Software Package Upgradeability Problem

* Set of packages to install $\{P_1, P_2, P_3, P_4\}$

* Each package P_i has a set of dependencies:
  → packages that must be installed for P_i to be installed.

* Each package P_i has set of conflicts:
  → package that cannot be installed for P_i to be installed.

Package	Dependencies	Conflicts
P_1	$\{P_2 \vee P_3\}$	$\{P_4\}$
P_2	$\{P_3\}$	$\{\}$
P_3	$\{P_2\}$	$\{P_4\}$
P_4	$\{P_2 \wedge P_3\}$	$\{\}$

Encode <u>Software Package Upgradeability Problem</u> as SAT!!

Encoding dependencies

$$P_1 \rightarrow (P_2 \vee P_3) \;\hat{=}\; (\neg P_1 \vee P_2 \vee P_3)$$

$$P_2 \rightarrow P_3 \;\hat{=}\; (\neg P_2 \vee P_3)$$

$$P_3 \rightarrow P_2 \;\hat{=}\; (\neg P_3 \vee P_2)$$

$$P_4 \rightarrow (P_2 \wedge P_3) \;\hat{=}\; (\neg P_4 \vee P_2) \wedge (\neg P_4 \vee P_3)$$

Encoding conflicts

$$P_1 \rightarrow \neg P_4 \;\hat{=}\; (\neg P_1 \vee \neg P_4)$$

$$P_3 \rightarrow \neg P_4 \;\hat{=}\; (\neg P_3 \vee \neg P_4)$$

Encoding installing all packages

$$P_1 \wedge P_2 \wedge P_3 \wedge P_4$$

Formula F :

dependencies $\neg P_1 \vee P_2 \vee P_3$ $\neg P_2 \vee P_3$ $\neg P_3 \vee P_2$

Conflicts $\neg P_4 \vee P_2$ $\neg P_4 \vee P_3$ $\neg P_1 \vee \neg P_4$ $\neg P_3 \vee \neg P_4$

Packages P_1 P_2 P_3 P_4

Is F SAT ?

Formula F :

dependencies $\neg P_1 \lor P_2 \lor P_3$ $\neg P_2 \lor P_3$ $\neg P_3 \lor P_2$

Conflicts $\neg P_4 \lor P_2$ $\neg P_4 \lor P_3$ $\neg P_1 \lor \neg P_4$ $\neg P_3 \lor \neg P_4$

Packages P_1 P_2 P_3 P_4

Formula F is UNSAT.

So, we can't install all packages.

How many packages can we install?

What is Maximum Satisfiability?

→ Maximum Satisfiability (MaxSAT):

→ Hard clauses: MUST be satisfied.
(e.g. conflicts, dependencies) ,me 7xtC

→ Soft clauses: DESIRABLE to be satisfied.
(e.g. package installation)

Goal: Maximize number of satisfied soft clauses.

Formula F :

Hard clauses	dependencies	$\neg P_1 \vee P_2 \vee P_3$	$\neg P_2 \vee P_3$	$\neg P_3 \vee P_2$	
	Conflicts	$\neg P_4 \vee P_2$	$\neg P_4 \vee P_3$	$\neg P_1 \vee \neg P_4$	$\neg P_3 \vee \neg P_4$
soft clauses	Packages	P_1	P_2	P_3	P_4

Goal : Maximize the number of installed packages.

Running Example :->

F_H : $\neg x_2 \vee \neg x_1$ $\quad$ $x_2 \vee \neg x_3$

F_S : x_1 $\quad$ x_3 $\quad$ $x_2 \vee \neg x_1$ $\quad$ $\neg x_3 \vee x_1$

$$F = F_H \wedge F_S$$

See that F is UNSAT.

tell us an algorithm to solve MaxSAT formula F !

F_H : $\neg x_2 \vee \neg x_1$ $x_2 \vee \neg x_3$

F_S : $x_1 \vee r_1$ $x_3 \vee r_2$ $x_2 \vee \neg x_1 \vee r_3$ $\neg x_3 \vee x_1 \vee r_4$

→ Relax all soft clauses.

→ Relaxation variables: $\{r_1, r_2, r_3, r_4\}$

Hint: if a soft clauses sc_i is unsatisfied, then $r_i = 1$

if a soft clauses sc_i is satisfied, then $r_i = 0$.

F_H : $\neg x_2 \vee \neg x_1$ $\quad$ $x_2 \vee \neg x_3$

F_S' : $x_1 \vee r_1$ $\quad$ $x_3 \vee r_2$ $\quad$ $x_2 \vee \neg x_1 \vee r_3$ $\quad$ $\neg x_3 \vee x_1 \vee r_4$

→ Relax all soft clauses.

