 \quad 3 \quad 4 \quad 10 \quad 11 \quad 12 \quad 25 \quad 26 \quad 27 \mid$$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary

$$\mid 2 \quad 3 \quad 4 \quad 10 \quad 11 \quad \color{red}{12} \mid \color{red}{25} \quad 26 \quad 27 \mid$$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary

$$\mid 2 \quad 3 \quad {\color{red}4} \mid {\color{red}10} \quad 11 \quad 12 \mid 25 \quad 26 \quad 27 \mid$$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary
- Replace weights by a representative weight of a cluster (say, arithmetic mean)

$$| \; 3 \quad 3 \quad 3 \; | \; 11 \quad 11 \quad 11 \; | \; 26 \quad 26 \quad 26 \; |$$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary
- Replace weights by a representative weight of a cluster (say, arithmetic mean)

$$\left|\ 3 \quad 3 \quad 3\ \right|\ 11 \quad 11 \quad 11\ \left|\ 26 \quad 26 \quad 26\ \right|$$

- Original problem: Minimize k in $(\sum w_i \cdot r_i) \leq k$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary
- Replace weights by a representative weight of a cluster (say, arithmetic mean)

$$\mid 3 \quad 3 \quad 3 \mid 11 \quad 11 \quad 11 \mid 26 \quad 26 \quad 26 \mid$$

- Original problem: Minimize k in $(\sum w_i \cdot r_i) \leq k$
- Modified problem: Minimize k in $(\sum w_i' \cdot r_i) \leq k$

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary
- Replace weights by a representative weight of a cluster (say, arithmetic mean)

$$\mid 3 \quad 3 \quad 3 \mid 11 \quad 11 \quad 11 \mid 26 \quad 26 \quad 26 \mid$$

- Original problem: Minimize k in $(\sum w_i \cdot r_i) \leq k$
- Modified problem: Minimize k in $(\sum w_i' \cdot r_i) \leq k$
- Keep decreasing k until you reach Unsat

Weight relaxation

- $m = 3$ are the number of clusters we want to form
- Sort clauses by weights in ascending order
- Initially everything in one cluster
- Keep dividing clusters by picking the largest weight difference as a cluster boundary
- Replace weights by a representative weight of a cluster (say, arithmetic mean)

$$| \; 3 \quad 3 \quad 3 \; | \; 11 \quad 11 \quad 11 \; | \; 26 \quad 26 \quad 26 \; |$$

- Original problem: Minimize k in $(\sum w_i \cdot r_i) \leq k$
- Modified problem: Minimize k in $(\sum w'_i \cdot r_i) \leq k$
- Keep decreasing k until you reach Unsat
- Keep reporting assignments ν with smallest $(\sum w_i \cdot r_i)$ seen so far

Weight relaxation

- As m increases accuracy increases. No approximation when $m = \#weights$
- Formula size increases as m increases thus making it more difficult for the solver
- If time permits, keep increasing m

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 1$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing ($\sum r_i$)

$r_8 + r_9 + r_{10} \leq 1$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 3$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 3$

20 20 20

r_8 r_9 r_{10}

4 4 4 4

r_4 r_5 r_6 r_7

1 1 1

r_1 r_2 r_3

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 2$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 2$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 3$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$$r_8 + r_9 + r_{10} \leq 2$$
$$r_4 + r_5 + r_6 + r_7 \leq 3$$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

- Greedy approach

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$r_8 + r_9 + r_{10} \leq 2$

$r_4 + r_5 + r_6 + r_7 \leq 3$

							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

- Greedy approach
- Does not converge to optimal if BMO property does not hold

Subproblem Minimization

- Weight at any level is higher than sum of all the weights below that level (BMO property)
- Starting from heaviest (highest) level keep reducing $(\sum r_i)$

$$r_8 + r_9 + r_{10} \leq 2$$
$$r_4 + r_5 + r_6 + r_7 \leq 3$$

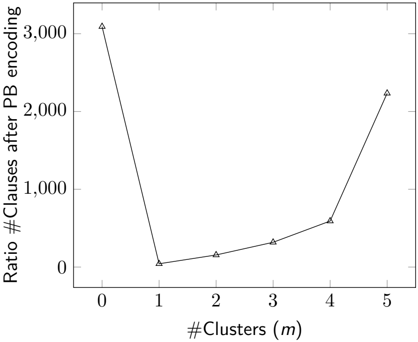
							20	20	20
							r_8	r_9	r_{10}
			4	4	4	4			
			r_4	r_5	r_6	r_7			
1	1	1							
r_1	r_2	r_3							

