Non-standard Conflict analysis for solving the Job shop problem

Mohamed Siala, Christian Artigues, and Emmanuel Hebrard

ROC group : Recherche Opérationnelle/Optimisation Combinatoire/Contraintes

LAAS-CNRS

Bordeaux, France

Outline

Boolean Satisfiability (SAT)

Problem

- Boolean variables (atoms)
- Propositional logic formula (often CNF)
- Literals: $a, \overline{a}$
- Clauses: $(\overline{a} \vee \overline{f} \vee g)$, $(\overline{a} \vee \overline{f} \vee g)$, $(\overline{a} \vee \overline{b})$, $(b \vee \overline{c} \vee g)$

Boolean Satisfiability (SAT)

Problem

- Boolean variables (atoms)
- Propositional logic formula (often CNF)
- Literals: $a, \overline{a}$
- Clauses: $(\overline{a} \vee \overline{f} \vee g)$, $(\overline{a} \vee \overline{f} \vee g)$, $(\overline{a} \vee \overline{b})$, $(b \vee \overline{c} \vee g)$

SAT Solving

- DPLL : Backtracking in Tree Search + Unit Propagation
- Conflict-Driven Clause Learning (CDCL) : DPLL + Learning
- But also :
  - Adaptive branching heuristics (weight conflicting literals)
  - Restarts
  - Simplifications
  - Forget clauses
  - Incrementality
  - ...

Unit Propagation

Given a clause C of arity n. If $n - 1$ literals are false then set the last one to be true.

Example

$$(h \vee \overline{o} \vee \bar{j} \vee n)$$

Unit Propagation

Given a clause C of arity n. If $n-1$ literals are false then set the last one to be true.

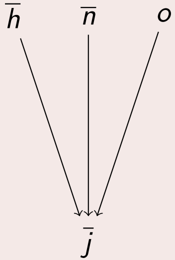
Example

$$(h \vee \overline{o} \vee \bar{j} \vee n)$$

Unit Propagation

Given a clause C of arity n. If $n-1$ literals are false then set the last one to be true.

Example



$$(h \vee \overline{o} \vee \overline{j} \vee n)$$
$$\equiv$$
$$(\overline{h} \wedge o \wedge \overline{n}) \rightarrow \overline{j}$$

Conflict-Driven Clause Learning

	f		

$\overline{a} \vee \overline{f} \vee g$

$\overline{a} \vee \overline{b} \vee \overline{h}$

$a \vee c$

$a \vee \overline{i} \vee \overline{l}$

$a \vee \overline{k} \vee \overline{j}$

$b \vee d$

$b \vee g \vee \overline{n}$

$b \vee \overline{f} \vee n \vee k$

$\overline{c} \vee k$

$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$

$c \vee l$

$d \vee \overline{k} \vee l$

$d \vee \overline{g} \vee l$

$\overline{g} \vee n \vee o$

$h \vee \overline{o} \vee \overline{j} \vee n$

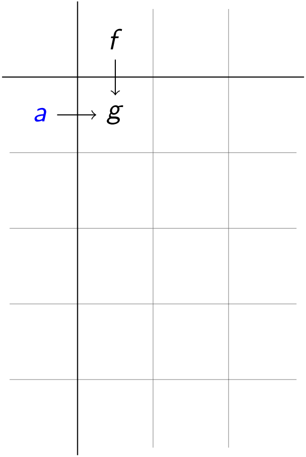
$\overline{i} \vee j$

$\overline{d} \vee \overline{l} \vee \overline{m}$

$\overline{e} \vee m \vee \overline{n}$

$\overline{f} \vee h \vee i$

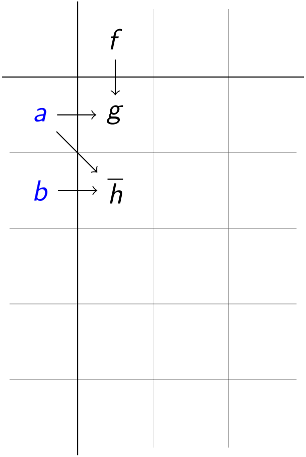
Conflict-Driven Clause Learning



$\overline{a} \vee \overline{f} \vee g$
$\overline{a} \vee \overline{b} \vee \overline{h}$
$a \vee c$
$a \vee \overline{i} \vee \overline{l}$
$a \vee \overline{k} \vee \overline{j}$
$b \vee d$
$b \vee g \vee \overline{n}$
$b \vee \overline{f} \vee n \vee k$
$\overline{c} \vee k$
$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$
$c \vee l$
$d \vee \overline{k} \vee l$
$d \vee \overline{g} \vee l$
$\overline{g} \vee n \vee o$
$h \vee \overline{o} \vee \overline{j} \vee n$
$\overline{i} \vee j$
$\overline{d} \vee \overline{l} \vee \overline{m}$
$\overline{e} \vee m \vee \overline{n}$
$\overline{f} \vee h \vee i$

