Branch-and-Bound MaxSAT Solving with MaxCDCL

Jordi Coll
Universitat de Girona

Djamal Habet, Kun He, Chu-Min Li, Shuolin Li, Felip Manyà, Zhenxing Xu

Lund, November 6, 2024

(Weighted) (Partial) MaxSAT

MaxSAT

Given:

- H: set of hard clauses
- S: a multiset of weighted soft clauses

Find an assignment ρ that:

- Satisfies all clauses in H
- Minimizes the sum of the weights of unsatisfied clauses in S

Example:

H: $(x \vee y), (x \vee z), (y \vee \overline{z})$ $\qquad$ S: $(x, 3), (y, 4), (\overline{z}, 5), (\overline{x} \vee z, 2)$

$\rho : x = 1, y = 1, z = 0,$ $\qquad$ (or $x\ y\ \overline{z}$)

$cost(\rho) = 2$

2021: State-of-the-art in MaxSAT Solving

State-of-the-art solvers are **SAT-based**:

- Problem translated to a series of satisfiability calls
- hitting set, model-based, core-based

Branch and Bound:

- Lack of clause learning
- Well-suited for random and some crafted
- Poor performance on industrial
- Not in MaxSAT evaluations since 2017

Solvers in MSE19-20	**HS**	**model**	**core**	**BnB**
EvalMaxSAT			X	
MaxHS	X			
Pacose		X		
UWrMaxSAT		X	X	
RC2			X	
Open-WBO			X	
maxino			X	
QMaxSAT		X		
smax			X	
ahmaxsat (MSE16)				X

2021: State-of-the-art in MaxSAT Solving

State-of-the-art solvers are **SAT-based**:

- Problem translated to a series of satisfiability calls
- hitting set, model-based, core-based

Branch and Bound:

- Lack of clause learning
- Well-suited for random and some crafted
- Poor performance on industrial
- Not in MaxSAT evaluations since 2017

MaxCDCL:

- BnB MaxSAT solver with clause learning
- Make BnB MaxSAT competitive for industrial MaxSAT

Solvers in MSE19-20	HS	model	core	BnB
EvalMaxSAT			X	
MaxHS	X			
Pacose		X		
UWrMaxSAT		X	X	
RC2			X	
Open-WBO			X	
maxino			X	
QMaxSAT		X		
smax			X	
ahmaxsat (MSE16)				X

MaxSAT in MaxCDCL

Normalization: express soft clauses as soft literals (blocking literals)

Convert:

H: $(x \vee y), (x \vee z), (y \vee \overline{z})$

S: $(x, 3), (y, 4), (\overline{z}, 5), (\overline{x} \vee z, 2)$

To:

H: $(x \vee y), (x \vee z), (y \vee \overline{z}), p \leftrightarrow (\overline{x} \vee \overline{z})$

S: $(x, 3), (y, 4), (\overline{z}, 5), (p, 2)$

MaxSAT

MaxSAT

Given:

- H: set of hard clauses
- S: a set of weighted soft literals

Find an assignment v that:

- Satisfies all clauses in H
- Minimizes the sum of the weights of unsatisfied literals in S

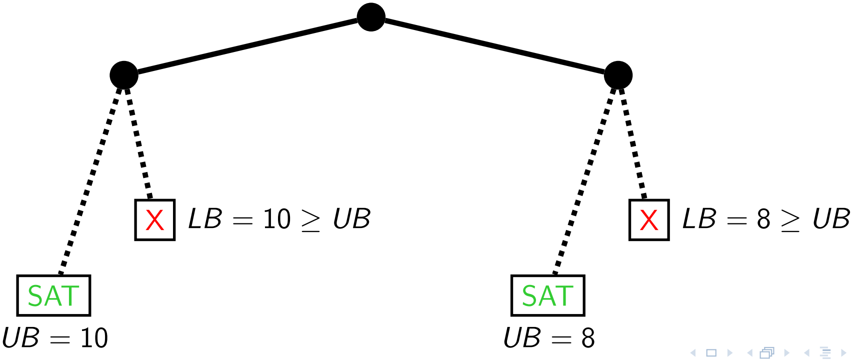
Example:

