Two Clause Learning Approaches for Disjunctive Scheduling

Mohamed Siala, Christian Artigues, and Emmanuel Hebrard

Context

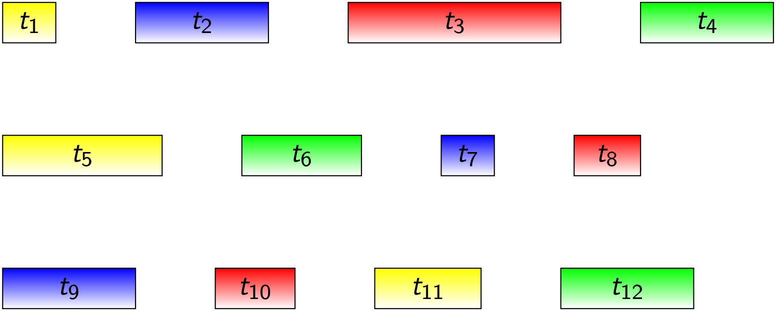
Disjunctive Scheduling

A family of scheduling problems having in common the `Unary Resource Constraint`.

Context

Disjunctive Scheduling

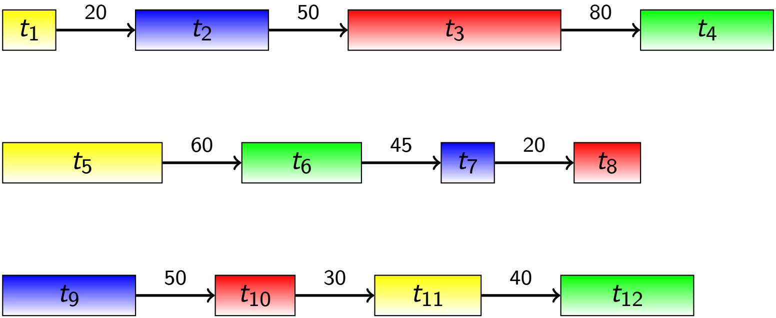
A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Disjunctive Scheduling

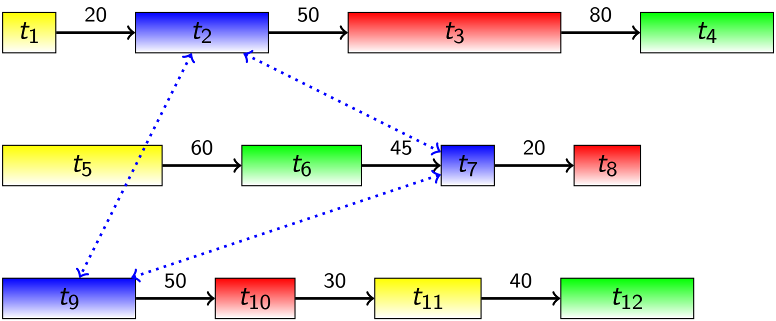
A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Disjunctive Scheduling

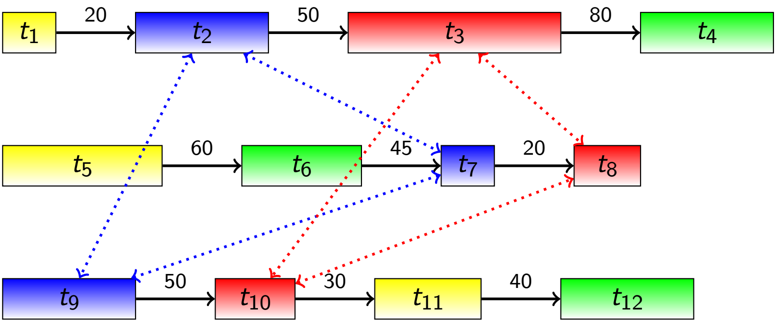
A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Disjunctive Scheduling

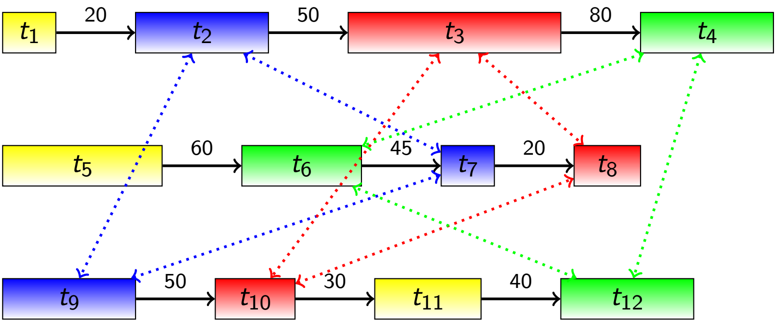
A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Disjunctive Scheduling