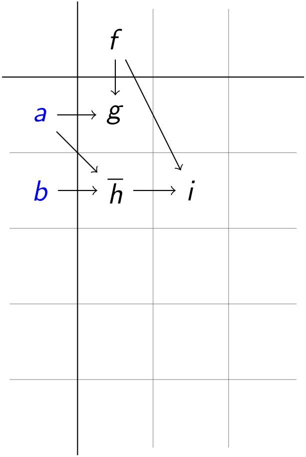
Conflict-Driven Clause Learning



$\overline{a} \vee \overline{f} \vee g$
$\overline{a} \vee \overline{b} \vee \overline{h}$
$a \vee c$
$a \vee \overline{i} \vee \overline{l}$
$a \vee \overline{k} \vee \overline{j}$
$b \vee d$
$b \vee g \vee \overline{n}$
$b \vee \overline{f} \vee n \vee k$
$\overline{c} \vee k$
$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$
$c \vee l$
$d \vee \overline{k} \vee l$
$d \vee \overline{g} \vee l$
$\overline{g} \vee n \vee o$
$h \vee \overline{o} \vee \overline{j} \vee n$
$\overline{i} \vee j$
$\overline{d} \vee \overline{l} \vee \overline{m}$
$\overline{e} \vee m \vee \overline{n}$
$\overline{f} \vee h \vee i$

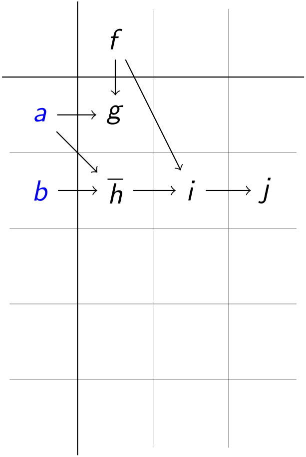
Conflict-Driven Clause Learning



$\bar{a} \vee \bar{f} \vee g$
$\bar{a} \vee \bar{b} \vee \bar{h}$
$a \vee c$
$a \vee \bar{i} \vee \bar{l}$
$a \vee \bar{k} \vee \bar{j}$
$b \vee d$
$b \vee g \vee \bar{n}$
$b \vee \bar{f} \vee n \vee k$
$\bar{c} \vee k$
$\bar{c} \vee \bar{k} \vee \bar{i} \vee l$

$c \vee h \vee n \vee \bar{m}$
$c \vee l$
$d \vee \bar{k} \vee l$
$d \vee \bar{g} \vee l$
$\bar{g} \vee n \vee o$
$h \vee \bar{o} \vee \bar{j} \vee n$
$\bar{i} \vee j$
$\bar{d} \vee \bar{l} \vee \bar{m}$
$\bar{e} \vee m \vee \bar{n}$
$\bar{f} \vee h \vee i$