H: $(x \vee y), (x \vee z), (y \vee \overline{z}), p \leftrightarrow (\overline{x} \vee \overline{z})$ S: $(x, 3), (y, 4), (\overline{z}, 5), (p, 2)$

minimize $3\overline{x} + 4\overline{y} + 5z + 2\overline{p}$

Branch and Bound: [Abramé and Habet, 2014]

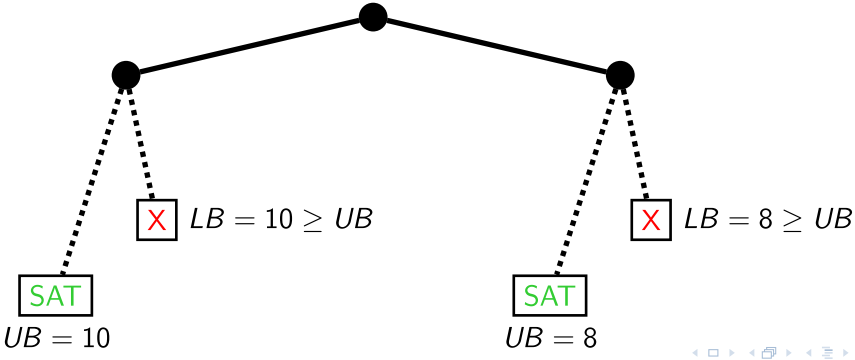
- Explore the search tree looking for optimal solutions
- Update Upper Bound UB when solution with better cost is found
- Prune branch when $LB \geq UB$
- Lookahead: underestimate LB of the cost at every node



MaxCDCL as Branch and Bound

Branch and Bound in MaxCDCL: [Li et. al, 2021, 2022]

- Explore the search tree **(CDCL)** looking for satisfiable assignments
- Update Upper Bound UB when solution with better cost is found
- Prune branch when $LB \geq UB$ **and learn a clause**
- Lookahead: underestimate LB of the cost at some nodes **using UP and core reasoning**



MaxCDCL as CDCL generalization

MaxCDCL execution: sequence of trails / assignments (ρ) that end with conflict

- **Hard conflict:**
  - A hard clause is falsified
- **Soft conflict:**
  - Check $cost(\rho)$
  - Maybe, underestimate LB of the cost
  - If LB$\geq$UB, an implicit hard clause is falsified

In both cases we do conflict analysis.

$x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p$ Conflict $\overline{x_5} \vee \overline{x_7}$ (**hard or soft conflict**)

$x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_5}^p \ \ldots$ Conflict analysis, Learn $\overline{x_3} \vee \overline{x_5}$, Backjump and UP

CDCL algorithm

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2^nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$x_3 \vee \overline{x}_5$

$\rho =$

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$x_3 \vee \overline{x}_5$

$\rho = x_1^d$

CDCL algorithm

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$x_3 \vee \overline{x}_5$

$\rho = x_1^d \; x_4^p$

CDCL algorithm

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3$

$\overline{x}_1 \lor x_4$

$\overline{x}_2 \lor \overline{x}_4 \lor x_5$

$x_3 \lor \overline{x}_5$

$\rho = x_1^d \; x_4^p \; x_2^d$

CDCL algorithm

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3$

$\overline{x}_1 \lor x_4$

$\overline{x}_2 \lor \overline{x}_4 \lor x_5$

$x_3 \lor \overline{x}_5$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p}$

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3$

$\overline{x}_1 \lor x_4$

$\overline{x}_2 \lor \overline{x}_4 \lor x_5$

$x_3 \lor \overline{x}_5$

$$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p} \; x_5^p$$

CDCL: Given hard clauses H, satisfy H

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP(H, ρ);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (SAT,ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$x_3 \vee \overline{x}_5$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p} \; x_5^p \qquad confl = (x_3 \vee \overline{x}_5)$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2^nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho =$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3$

$\overline{x}_1 \lor x_4$

$\overline{x}_2 \lor \overline{x}_4 \lor x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p}$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p} \; x_5^p$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$\overline{x}_2 \vee \overline{x}_4 \vee x_5$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p} \; x_5^p \quad confl = (x_3 \vee \overline{x}_5)$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

MaxCDCL algorithm

MaxCDCL: Given hard clauses H,soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho =$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d$

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2^nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d$

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \lor \overline{x}_2 \lor \overline{x}_3$