→ Relaxation variables: $\{r_1, r_2, r_3, r_4\}$

Hint: if a soft clauses sc_i is unsatisfied, then $r_i = 1$

if a soft clauses sc_i is satisfied, then $r_i = 0$.

Goal: Minimize number of relaxation variables assigned to 1.

$F = F_H \wedge F'_S$

$\hookrightarrow$ with relaxation variables.

F is SAT.

let $\sigma \models F$, $\sigma = \{ x_1 \mapsto 1,\ x_2 \mapsto 0,\ x_3 \mapsto 0,$

$r_1 \mapsto 0,\ r_2 \mapsto 1,\ r_3 \mapsto 1,\ r_4 \mapsto 0 \}$

Minimize number of "R" variables assigned to $\underline{1}$.

$F = F_H \wedge \underbrace{F'_S}$ with relaxation variables.

let $\sigma \models F$, $\sigma = \{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0,$
$r_1 \mapsto 0, r_2 \mapsto 1, r_3 \mapsto 1, r_4 \mapsto 0\}$

Minimize number of "R" variables assigned to 1.

* Add cardinality constraint that excludes solutions that unsatisfies 2 or More soft constraints.

Add in the Hard constraints. $\longrightarrow$ $r_1 + r_2 + r_3 + r_4 \leq 1$

F_H' : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 + r_3 + r_4 \leq 1$

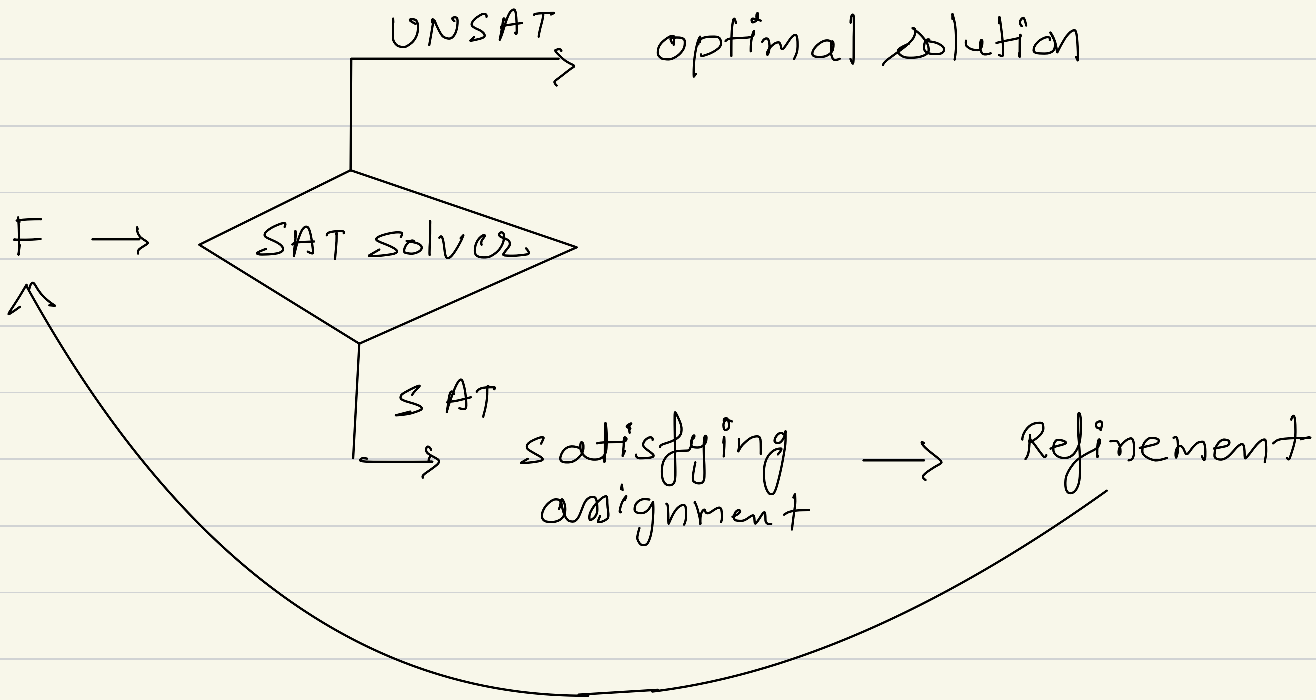
F_S' : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1 \vee r_3$ $\quad \neg x_3 \vee x_1 \vee r_4$

→ $F = F_H' \wedge F_S'$ is UNSAT.