Conflict-Driven Clause Learning



$\bar{a} \vee \bar{f} \vee g$

$\bar{a} \vee \bar{b} \vee \bar{h}$

$a \vee c$

$a \vee \bar{i} \vee \bar{l}$

$a \vee \bar{k} \vee \bar{j}$

$b \vee d$

$b \vee g \vee \bar{n}$

$b \vee \bar{f} \vee n \vee k$

$\bar{c} \vee k$

$\bar{c} \vee \bar{k} \vee \bar{i} \vee l$

$c \vee h \vee n \vee \bar{m}$

$c \vee l$

$d \vee \bar{k} \vee l$

$d \vee \bar{g} \vee l$

$\bar{g} \vee n \vee o$

$h \vee \bar{o} \vee \bar{j} \vee n$

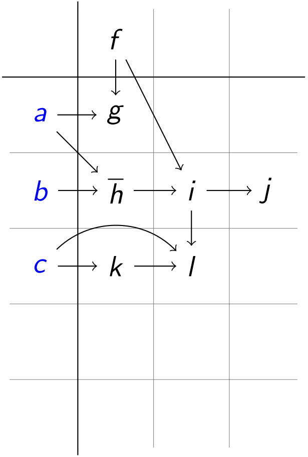
$\bar{i} \vee j$

$\bar{d} \vee \bar{l} \vee \bar{m}$

$\bar{e} \vee m \vee \bar{n}$

$\bar{f} \vee h \vee i$

Conflict-Driven Clause Learning



$\overline{a} \vee \overline{f} \vee g$

$\overline{a} \vee \overline{b} \vee \overline{h}$

$a \vee c$

$a \vee \overline{i} \vee \overline{l}$

$a \vee \overline{k} \vee \overline{j}$

$b \vee d$

$b \vee g \vee \overline{n}$

$b \vee \overline{f} \vee n \vee k$

$\overline{c} \vee k$

$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$

$c \vee l$

$d \vee \overline{k} \vee l$

$d \vee \overline{g} \vee l$

$\overline{g} \vee n \vee o$

$h \vee \overline{o} \vee \overline{j} \vee n$

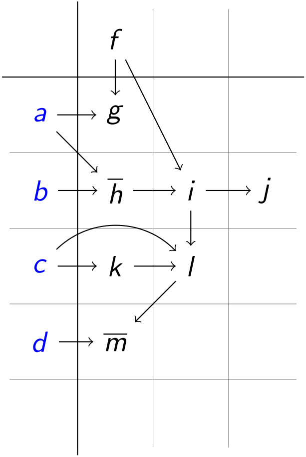
$\overline{i} \vee j$

$\overline{d} \vee \overline{l} \vee \overline{m}$

$\overline{e} \vee m \vee \overline{n}$

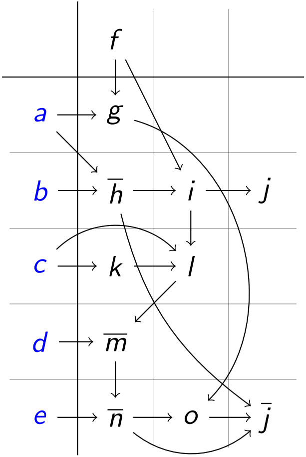
$\overline{f} \vee h \vee i$

Conflict-Driven Clause Learning



$\overline{a} \vee \overline{f} \vee g$
$\overline{a} \vee \overline{b} \vee \overline{h}$
$a \vee c$
$a \vee \overline{i} \vee \overline{l}$
$a \vee \overline{k} \vee \overline{j}$
$b \vee d$
$b \vee g \vee \overline{n}$
$b \vee \overline{f} \vee n \vee k$
$\overline{c} \vee k$
$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$
$c \vee h \vee n \vee \overline{m}$
$c \vee l$
$d \vee \overline{k} \vee l$
$d \vee \overline{g} \vee l$
$\overline{g} \vee n \vee o$
$h \vee \overline{o} \vee \overline{j} \vee n$
$\overline{i} \vee j$
$\overline{d} \vee \overline{l} \vee \overline{m}$
$\overline{e} \vee m \vee \overline{n}$
$\overline{f} \vee h \vee i$

Conflict-Driven Clause Learning



$\overline{a} \vee \overline{f} \vee g$

$\overline{a} \vee \overline{b} \vee \overline{h}$

$a \vee c$

$a \vee \overline{i} \vee \overline{l}$

$a \vee \overline{k} \vee \overline{j}$

$b \vee d$

$b \vee g \vee \overline{n}$

$b \vee \overline{f} \vee n \vee k$

$\overline{c} \vee k$

$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$

$c \vee l$

$d \vee \overline{k} \vee l$

$d \vee \overline{g} \vee l$

$\overline{g} \vee n \vee o$

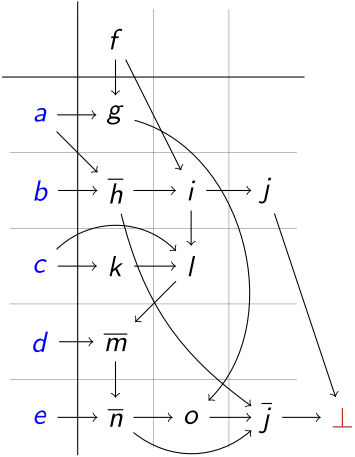
$h \vee \overline{o} \vee \overline{j} \vee n$