$\overline{x}_1 \lor x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p}$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Annotated assignment (trail):

l^d: decision

l^p: unit propagation

Example CNF + UB constraint:

$\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3$

$\overline{x}_1 \vee x_4$

$2\overline{x}_1 + 4\overline{x}_3 + 4x_5 < 8$

$x_3 \vee \overline{x}_5$

$\rho = x_1^d \; x_4^p \; x_2^d \; \overline{x_3^p} \; \overline{x_5^p}$

MaxCDCL algorithm

MaxCDCL: Given hard clauses H, soft literals S, upper bound UB, satisfy H and $cost(\rho) < UB$

```
ρ ← {}; /* Assignment */
while true do
    (confl, ρ) ← UP_UB_Harden_Lookahead(H, S, ρ, UB);
    if confl is a conflicting clause then
        if decisionLevel(ρ) = 0 then
            return UNSAT;
        else
            learntClause ← analyze(confl);
            H ← H ∪ {learntClause};
            dl ← 2nd highest decision level in learntClause;
            ρ ← backtrackTo(dl, ρ);
    else
        if all variables are assigned in ρ then
            return (cost(ρ), ρ);
        else
            l ← pickBranchLit();
            ρ ← ρ ∪ {l}; /* Make a decision */
```

Lookahead: LB underestimation, unweighted

Lookahead by UP for underestimating LB. At some nodes of the search:

1. Assume (y^a) unassigned soft literals and apply UP until:
    - A hard clause is falsified
    - Or a soft literal is falsified
2. We have found a local unsatisfiable core
3. Each **disjoint** core increases the LB by
4. Repeat the process until $LB \geq UB$, o no more disjoint cores are found
5. When $LB \geq UB$, a soft conflict is found

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8$

Initial trail: $x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p$

Find a core (by falsified clause):

$x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p \ y_1^a \ x_9^p \ x_{10}^p \ y_2^a \ \overline{x_{11}}^p \ y_3^a \ y_4^a \ x_{12}^p$ (Falsify $\overline{x_{12}} \vee x_{11}$)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8$

Initial trail: $x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p$

Find a core (by falsified clause):

$x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p \ y_1^a \ x_9^p \ x_{10}^p \ y_2^a \ \overline{x_{11}}^p \ y_3^a \ y_4^a \ x_{12}^p$ (Falsify $\overline{x_{12}} \vee x_{11}$)

$x_1^d \ \overline{x_2}^p \ x_3^p \ \overline{x_4}^d \ x_5^p \ x_6^p \ x_7^p \ y_1^a \ y_2^a \ y_3^a \ y_4^a$ (Assumptions suffice)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Find a core (by falsified clause):

x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p y_1^a x_9^p x_{10}^p y_2^a $\overline{x_{11}}^p$ y_3^a y_4^a x_{12}^p (Falsify $\overline{x_{12}} \vee x_{11}$)

x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p y_1^a y_2^a y_3^a y_4^a (Assumptions suffice)

$\overline{x_2}^p$ $\overline{x_4}^d$ y_1^a y_4^a (Conflict analysis)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \quad y_8$

Initial trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Find a core (by falsified clause):

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ y_1^a\ x_9^p\ x_{10}^p\ y_2^a\ \overline{x_{11}}^p\ y_3^a\ y_4^a\ x_{12}^p$ (Falsify $\overline{x_{12}} \vee x_{11}$)

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ y_1^a\ y_2^a\ y_3^a\ y_4^a$ (Assumptions suffice)

$\overline{x_2}^p\ \overline{x_4}^d\ y_1^a\ y_4^a$ (Conflict analysis)

Local core:

$\overline{x_2} \wedge \overline{x_4} \wedge y_1 \wedge y_4 \vdash UP \bot$

$\overline{x_2} \wedge \overline{x_4} \to \overline{y_1} \vee \overline{y_4}$ (Reasons $\to$ Core)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ y_2 y_3 $\cancel{y_4}$ y_5 y_6 y_7 y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\not{y_1}$ y_2 y_3 $\not{y_4}$ y_5 y_6 y_7 y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Find a core (by falsified soft literal):

x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p y_2^a y_3^a $\overline{y_5}^p$ (Falsify y_5)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\not{y_1}$ y_2 y_3 $\not{y_4}$ y_5 y_6 y_7 y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Find a core (by falsified soft literal):

x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p y_2^a y_3^a $\overline{y_5}^p$ (Falsify y_5)