→ There are no solutions that unsatisfy 1 or less soft clauses.

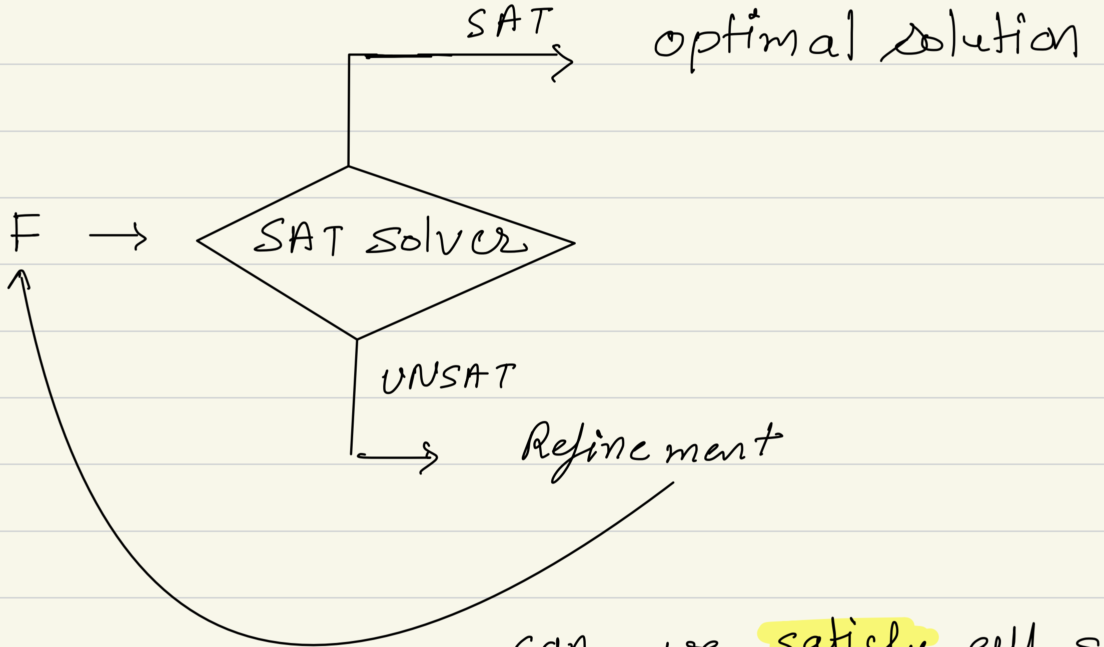
→ Optimal solution : $\langle x_1 \mapsto 1,\ x_2 \mapsto 0,\ x_3 \mapsto 0 \rangle$

Upper bound search for MaxSAT.



Can we unsatisfy less than k clauses

Similar approach for lower bound.



can we satisfy all soft clauses but K

Problem with Linear Search Algorithm :

→ Based on upper bound or lower bound MaxSAT solving.

1. # of relaxation variables.
2. size of cardinality constraints

→ We relax all soft clauses, then impose cardinality constraint.

→ Can we do better?

Unsatisfiability-Based Algorithm :

F_H : $\neg x_2 \vee \neg x_1$ $\quad$ $x_2 \vee \neg x_3$

F_S : x_1 $\quad$ x_3 $\quad$ $x_2 \vee \neg x_1$ $\quad$ $\neg x_3 \vee x_1$

$F = F_H \wedge F_S$ is UNSAT.

→ Identify an UNSAT CORE. $x_1 \wedge x_3 \wedge (\neg x_2 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)$

→ Relax "non-relaxed" soft constraint in UNSAT CORE.

F_H : $\neg x_2 \vee \neg x_1$ $\quad$ $x_2 \vee \neg x_3$

F_S : $x_1 \vee r_1$ $\quad$ $x_3 \vee r_2$ $\quad$ $x_2 \vee \neg x_1$ $\quad$ $\neg x_3 \vee x_1$

→ $r_1 + r_2 \leq 1$ (add this to F_H)

→ Relaxation on demand.

F'_H : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 \leq 1$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

* $F = F'_H \wedge F_S$ is UNSAT
* Again identify the unsat core :

F'_H : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 \leq 1$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

→ Relax non-relaxed soft clauses in UNSAT CORE.

F'_H : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 + r_3 + r_4 \leq 2$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1 \vee r_3$ $\quad \neg x_3 \vee x_1 \vee r_4$

Add cardinality constraint that excludes solutions that unsatisfies 3 or more soft clauses.

Problem with Unsatisfiability-Based Algorithm :

Worst case :

1. # of relaxation variables.
2. size of cardinality constraints

Can we do better?