$\overline{i} \vee j$

$\overline{d} \vee \overline{l} \vee \overline{m}$

$\overline{e} \vee m \vee \overline{n}$

$\overline{f} \vee h \vee i$



$\overline{a} \vee \overline{f} \vee g$

$\overline{a} \vee \overline{b} \vee \overline{h}$

$a \vee c$

$a \vee \overline{i} \vee \overline{l}$

$a \vee \overline{k} \vee \overline{j}$

$b \vee d$

$b \vee g \vee \overline{n}$

$b \vee \overline{f} \vee n \vee k$

$\overline{c} \vee k$

$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$

$c \vee l$

$d \vee \overline{k} \vee l$

$d \vee \overline{g} \vee l$

$\overline{g} \vee n \vee o$

$h \vee \overline{o} \vee \overline{j} \vee n$

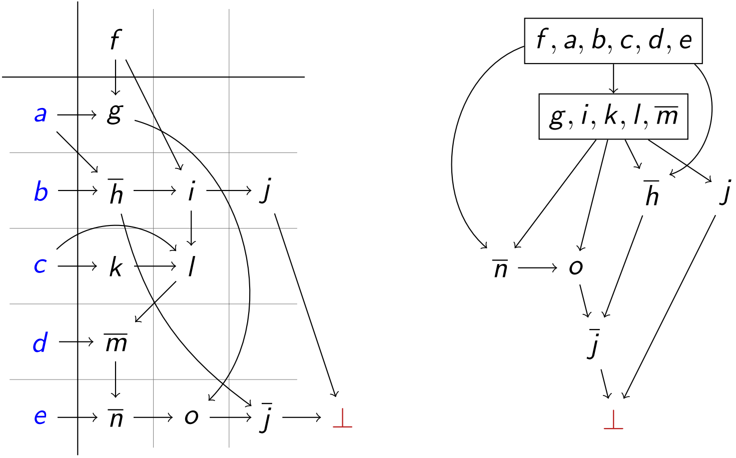
$\overline{i} \vee j$

$\overline{d} \vee \overline{l} \vee \overline{m}$

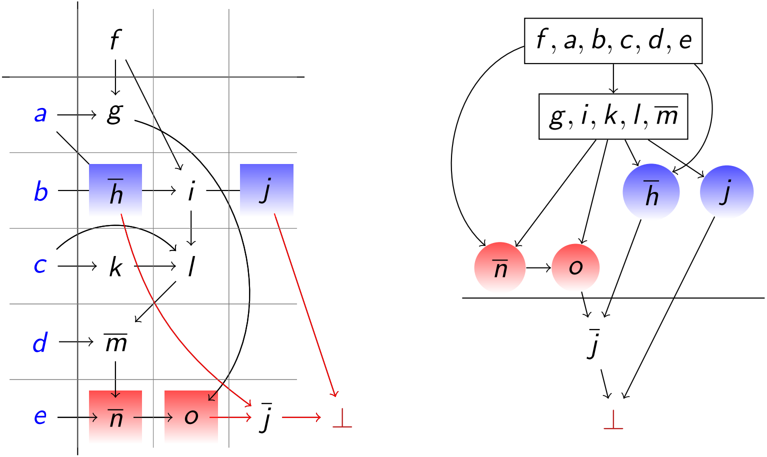
$\overline{e} \vee m \vee \overline{n}$

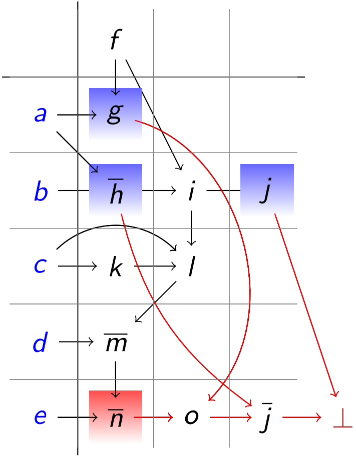
$\overline{f} \vee h \vee i$

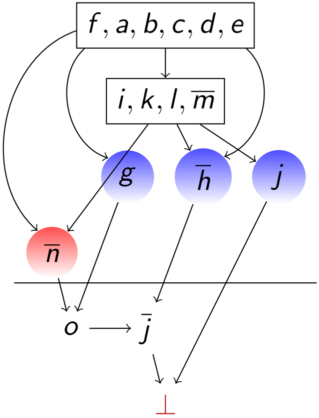
Conflict-Driven Clause Learning

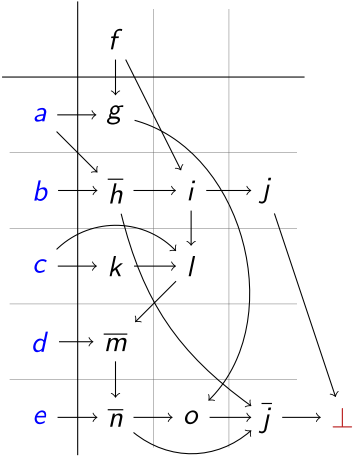


Conflict-Driven Clause Learning



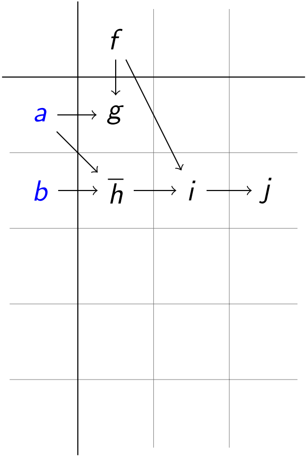

f
a
g
b
h̄
i
j
c
k
l
d
m̄
e
n̄
o
j̄
⊥


f, a, b, c, d, e
i, k, l, m̄
g
h̄
j
n̄
o
j̄
⊥



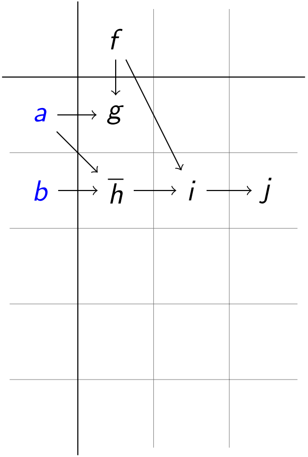
$\bar{a} \vee \bar{f} \vee g$
$\bar{a} \vee \bar{b} \vee \bar{h}$
$a \vee c$
$a \vee \bar{i} \vee \bar{l}$
$a \vee \bar{k} \vee \bar{j}$
$b \vee d$
$b \vee g \vee \bar{n}$
$b \vee \bar{f} \vee n \vee k$
$\bar{c} \vee k$
$\bar{c} \vee \bar{k} \vee \bar{i} \vee l$

$c \vee h \vee n \vee \bar{m}$
$c \vee l$
$d \vee \bar{k} \vee l$
$d \vee \bar{g} \vee l$
$\bar{g} \vee n \vee o$
$h \vee \bar{o} \vee \bar{j} \vee n$
$\bar{i} \vee j$
$\bar{d} \vee \bar{l} \vee \bar{m}$
$\bar{e} \vee m \vee \bar{n}$
$\bar{f} \vee h \vee i$



$\overline{a} \lor \overline{f} \lor g$
$\overline{a} \lor \overline{b} \lor \overline{h}$
$a \lor c$
$a \lor \overline{i} \lor \overline{l}$
$a \lor \overline{k} \lor \overline{j}$
$b \lor d$