$\overline{x_2}^p$ x_7^p y_2^a y_3^a $\overline{y_5}^p$ (Conflict analysis)

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ y_2 y_3 $\cancel{y_4}$ y_5 y_6 y_7 y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Find a core (by falsified soft literal):

x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p y_2^a y_3^a $\overline{y_5}^p$ (Falsify y_5)

$\overline{x_2}^p$ x_7^p y_2^a y_3^a $\overline{y_5}^p$ (Conflict analysis)

Found disjoint cores:

$\overline{x_2} \wedge x_7 \wedge y_2 \wedge y_3 \vdash_{UP} \overline{y_5}$

$\overline{x_2} \wedge x_7 \rightarrow \overline{y_2} \vee \overline{y_3} \vee \overline{y_5}$

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Found disjoint cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow \overline{y_2} \vee \overline{y_3} \vee \overline{y_5}$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow \overline{y_6} \vee \overline{y_7}$

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Found disjoint cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow \overline{y_2} \vee \overline{y_3} \vee \overline{y_5}$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow \overline{y_6} \vee \overline{y_7}$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (\overline{y}_1 \vee \overline{y}_4) \wedge (\overline{y_2} \vee \overline{y_3} \vee \overline{y_5}) \wedge (\overline{y_6} \vee \overline{y_7})$$

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Initial trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

Found disjoint cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow \overline{y_2} \vee \overline{y_3} \vee \overline{y_5}$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow \overline{y_6} \vee \overline{y_7}$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (\overline{y}_1 \vee \overline{y}_4) \wedge (\overline{y_2} \vee \overline{y_3} \vee \overline{y_5}) \wedge (\overline{y_6} \vee \overline{y_7})$

$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow LB = 3 = UB$

Soft conflict detection: full example

UB $= 3$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Initial trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

Found disjoint cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$

Core 2: $\overline{x_2} \wedge x_7 \rightarrow \overline{y_2} \vee \overline{y_3} \vee \overline{y_5}$

Core 3: $x_1 \wedge \overline{x_4} \wedge x_7 \rightarrow \overline{y_6} \vee \overline{y_7}$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (\overline{y}_1 \vee \overline{y}_4) \wedge (\overline{y_2} \vee \overline{y_3} \vee \overline{y_5}) \wedge (\overline{y_6} \vee \overline{y_7})$$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow LB = 3 = UB$$

Soft conflict:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$, Conflict $\overline{x_1} \vee x_2 \vee x_4 \vee \overline{x_7}$ (**soft conflict**)

Hardening

When $LB = UB - 1$, all unassigned soft literals not in cores must be true.

UB $= 4$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

$$x_1 \land \overline{x_2} \land \overline{x_4} \land x_7 \to (\overline{y}_1 \lor \overline{y}_4) \land (\overline{y_2} \lor \overline{y_3} \lor \overline{y_5}) \land (\overline{y_6} \lor \overline{y_7})$$

Hardening

When $LB = UB - 1$, all unassigned soft literals not in cores must be true.

UB $= 4$ **Soft literals:** ~~y_1~~ ~~y_2~~ ~~y_3~~ ~~y_4~~ ~~y_5~~ ~~y_6~~ ~~y_7~~ y_8

Trail: x_1^d $\overline{x_2}^p$ x_3^p $\overline{x_4}^d$ x_5^p x_6^p x_7^p

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (\overline{y}_1 \vee \overline{y}_4) \wedge (\overline{y_2} \vee \overline{y_3} \vee \overline{y_5}) \wedge (\overline{y_6} \vee \overline{y_7})$$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \wedge \overline{y_8} \rightarrow LB = 4 = UB$$

Hardening

When $LB = UB - 1$, all unassigned soft literals not in cores must be true.