Advanced Unsatisfiability-Based Algorithm :

F_H : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$

F_S : x_1 $\quad x_3$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

$F = F_H \wedge F_S$ is UNSAT.

→ Identify an UNSAT CORE. $x_1 \wedge x_3 \wedge (\neg x_2 \vee \neg x_1) \wedge (x_2 \vee \neg x_3)$

→ Relax "non-relaxed" soft constraint in UNSAT CORE.

F_H : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

→ $r_1 + r_2 \leq 1$] add atmost 1 constraint.

F_H' : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 \leq 1$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

* $F = F_H' \wedge F_S$ is UNSAT
* Again identify the unsat core :

F_H' : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 \leq 1$

F_S : $x_1 \vee r_1$ $\quad x_3 \vee r_2$ $\quad x_2 \vee \neg x_1$ $\quad \neg x_3 \vee x_1$

Relax unsatisfiable soft clause.

F_H' : $\neg x_2 \vee \neg x_1$ $\quad x_2 \vee \neg x_3$ $\quad r_1 + r_2 \leq 1$ $\quad r_3 + r_4 + r_5 + r_6 \leq 1$

F_S : $x_1 \vee r_1 \vee r_3$ $\quad x_2 \vee r_2 \vee r_4$ $\quad x_2 \vee \neg x_1 \vee r_5$ $\quad \neg x_3 \vee x_1 \vee r_6$

→ Add relaxation variable

→ Add Atmost 1 constraints.

Can you suggest another approach for MaxSAT?

Recall - Hitting set, MUS, MCS

Hitting Set and MaxSAT based MaxSAT Solving

→ Find an (implicit) hitting set HS of the unsat cores of F.

→ find a soluation to $\underline{F \setminus HS}$.

Weighted MaxSAT :

→ there will be a non-negative weight associated with each clauses.

→ find a truth assignment that maximize the combined weight of the satisfied clauses.

↳ Hard clauses = ∞ weight.
Soft clauses = 1 weight.
} problem we discussed so far.

→ Can we extend algorithms to handle weighted MaxSAT ??

MaxSAT Formula Format

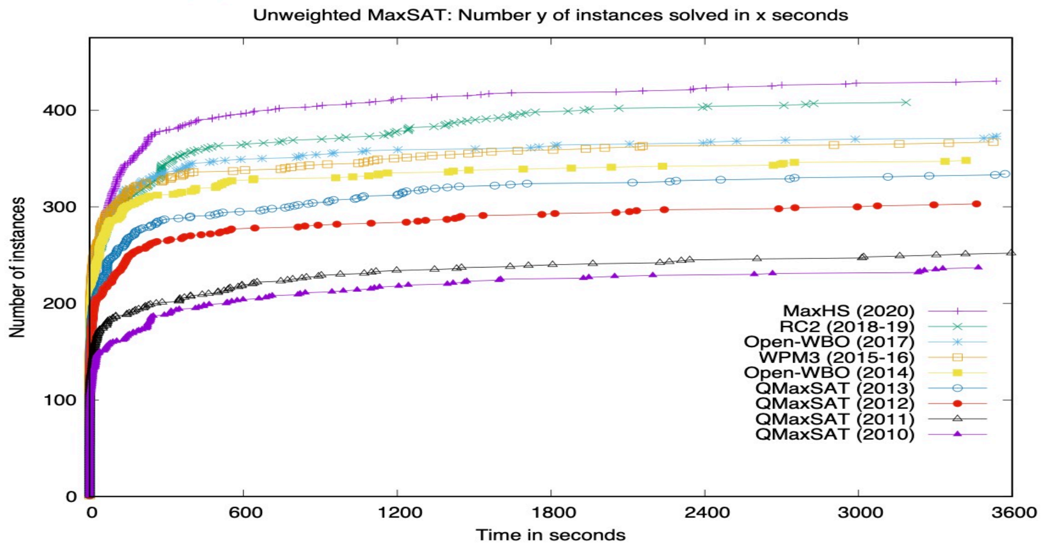
```
                          p wcnf 4 5 15
Hard constraints        { 15  1 -2 4 0
                        { 15 -1 -2 3 0
                        { 2 -2 -4 0
Soft constraints        { 5 -3 2 0
                        { 3 1 3 0
```

Represents "∞" (called top)
has to be greater than total weights of soft constraints

top = "weight" of hard clauses.

WDIMACS format

The MaxSAT (r)evolution



Comparing some of the best solvers from 2010-2020:

- In 2020: 81% more instances solved than in 2010!
- On same computer, same set of benchmarks