$b \lor g \lor \overline{n}$
$b \lor \overline{f} \lor n \lor k$
$\overline{c} \lor k$
$\overline{c} \lor \overline{k} \lor \overline{i} \lor l$

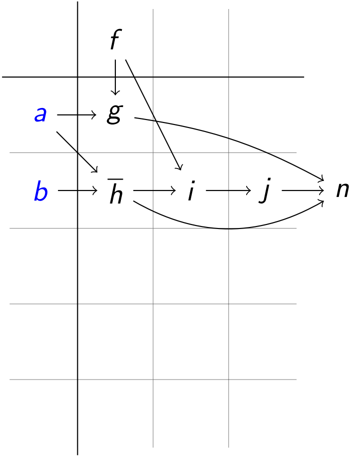
$c \lor h \lor n \lor \overline{m}$
$c \lor l$
$d \lor \overline{k} \lor l$
$d \lor \overline{g} \lor l$
$\overline{g} \lor n \lor o$
$h \lor \overline{o} \lor \overline{j} \lor n$
$\overline{i} \lor j$
$\overline{d} \lor \overline{l} \lor \overline{m}$
$\overline{e} \lor m \lor \overline{n}$
$\overline{f} \lor h \lor i$



$\overline{a} \vee \overline{f} \vee g$
$\overline{a} \vee \overline{b} \vee h$
$a \vee c$
$a \vee \overline{i} \vee \overline{l}$
$a \vee \overline{k} \vee \overline{j}$
$b \vee d$
$b \vee g \vee \overline{n}$
$b \vee \overline{f} \vee n \vee k$
$\overline{c} \vee k$
$\overline{c} \vee \overline{k} \vee \overline{i} \vee l$

$c \vee h \vee n \vee \overline{m}$
$c \vee l$
$d \vee \overline{k} \vee l$
$d \vee \overline{g} \vee l$
$\overline{g} \vee n \vee o$
$h \vee \overline{o} \vee \overline{j} \vee n$
$\overline{i} \vee j$
$\overline{d} \vee \overline{l} \vee \overline{m}$
$\overline{e} \vee m \vee \overline{n}$
$\overline{f} \vee h \vee i$
$\boxed{\overline{g} \vee h \vee \overline{j} \vee n}$

Conflict-Driven Clause Learning



$\bar a \vee \bar f \vee g$
$\bar a \vee \bar b \vee \bar h$
$a \vee c$
$a \vee \bar i \vee \bar l$
$a \vee \bar k \vee \bar j$
$b \vee d$
$b \vee g \vee \bar n$
$b \vee \bar f \vee n \vee k$
$\bar c \vee k$
$\bar c \vee \bar k \vee \bar i \vee l$

$c \vee h \vee n \vee \bar m$
$c \vee l$
$d \vee \bar k \vee l$
$d \vee \bar g \vee l$
$\bar g \vee n \vee o$
$h \vee \bar o \vee \bar j \vee n$
$\bar i \vee j$
$\bar d \vee \bar l \vee \bar m$
$\bar e \vee m \vee \bar n$
$\bar f \vee h \vee i$
$\bar g \vee h \vee \bar j \vee n$

SAT & CP :

- Can we get the best from both approaches?
- to encode into SAT or to use global constraints?
  $\rightarrow$A key concept in hybrid solvers : Explanations