UB $= 4$ **Soft literals:** $\not{y_1}$ $\not{y_2}$ $\not{y_3}$ $\not{y_4}$ $\not{y_5}$ $\not{y_6}$ $\not{y_7}$ y_8

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow (\overline{y}_1 \vee \overline{y}_4) \wedge (\overline{y_2} \vee \overline{y_3} \vee \overline{y_5}) \wedge (\overline{y_6} \vee \overline{y_7})$$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \wedge \overline{y_8} \rightarrow LB = 4 = UB$$

$$x_1 \wedge \overline{x_2} \wedge \overline{x_4} \wedge x_7 \rightarrow y_8 \equiv \overline{x_1} \vee x_2 \vee x_4 \vee \overline{x_7} \vee y_8$$

Hardening

When $LB = UB - 1$, all unassigned soft literals not in cores must be true.

UB $= 4$ **Soft literals:** $\cancel{y_1}$ $\cancel{y_2}$ $\cancel{y_3}$ $\cancel{y_4}$ $\cancel{y_5}$ $\cancel{y_6}$ $\cancel{y_7}$ y_8

Trail: $x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p$

$$x_1 \land \overline{x_2} \land \overline{x_4} \land x_7 \rightarrow (\overline{y}_1 \lor \overline{y}_4) \land (\overline{y_2} \lor \overline{y_3} \lor \overline{y_5}) \land (\overline{y_6} \lor \overline{y_7})$$

$$x_1 \land \overline{x_2} \land \overline{x_4} \land x_7 \land \overline{y_8} \rightarrow LB = 4 = UB$$

$$x_1 \land \overline{x_2} \land \overline{x_4} \land x_7 \rightarrow y_8 \equiv \overline{x_1} \lor x_2 \lor x_4 \lor \overline{x_7} \lor y_8$$

Harden by UP:

$x_1^d\ \overline{x_2}^p\ x_3^p\ \overline{x_4}^d\ x_5^p\ x_6^p\ x_7^p\ y_8^p$

CDCL invariant:

every assigned literal is decided (x^d) or unit-propagated (x^p)

Weighted MaxCDCL

Weighted MaxCDCL

- Soft literals are weighted
- Cores are weighted
- Each soft literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

UB $= 4$
Soft literals: $3y_1 \quad 2y_2 \quad 1y_3 \quad 1y_4 \quad 1y_5 \quad 2y_6 \quad 1y_7 \quad 2y_8$

Found "disjoint" cores

Weighted MaxCDCL

- Soft literals are weighted
- Cores are weighted
- Each soft literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

UB $= 4$

Soft literals: $\not{3}\ 1y_1 \quad \not{2}\ 0y_2 \quad 1y_3 \quad 1y_4 \quad 1y_5 \quad 2y_6 \quad 1y_7 \quad 2y_8$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$ (2)

Weighted MaxCDCL

Weighted MaxCDCL

- Soft literals are weighted
- Cores are weighted
- Each soft literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

UB $= 4$

Soft literals: $\cancel{3}\ \cancel{1}\ {\color{red}0} y_1 \quad \cancel{2}\ {\color{red}0} y_2 \quad 1 y_3 \quad 1 y_4 \quad \cancel{1}\ {\color{red}0} y_5 \quad 2 y_6 \quad 1 y_7 \quad 2 y_8$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$ (2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_5$ (1)

Weighted MaxCDCL

Weighted MaxCDCL

- Soft literals are weighted
- Cores are weighted
- Each soft literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight

UB $= 4$

Soft literals: $\not{3}\ \not{1}\ 0 y_1 \quad \not{2}\ 0 y_2 \quad 1 y_3 \quad 1 y_4 \quad \not{1}\ 0 y_5 \quad \not{2}\ 0 y_6 \quad 1 y_7 \quad \not{2}\ 0 y_8$

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$ (2)

Core 2: $x_3 \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_5$ (1)

Core 3: $x_1 \rightarrow \overline{y}_6 \vee \overline{y}_8$ (2)

Weighted MaxCDCL

Weighted MaxCDCL

- Soft literals are weighted
- Cores are weighted
- Each soft literal can contribute to many cores
- The total contribution of a literal cannot exceed its weight
- Some cores can be removed in a postprocess

UB $= 4$

Soft literals: ~~3~~ ~~1~~ 0y_1 ~~2~~ 0y_2 1y_3 1y_4 ~~1~~ 0y_5 ~~2~~ 0y_6 1y_7 ~~2~~ 0y_8

Found "disjoint" cores

Core 1: $\overline{x_2} \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_4$ (2)