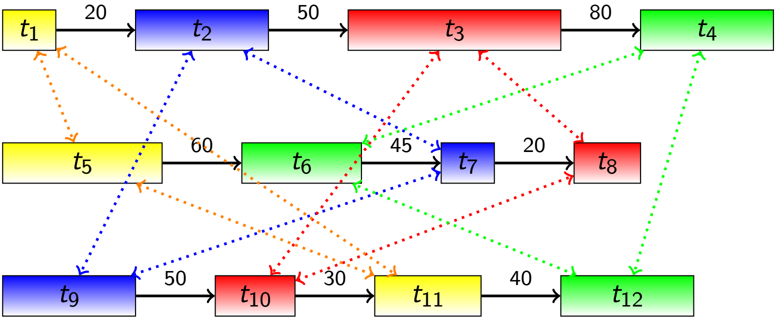
A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Disjunctive Scheduling

A family of scheduling problems having in common the `Unary Resource Constraint`.



Context

Scheduling in CP

- Tradition
  - Tailored propagation algorithms (such as `Edge-Finding` [Carlier and Pinson, 1989]
  - Tailored search strategies (such as `Texture` [Sadeh, 1991]).

Context

Scheduling in CP

- Tradition
  - Tailored propagation algorithms (such as `Edge-Finding` [Carlier and Pinson, 1989]
  - Tailored search strategies (such as `Texture` [Sadeh, 1991]).
- New trend: Focus on what can be learnt during search
  - Lazy Clause Generation for `RCPSP` [Schutt et al., 2013].
  - Weight-based heuristic learning on disjunctive scheduling [Grimes and Hebrard, 2015].

Example

$$x_1 + x_7 \geq 4 \wedge$$
$$x_2 + x_{10} \geq 11 \wedge$$
$$x_3 + x_9 = 16 \wedge$$
$$x_5 \geq x_8 + x_9 \wedge$$
$$b \leftrightarrow (x_9 - x_4 = 14) \wedge$$
$$b \rightarrow (x_6 \geq 7) \wedge$$
$$b \rightarrow (x_6 + x_7 \leq 9) \wedge$$
$$x_{11} \geq x_9 + x_{10}$$

b	0	1
x_1	1	30
x_2	9	30
x_3	0	3
x_4	0	30
x_5	24	45
x_6	5	10
x_7	2	10
x_8	9	30
x_9	13	16
x_9	13	16
x_{10}	0	3
x_{11}	15	25

Example

$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$

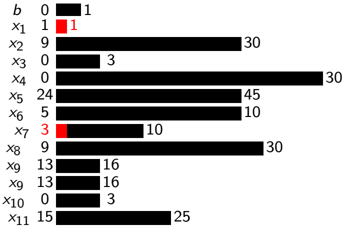
$[\![x_1 = 1]\!]$

b	0	1
x_1	1	1
x_2	9	30
x_3	0	3
x_4	0	30
x_5	24	45
x_6	5	10
x_7	2	10
x_8	9	30
x_9	13	16
x_9	13	16
x_{10}	0	3
x_{11}	15	25

Example

$$
\begin{aligned}
&x_1 + x_7 \geq 4 \wedge \\
&x_2 + x_{10} \geq 11 \wedge \\
&x_3 + x_9 = 16 \wedge \\
&x_5 \geq x_8 + x_9 \wedge \\
&b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
&b \rightarrow (x_6 \geq 7) \wedge \\
&b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
&x_{11} \geq x_9 + x_{10}
\end{aligned}
$$

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

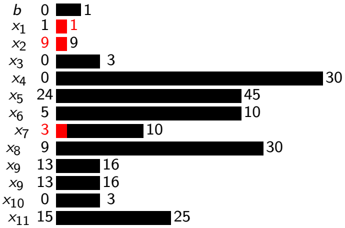


Example

$$
\begin{aligned}
&x_1 + x_7 \geq 4 \wedge \\
&x_2 + x_{10} \geq 11 \wedge \\
&x_3 + x_9 = 16 \wedge \\
&x_5 \geq x_8 + x_9 \wedge \\
&b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
&b \rightarrow (x_6 \geq 7) \wedge \\
&b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
&x_{11} \geq x_9 + x_{10}
\end{aligned}
$$