An explanation is a set of atomic constraints triggering a failure/filtering.

example

Cardinality Constraint : $\sum_{i=1}^{n} x_i \leq k$; $D(x_i) = \{0, 1\}$.
$x_i \leftarrow 1$ is pruned if we already have k appearances of the value 1.

$$\{x_j \leftarrow 1 | D(x_j) = \{1\}\} \rightarrow x_i \nleftarrow 1 .$$

SAT & CP :

- Can we get the best from both approaches?
- to encode into SAT or to use global constraints?
  $\rightarrow$A key concept in hybrid solvers : Explanations

An explanation is a set of atomic constraints triggering a failure/filtering.

example

Cardinality Constraint : $\sum_{i=1}^{n} x_i \leq k$; $D(x_i) = \{0, 1\}$.
$x_i \leftarrow 1$ is pruned if we already have k appearances of the value 1.

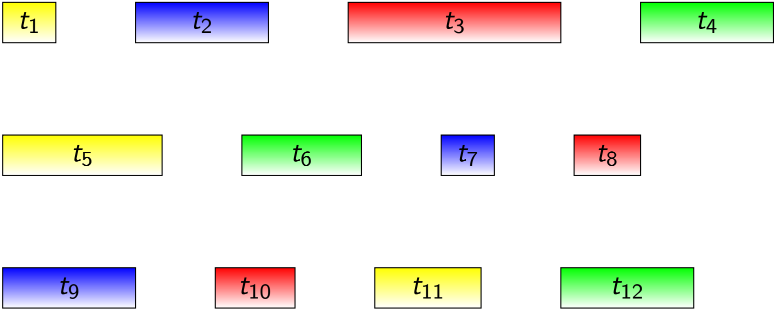
$$\{x_j \leftarrow 1 | D(x_j) = \{1\}\} \quad \rightarrow x_i \not\leftarrow 1 \ .$$

Encoding the Domains

Given a variable X, s.t. $D(X) = \{v_1, v_2, ..v_n\}$

- The Direct Encoding : $[\![X = v_i]\!]$
  - $[\![X = v_1]\!] \vee [\![X = v_2]\!].. \vee [\![X = v_n]\!]$
  - $\neg[\![X = v_1]\!] \vee \neg[\![X = v_2]\!]$
  - $\neg[\![X = v_1]\!] \vee \neg[\![X = v_3]\!]$
  - ..
- The order Encoding : $[\![X \leq v_i]\!]$
  - $\neg[\![X \leq v_1]\!] \vee [\![X \leq v_2]\!]$
  - $\neg[\![X \leq v_3]\!] \vee [\![X \leq v_3]\!]$
  - .. $\neg[\![X \leq v_{n-1}]\!] \vee [\![X \leq v_n]\!]$

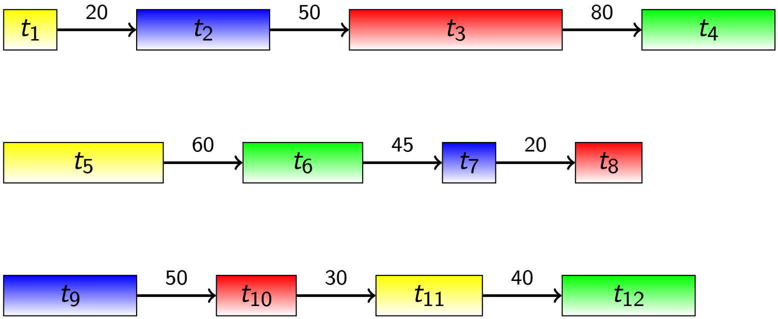
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks

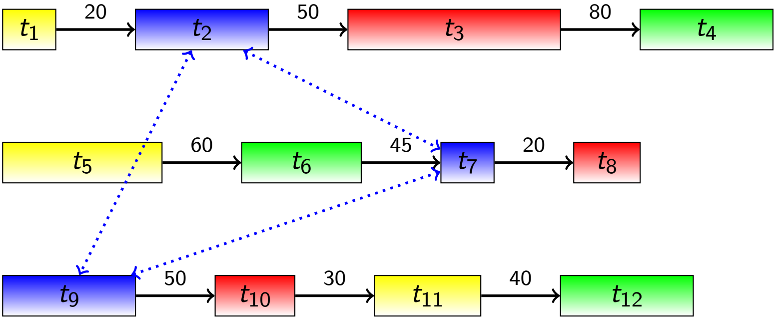
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)