~~Core 2: $x_3 \wedge \overline{x_4} \rightarrow \overline{y}_1 \vee \overline{y}_5$ (1)~~

Core 3: $x_1 \rightarrow \overline{y}_6 \vee \overline{y}_8$ (2)

Weighted MaxCDCL

- Hardening happens with different $LB \leq UB$ (no necessarily $LB = UB$)

$$\textbf{unweighted}: \quad {\color{red}x_1} + {\color{red}x_2} + {\color{red}x_3} + x_4 + x_5 + x_6 < 4$$

$$\textbf{weighted}: \quad {\color{red}x_1} + {\color{red}x_2} + {\color{red}x_3} + x_4 + x_5 + 2x_6 < 5$$

- UP and hardening feed back each other

Effective bounding

Lookahead potential drawbacks:

- Running lookahead at each node is time-consuming
- Not every lookahead detects a soft conflict

Our bounding procedure:

- With probability 0.01, run lookahead for probing
  - Record average and standard deviation of LB increase in successful lookaheads
- Let $k = UB - cost(\rho)$: extra cost needed for soft conflict
- Do not run lookahead when $k > avg + coef \cdot stddev$
- We dynamically adjust *coef* to get success rate between 0.6 and 0.75

Results: experimental setting

Considered sets:

- **MSE19∪20:** 1000 instances.
  Union of all instances from MaxSAT Evaluations 2019 and 2020
- **MC (Master Collection):** 3614 instances.
  Subset of the master collection of instances from Evaluations, $\leq$ 100 instances per family

MaxSAT evaluations cutoff: 3600 seconds

Results: analysis of components

Table: Comparison of MaxCDCL with its variants

	MSE19∪20		MC	
	#solv	avg	#solv	avg
MaxCDCL without lookahead	505	255s	2183	194s
MaxCDCL without hardening	664	281s	2878	194s
MaxCDCL always lookahead	681	249s	2962	193s
MaxCDCL lexicographic order lookahead	704	268s	2963	168s
MaxCDCL	**734**	256s	**3022**	156s

Other implementation details:

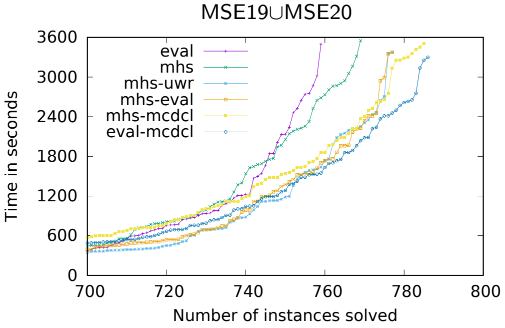
- Lookahead assumptions: prioritize literals recently involved in cores

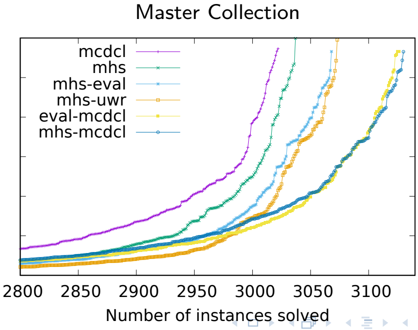
Results: pairwise-combinations of solvers

Table: Combined execution of two solvers (1800 seconds for each)

	mhs	eval	uwr	rc2	owbo	mcdcl
mhs	747	777	777	770	763	**785**
eval	777	745	760	751	760	**786**
uwr	777	760	730	745	746	774
rc2	770	751	745	713	745	**778**
owbo	763	760	746	745	675	746
mcdcl	**785**	**786**	774	**778**	746	711
3600s	769	759	745	728	695	734

mhs	eval	uwr	rc2	owbo	mcdcl
3009	3068	3073	3056	3049	**3130**
3068	2972	3019	2986	3013	**3126**
3073	3019	2951	3000	2998	**3098**
3056	2986	3000	2921	2981	**3105**
3049	3013	2998	2981	2865	**3076**
3130	**3126**	**3098**	**3105**	**3076**	2992
3037	3002	2969	2948	2906	3022





MSE22 Exact track, unweighted (left), weighted (right)