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

$[\![x_2 = 9]\!]$

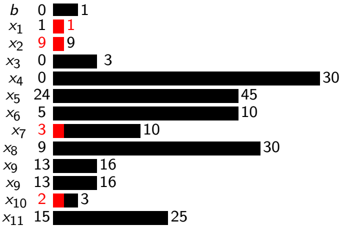


Example

$$x_1 + x_7 \geq 4 \wedge$$
$$x_2 + x_{10} \geq 11 \wedge$$
$$x_3 + x_9 = 16 \wedge$$
$$x_5 \geq x_8 + x_9 \wedge$$
$$b \leftrightarrow (x_9 - x_4 = 14) \wedge$$
$$b \rightarrow (x_6 \geq 7) \wedge$$
$$b \rightarrow (x_6 + x_7 \leq 9) \wedge$$
$$x_{11} \geq x_9 + x_{10}$$

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

$[\![x_2 = 9]\!] \longrightarrow [\![x_{10} \geq 2]\!]$



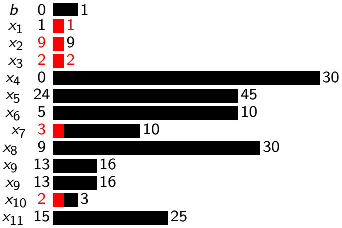
Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

$[\![x_2 = 9]\!] \longrightarrow [\![x_{10} \geq 2]\!]$

$[\![x_3 = 2]\!]$

$$
\begin{aligned}
& x_1 + x_7 \geq 4 \wedge \\
& x_2 + x_{10} \geq 11 \wedge \\
& x_3 + x_9 = 16 \wedge \\
& x_5 \geq x_8 + x_9 \wedge \\
& b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
& b \rightarrow (x_6 \geq 7) \wedge \\
& b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
& x_{11} \geq x_9 + x_{10}
\end{aligned}
$$



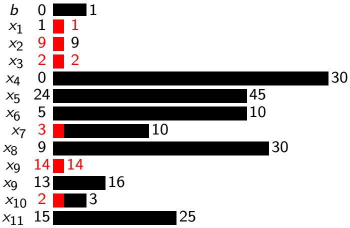
Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

$[\![x_2 = 9]\!] \longrightarrow [\![x_{10} \geq 2]\!]$

$[\![x_3 = 2]\!] \longrightarrow [\![x_9 = 14]\!]$

$$
\begin{aligned}
& x_1 + x_7 \geq 4 \wedge \\
& x_2 + x_{10} \geq 11 \wedge \\
& x_3 + x_9 = 16 \wedge \\
& x_5 \geq x_8 + x_9 \wedge \\
& b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
& b \rightarrow (x_6 \geq 7) \wedge \\
& b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
& x_{11} \geq x_9 + x_{10}
\end{aligned}
$$



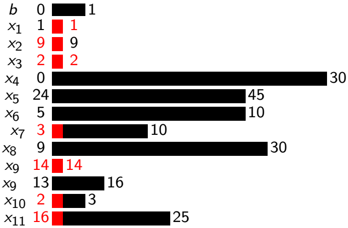
Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

$[\![x_2 = 9]\!] \longrightarrow [\![x_{10} \geq 2]\!]$

$[\![x_3 = 2]\!] \longrightarrow [\![x_9 = 14]\!] \rightarrow [\![x_{11} \geq 16]\!]$

$$
\begin{aligned}
&x_1 + x_7 \geq 4 \wedge \\
&x_2 + x_{10} \geq 11 \wedge \\
&x_3 + x_9 = 16 \wedge \\
&x_5 \geq x_8 + x_9 \wedge \\
&b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
&b \rightarrow (x_6 \geq 7) \wedge \\
&b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
&x_{11} \geq x_9 + x_{10}
\end{aligned}
$$



Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

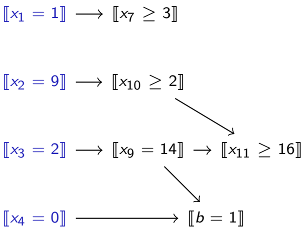
$[\![x_2 = 9]\!] \longrightarrow [\![x_{10} \geq 2]\!]$