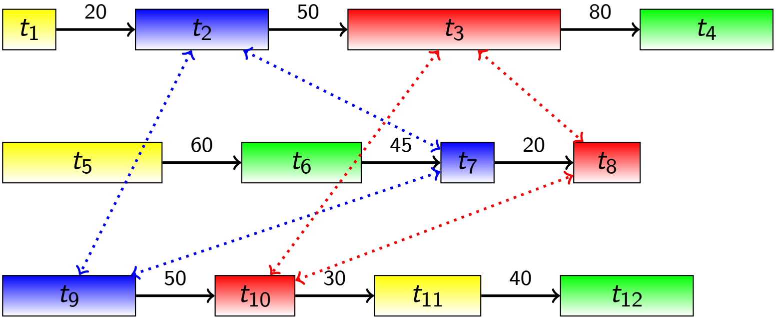
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)
- Requiring one of *m* disjunctive resources

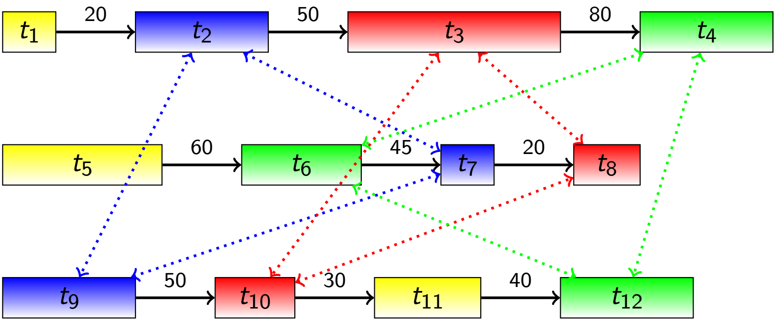
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)
- Requiring one of m disjunctive resources

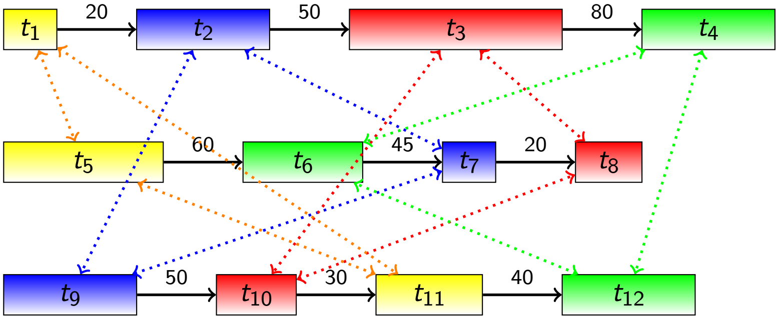
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)
- Requiring one of m disjunctive resources

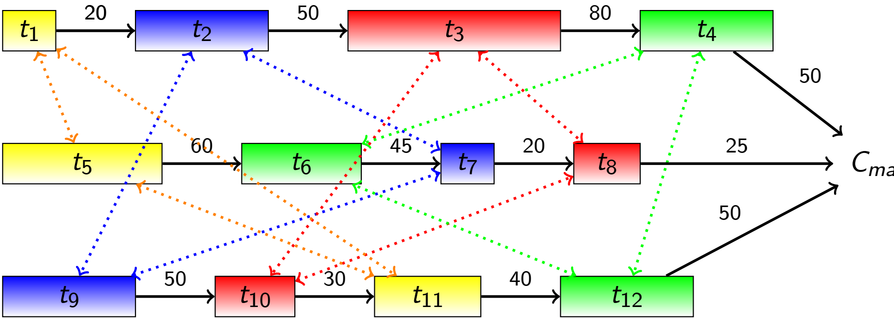
Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)
- Requiring one of *m* disjunctive resources

Job shop Scheduling Problem



Problem description

- A set of non-preemptive tasks
- Organized in jobs (sequences)
- Requiring one of m disjunctive resources
- Objective: minimize the total duration (C_{max})

A basic CP Model

- Precedence constraints: $t_i + p_i \leq t_{i+1}$.
- Binary Disjunctive constraints: $b_{ij} = \begin{cases} 0 \Leftrightarrow t_i + p_i \leq t_j \\ 1 \Leftrightarrow t_j + p_j \leq t_i \end{cases}$

Explaining $X + p \leq Y$

Failure

$$[\![X \geq l_X]\!] \wedge [\![Y \leq u_Y]\!] \rightarrow \perp$$

Pruning

$$[\![Y \leq u_X + p]\!] \rightarrow [\![X \leq u_X]\!]$$
$$[\![X \geq l_Y - p]\!] \rightarrow [\![Y \geq l_Y]\!]$$

Explaining the disjunctive Constraint

$$b = \begin{cases} 1 & \Leftrightarrow & X + p \leq Y \\ 0 & \Leftrightarrow & Y + p' \leq X \end{cases} \tag{3.1}$$

Failure

$$b \wedge [\![X \geq l_X]\!] \wedge [\![Y \leq u_Y]\!] \rightarrow \perp \text{ or}$$
$$\neg b \wedge [\![Y \geq l_Y]\!] \wedge [\![X \leq u_X]\!] \rightarrow \perp$$