Solver	#Solved	Time(avg)
CASHWMaxSAT-CorePlus	438	285.59
CASHWMaxSAT-Plus	433	293.92
UWrMaxSat-SCIP	432	308.53
MaxHS	430	284.05
EvalMaxSAT	424	230.59
MaxCDCL	415	240.15
UWrMaxSat	407	185.77
CGSS	404	264.29
Open-WBO-Mergesat	377	211.7
Open-WBO-Glucose	375	158.8
Exact	296	246.45

Solver	#Solved	Time(avg)
CASHWMaxSAT-CorePlus	438	304.82
CASHWMaxSAT-Plus	435	303.16
UWrMaxSat-SCIP	427	305.7
MaxHS	426	349.93
WMaxCDCL	422	392.01
WMaxCDCL-bandall	416	366.43
UWrMaxSat	407	228.15
EvalMaxSAT	395	246.23
CGSS	391	264.22
Exact	310	249.43

MSE 2023

MSE23 Exact track, unweighted (left), weighted (right)

Solver	#Solved	Time(avg)
EvalMaxSAT-SCIP	433	326.11
MaxCDCL-S6-HS9	430	244.12
MaxCDCL-S6-HS12	430	252.4
CASHWMAXSAT-CorePlus	428	401.84
WMaxCDCL-S6-HS12	427	247.12
WMaxCDCL-S9-HS9	427	280.45
CASHWMAXSAT-CorePlus-m	427	403.8
EvalMaxSAT	425	239.75
CGSS2-SCIP	409	193.65
MaxCDCL	406	135.26
WMaxCDCL	399	160.24
CGSS2	394	107.32
Open-WBO-RES	387	172.26
Open-WBO-OLL	366	143.26

Solver	#Solved	Time(avg)
WMaxCDCL-S6-HS12	446	276.74
WMaxCDCL-S9-HS9	446	342.86
EvalMaxSAT-SCIP	442	405.04
CASHWMAXSAT-CorePlus	432	128.04
CASHWMAXSAT-CorePlus-m	431	402.39
EvalMaxSAT	429	330.39
CGSS2-SCIP	420	203.89
CGSS2	419	97.6
WMaxCDCL	389	120.85
Pacose	372	255.94
Pacose-MaxPre2	350	185.71

Beyond disjoint local cores

Disjoint cores:

- Literals can only belong to one core (accepting weight splitting)
- LB easily computable
- Every round of assumptions increases the LB

Beyond disjoint local cores

Disjoint cores:

- Literals can only belong to one core (accepting weight splitting)
- LB easily computable
- Every round of assumptions increases the LB

Non-disjoint:

- Avoid repeated cores when making assumptions
- LB computation is a hitting set problem, NP-Hard
- No guarantee that a round of assumptions increases the LB

Beyond disjoint local cores

Disjoint cores:

- Literals can only belong to one core (accepting weight splitting)
- LB easily computable
- Every round of assumptions increases the LB

Non-disjoint:

- Avoid repeated cores when making assumptions
- LB computation is a hitting set problem, NP-Hard
- No guarantee that a round of assumptions increases the LB

Disjoint cardinality constraints:

- Literals can only belong to one cardinality constraint, must be **unlocked** before adding
- LB easily computable
- Every round of assumptions increases the LB

Literal unlocking

Trail: ...

Initial disjoint cardinality constraints

Core 1: $R_1 \rightarrow \overline{y}_1 \vee \overline{y}_2 \vee \overline{y}_3$

Literal unlocking

Trail: . . .

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

Literal unlocking

Trail: $\ldots y_4^a$

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

Literal unlocking

Trail: $\ldots y_4^a \; \overline{y}_1^p$

Initial disjoint cardinality constraints
Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1

Literal unlocking

Trail: $\ldots y_4^a \; \overline{y}_1^p \; y_2^a$

Initial disjoint cardinality constraints
Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1

Literal unlocking

Trail: $\ldots y_4^a\ \overline{y}_1^p\ y_2^a\ y_3^a$ Conflict

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1

Core 2: $\overline{y}_4 + \overline{y}_2 + \overline{y}_3 \geq 1$

Literal unlocking

Trail: $\dots y_4^a \; \overline{y}_1^p \; y_2^a \; y_3^a$ Conflict

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1

Core 2: $\overline{y}_4 + \overline{y}_2 + \overline{y}_3 \geq 1$

Final disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 + \overline{y}_4 \geq 2$