- Greedy approach
- Does not converge to optimal if BMO property does not hold
- Switch to alternatives if time permits (complete search, local search)

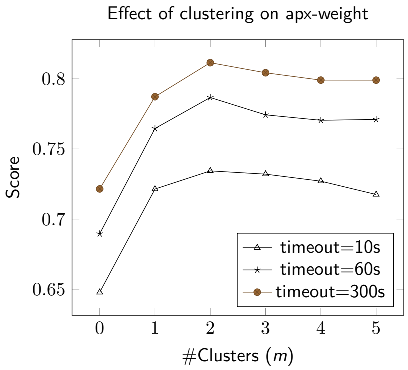
Experiments

- Techniques implemented as Open-WBO-Inc on top of Open-WBO framework
- MaxSAT evaluations 2017 benchmarks used
- Compared maxroster (MSE17-1), WPM3 (MSE17-2), QMaxSAT (MSE17-complete-2), apx-weight, apx-subprob
- Score $= \dfrac{\left(\sum_{b \in Benchmarks} \left(\dfrac{best(b)}{solver(b)}\right)\right)}{|Benchmarks|}$
  - The solver providing the best result for a benchmark scores 1
  - The score deteriorates as the result deviates from the best known
  - Score of 0 if solver fails for some reason.
- Timeout $= 10$s, 60s, 300s

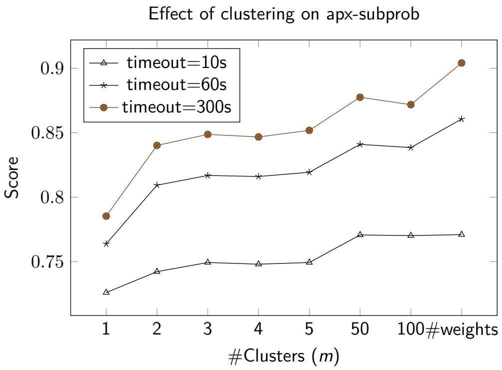
Results: Clustering effect on Formula size



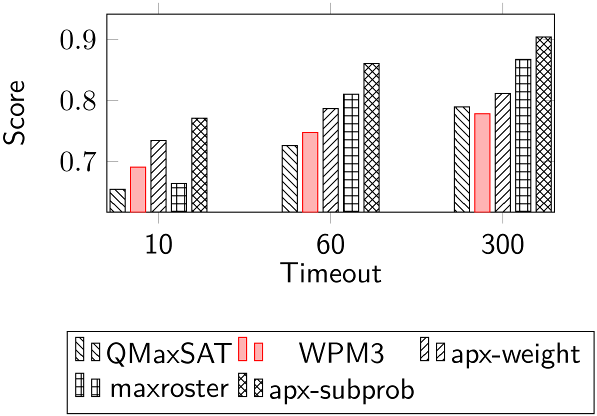
Results: Clustering effect on apx-weight performance



Results: Clustering effect on apx-subprob performance



Results: Comparision with other solvers



Results: Validation by others

- apx-weight placed fourth in MaxSAT 2018 evaluations in weighted incomplete tracks for 60s and 300s Timeout
- apx-subprob placed second in MaxSAT 2018 evaluations in weighted incomplete track for 300s Timeout
- apx-subprob placed first in MaxSAT 2018 evaluations in weighted incomplete track for 60s Timeout
- apx-subprob with switching to complete search placed sixth and fourth in MaxSAT 2019 in weighted incomplete tracks for 300s and60s Timeout respectively
- apx-subprob with switching to local search placed third in MaxSAT 2019 in weighted incomplete tracks for 300s and60s Timeout

Thank You!

Try Open-WBO-Inc :

`https://github.com/sbjoshi/Open-WBO-Inc`