Pruning

$$b \wedge [\![Y \leq u_X + p]\!] \rightarrow [\![X \leq u_X]\!]$$
$$b \wedge [\![X \geq l_Y - p]\!] \rightarrow [\![Y \geq l_Y]\!]$$
$$\neg b \wedge [\![X \leq u_Y + p']\!] \rightarrow [\![Y \leq u_Y]\!]$$
$$\neg b \wedge [\![Y \geq l_X - p']\!] \rightarrow [\![X \geq l_X]\!]$$

$$[\![Y \geq l_Y]\!] \wedge [\![X \leq u_X]\!] \rightarrow [\![b = 1]\!]$$
$$[\![X \geq l_X]\!] \wedge [\![Y \leq u_Y]\!] \rightarrow [\![b = 0]\!]$$

Dealing with large domains

Suppose $|D(X)| = 10^6$

- Generate 10^6 atoms only to encode the domain of X
- Check 10^6 clauses just for the consistency of the domain of X

$\rightarrow$ Solution : Lazy generation

$\rightarrow$ Much better ! but not that good!

- Redundancy of clauses : suppose that $a < b < c$ and $[\![X \leq a]\!], [\![X \leq c]\!]$, and $\neg[\![X \leq a]\!] \vee [\![X \leq c]\!]$ are already generated. If $[\![X \leq b]\!]$ needed to be generated, then add
  - $\neg[\![X \leq a]\!] \vee [\![X \leq b]\!]$
  - $\neg[\![X \leq b]\!] \vee [\![X \leq c]\!]$

  $\rightarrow \neg[\![X \leq a]\!] \vee [\![X \leq c]\!]$ becomes redundant
- Possibly we will end up with a large number of generated atoms

Some observations

- Deciding the set of boolean variables is sufficient to decide the problem!
- What about learning nogoods defined only over these variables?
- What to do to bound assignments $[\![X \leq v]\!]$ coming from explanations?

Virtual Literals

- Literals of the form $[\![X \leq v]\!]$ are no longer represented by integers
- We propose a data structure called 'Virtual Literal' containing these fields :
  - is_a_bound_literal : bool
  - is_a_lower_bound : bool
  - value : integer
  - id_variable : integer
- for instance with a 64bit integer encoding, one can do the following :

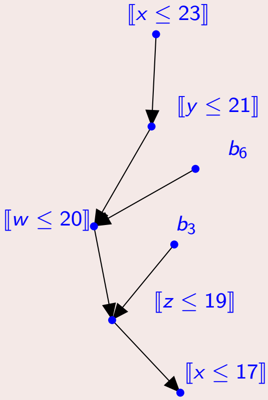
1/0	1/0		
is_a_bound_literal	is_a_lower_bound	value (32) bits	id_variable (30 bits)

Conflict Analysis

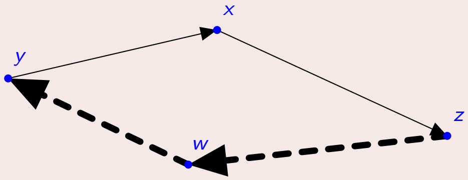
- Backward explanations : Whenever a domain change occurs during propagation, do not generate a clause! Just record the constraint triggering that propagation.
- During conflict analysis, we ask constraints to explain themselves.
- Whenever a bound literal occurs, replace it with its reason.

- Suppose that during conflict analysis a (virtual) literal $[\![X \leq 17]\!]$ has already been explored and afterwards a (virtual) literal $[\![X \leq 23]\!]$ occurs.
- One can consider $[\![X \leq 17]\!]$ as a valid explanation for $[\![X \leq 23]\!]$ since $[\![X \leq 17]\!] \rightarrow [\![X \leq 23]\!]$ is always correct.
- $\rightarrow$ Do not explore $[\![X \leq 23]\!]$, just drop it.

Suppose that when exploring $[\![X \leq 17]\!]$, $[\![X \leq 23]\!]$ occurs.



Improvements : Cycles



lemme

If in the predecessors of an upper (lower) bound literal $[\![X \leq v]\!]$ ($\neg[\![X \leq v]\!]$) there exists a literal of the form $[\![X \leq v']\!]$ ($\neg[\![X \leq v']\!]$) then a 'precedence' cycle has appeared in the constraint graph and has at least two disjunctions.

Theorem

The set of disjunctions appearing in such cycle is a valid nogood.

Conclusion

- A new method to tackle large domain SAT encoding
- A perfect framework to exploit lazy explanations
- Several improvements are being proposed
- Learning from cycles opens new learning perspectives but should be carefully studied as it doesn't garantee UIP.

Thank you!