Literal unlocking

Trail: $\ldots y_4^a\ \overline{y}_1^p\ y_2^a\ y_3^a$ Conflict

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1: $\overline{y}_4 + \overline{y}_1 \geq 1$

Core 2: $\overline{y}_4 + \overline{y}_2 + \overline{y}_3 \geq 1$

Final disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 + \overline{y}_4 \geq 2$

Literal unlocking

Trail: $\ldots y_4^a\ \overline{y}_1^p\ y_2^a\ y_3^a$ Conflict

Initial disjoint cardinality constraints

Core 1: $R_1 \rightarrow \overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 1$

$\overline{y}_1$ unlocks y_2 and y_3 from Core 1: $R_u \rightarrow \overline{y}_4 + \overline{y}_1 \geq 1$

Core 2: $R_2 \rightarrow \overline{y}_4 + \overline{y}_2 + \overline{y}_3 \geq 1$

Final disjoint cardinality constraints

Core 1: $R_1 \wedge R_2 \wedge R_u \rightarrow \overline{y}_1 + \overline{y}_2 + \overline{y}_3 + \overline{y}_4 \geq 2$

Literal unlocking

Trail: $y_7^a \; y_8^a \; \overline{y}_1^p \; \overline{y}_2^p \; y_3^a \; \overline{y}_4^p \; \overline{y}_5^p \; y_6^a$ Conflict

Initial disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 \geq 2$

Core 2: $\overline{y}_4 + \overline{y}_5 + \overline{y}_6 \geq 2$

Unlockings:

$\overline{y}_7 + \overline{y}_8 + \overline{y}_1 \geq 1$

$\overline{y}_7 + \overline{y}_8 + \overline{y}_2 \geq 1$

$\overline{y}_7 + \overline{y}_8 + \overline{y}_3 + \overline{y}_4 \geq 1$

$\overline{y}_7 + \overline{y}_8 + \overline{y}_3 + \overline{y}_5 \geq 1$

Core 3: $\overline{y}_7 + \overline{y}_8 + \overline{y}_6 \geq 1$

Final disjoint cardinality constraints

Core 1: $\overline{y}_1 + \overline{y}_2 + \overline{y}_3 + \overline{y}_4 + \overline{y}_5 + \overline{y}_6 + \overline{y}_7 + \overline{y}_8 \geq 5$

Literal unlocking: general case

A cardinality $C_i \geq K_i$ is unlocked if K_i of its literals are propagated to false.

Initial disjoint cardinality constraints

$R_1 \rightarrow C_1 \geq K_1, \ldots, R_n \rightarrow C_n \geq K_n$

Unlockings:

$R_{1,1} \rightarrow C_{1,1} \geq 1, \ldots, R_{1,K_1} \rightarrow C_{1,K_1} \geq 1$

...

$R_{n,1} \rightarrow C_{n,1} \geq 1, \ldots, R_{n,K_n} \rightarrow C_{n,K_n} \geq 1$

New constraint: $R_{n+1} \rightarrow C_{n+1} \geq 1$

Final disjoint cardinality constraints

The union of all reasons $\rightarrow$ the sum of all soft literals $\geq K_1 + \cdots + K_n + 1$

Conclusions and future work

- MaxCDCL: combining BnB and clause learning
- Potential to solve new kinds of instances
- Somehow, shares features from model-based, core-based and implicit hitting set solvers
- Can we achieve a smart combination of these techniques?
- Can we get stronger but still fast lookahead reasoning?

References

André Abramé and Djamal Habet
Ahmaxsat: Description and evaluation of a branch and bound Max-SAT solver.
Journal on Satisfiability, Boolean Modeling and Computation, 9:89–128, 2014

Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamal Habet, Kun He
Combining Clause Learning and Branch and Bound for MaxSAT
27th International Conference on Principles and Practice of Constraint Programming (CP 2021), pp. 38:1-38:18, 2021

Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamal Habet, Kun He
Boosting branch-and-bound MaxSAT solvers with clause learning
AI Communications 35(2): 131-151 (2022)

Branch-and-Bound MaxSAT Solving with MaxCDCL

Jordi Coll
Universitat de Girona

Djamal Habet, Kun He, Chu-Min Li, Shuolin Li, Felip Manyà, Zhenxing Xu

Lund, November 6, 2024