$[\![x_3 = 2]\!] \longrightarrow [\![x_9 = 14]\!] \rightarrow [\![x_{11} \geq 16]\!]$

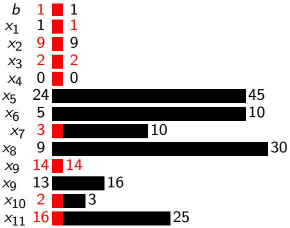
$[\![x_4 = 0]\!]$

$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$

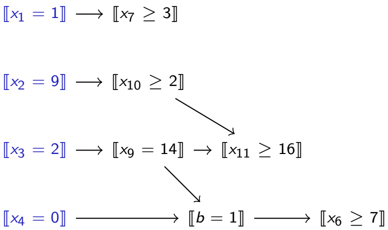
Example



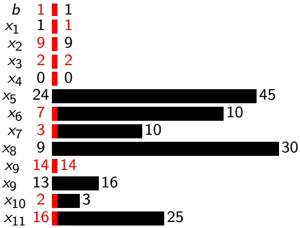
$$
\begin{aligned}
& x_1 + x_7 \geq 4 \wedge \\
& x_2 + x_{10} \geq 11 \wedge \\
& x_3 + x_9 = 16 \wedge \\
& x_5 \geq x_8 + x_9 \wedge \\
& b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
& b \rightarrow (x_6 \geq 7) \wedge \\
& b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
& x_{11} \geq x_9 + x_{10}
\end{aligned}
$$



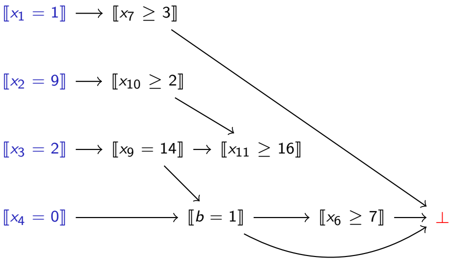
Example



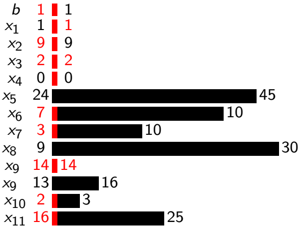
$$
\begin{aligned}
&x_1 + x_7 \geq 4 \wedge \\
&x_2 + x_{10} \geq 11 \wedge \\
&x_3 + x_9 = 16 \wedge \\
&x_5 \geq x_8 + x_9 \wedge \\
&b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
&b \rightarrow (x_6 \geq 7) \wedge \\
&b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
&x_{11} \geq x_9 + x_{10}
\end{aligned}
$$



Example

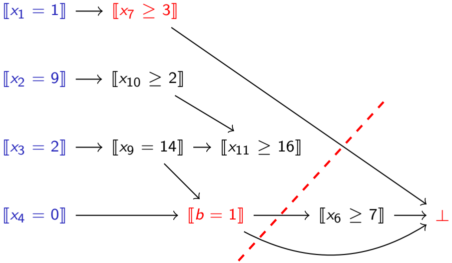


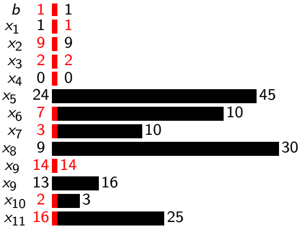
$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$



Example

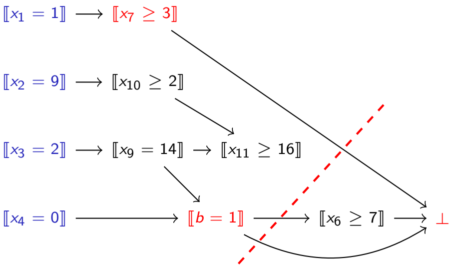
$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$



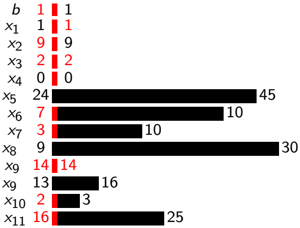


- Conflict analysis: $[\![b = 1]\!] \wedge [\![x_7 \geq 3]\!] \Rightarrow \bot$

Example



$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$



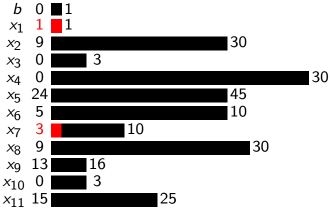
- Conflict analysis: $[\![b = 1]\!] \wedge [\![x_7 \geq 3]\!] \Rightarrow \bot$
- New clause: $[\![b \neq 0]\!] \vee [\![x_7 \leq 2]\!]$

Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!]$

- Conflict analysis: $[\![b = 1]\!] \wedge [\![x_7 \geq 3]\!] \Rightarrow \bot$
- New clause: $[\![b \neq 0]\!] \vee [\![x_7 \leq 2]\!]$
- Backtrack to level 1

$$
\begin{aligned}
&x_1 + x_7 \geq 4 \wedge \\
&x_2 + x_{10} \geq 11 \wedge \\
&x_3 + x_9 = 16 \wedge \\
&x_5 \geq x_8 + x_9 \wedge \\
&b \leftrightarrow (x_9 - x_4 = 14) \wedge \\
&b \rightarrow (x_6 \geq 7) \wedge \\
&b \rightarrow (x_6 + x_7 \leq 9) \wedge \\
&x_{11} \geq x_9 + x_{10}
\end{aligned}
$$

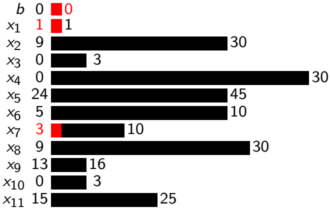


Example

$$x_1 + x_7 \geq 4 \wedge$$
$$x_2 + x_{10} \geq 11 \wedge$$
$$x_3 + x_9 = 16 \wedge$$
$$x_5 \geq x_8 + x_9 \wedge$$
$$b \leftrightarrow (x_9 - x_4 = 14) \wedge$$
$$b \rightarrow (x_6 \geq 7) \wedge$$
$$b \rightarrow (x_6 + x_7 \leq 9) \wedge$$
$$x_{11} \geq x_9 + x_{10}$$

$$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!] \longrightarrow [\![b = 0]\!]$$

- Conflict analysis: $[\![b = 1]\!] \wedge [\![x_7 \geq 3]\!] \Rightarrow \bot$
- New clause: $[\![b \neq 0]\!] \vee [\![x_7 \leq 2]\!]$
- Backtrack to level 1
- Propagate the learnt clause

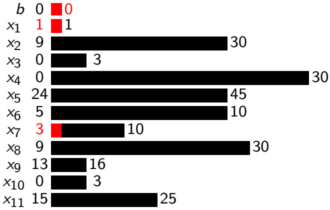


Example

$[\![x_1 = 1]\!] \longrightarrow [\![x_7 \geq 3]\!] \longrightarrow [\![b = 0]\!]$

- Conflict analysis: $[\![b = 1]\!] \wedge [\![x_7 \geq 3]\!] \Rightarrow \bot$
- New clause: $[\![b \neq 0]\!] \vee [\![x_7 \leq 2]\!]$
- Backtrack to level 1
- Propagate the learnt clause
- Continue exploration

$x_1 + x_7 \geq 4 \wedge$
$x_2 + x_{10} \geq 11 \wedge$
$x_3 + x_9 = 16 \wedge$
$x_5 \geq x_8 + x_9 \wedge$
$b \leftrightarrow (x_9 - x_4 = 14) \wedge$
$b \rightarrow (x_6 \geq 7) \wedge$
$b \rightarrow (x_6 + x_7 \leq 9) \wedge$
$x_{11} \geq x_9 + x_{10}$



Learning in CP

- Hybrid CP/SAT
- Conflict Driven Clause Learning (CDCL) [Moskewicz et al., 2001]
- Based on the notion of explanation
- Forward/Backward explanations
- Domain atoms can be generated Eagerly/Lazily

Our contributions

- Alternative lazy (atom) generation approach
- Novel conflict analysis scheme tailored to disjunctive scheduling

Modelling [Grimes and Hebrard, 2015]

Unary Resource Constraint

- $O(n^2)$ Boolean variables δ_{kij} $(i < j \in [1, n])$ per machine M_k.
- Decomposition using the following DISJUNCTIVE constraints:

$$\delta_{kij} = \begin{cases} 0 & \Leftrightarrow \quad t_{ik} + p_{ik} \leq t_{jk} \\ 1 & \Leftrightarrow \quad t_{jk} + p_{jk} \leq t_{ik} \end{cases} \tag{1}$$

Modelling [Grimes and Hebrard, 2015]

Unary Resource Constraint

- $O(n^2)$ Boolean variables δ_{kij} $(i < j \in [1, n])$ per machine M_k.
- Decomposition using the following DISJUNCTIVE constraints:

$$\delta_{kij} = \begin{cases} 0 & \Leftrightarrow \quad t_{ik} + p_{ik} \leq t_{jk} \\ 1 & \Leftrightarrow \quad t_{jk} + p_{jk} \leq t_{ik} \end{cases} \tag{1}$$

- Branch on the Boolean variables of the DISJUNCTIVE constraints.

Modelling [Grimes and Hebrard, 2015]

Unary Resource Constraint

- $O(n^2)$ Boolean variables δ_{kij} $(i < j \in [1, n])$ per machine M_k.
- Decomposition using the following DISJUNCTIVE constraints:

$$\delta_{kij} = \begin{cases} 0 & \Leftrightarrow \quad t_{ik} + p_{ik} \leq t_{jk} \\ 1 & \Leftrightarrow \quad t_{jk} + p_{jk} \leq t_{ik} \end{cases} \tag{1}$$

- Branch on the Boolean variables of the DISJUNCTIVE constraints.
- Variable ordering: wdeg, VSIDS.

Modelling [Grimes and Hebrard, 2015]

Unary Resource Constraint

- $O(n^2)$ Boolean variables δ_{kij} $(i < j \in [1, n])$ per machine M_k.
- Decomposition using the following DISJUNCTIVE constraints:

$$\delta_{kij} = \begin{cases} 0 & \Leftrightarrow \quad t_{ik} + p_{ik} \leq t_{jk} \\ 1 & \Leftrightarrow \quad t_{jk} + p_{jk} \leq t_{ik} \end{cases} \tag{1}$$

- Branch on the Boolean variables of the DISJUNCTIVE constraints.
- Variable ordering: wdeg, VSIDS.
- Value ordering: Solution guided [Beck, 2007].

Modelling [Grimes and Hebrard, 2015]

Unary Resource Constraint

- $O(n^2)$ Boolean variables δ_{kij} $(i < j \in [1, n])$ per machine M_k.
- Decomposition using the following DISJUNCTIVE constraints:

$$\delta_{kij} = \begin{cases} 0 & \Leftrightarrow & t_{ik} + p_{ik} \leq t_{jk} \\ 1 & \Leftrightarrow & t_{jk} + p_{jk} \leq t_{ik} \end{cases} \tag{1}$$

- Branch on the Boolean variables of the DISJUNCTIVE constraints.
- Variable ordering: wdeg, VSIDS.
- Value ordering: Solution guided [Beck, 2007].
- Greedy heuristic, dichotomic search, branch and bound

Revisiting Lazy Atom Generation

Revisiting Lazy Atom Generation

Domain Encoding: standard approach

1. Generate domain atoms: $a \leftrightarrow [\![x = d]\!]$, $b \leftrightarrow [\![x \leq d]\!]$
2. Generate domain clauses: $\neg[\![x \leq d]\!] \vee [\![x \leq d+1]\!]$, $\neg[\![x = d]\!] \vee [\![x \leq d]\!]$, etc

Revisiting Lazy Atom Generation

Domain Encoding: standard approach

1. Generate domain atoms: $a \leftrightarrow [\![x = d]\!]$, $b \leftrightarrow [\![x \leq d]\!]$
2. Generate domain clauses: $\neg[\![x \leq d]\!] \vee [\![x \leq d + 1]\!]$, $\neg[\![x = d]\!] \vee [\![x \leq d]\!]$, etc

Lazy Atom Generation

1. Atoms and domain clauses are generated during conflict analysis
2. There is a redundancy issue

Revisiting Lazy Atom Generation

Domain Encoding: standard approach

1. Generate domain atoms: $a \leftrightarrow [\![x = d]\!]$, $b \leftrightarrow [\![x \leq d]\!]$
2. Generate domain clauses: $\neg[\![x \leq d]\!] \vee [\![x \leq d+1]\!]$, $\neg[\![x = d]\!] \vee [\![x \leq d]\!]$, etc

Lazy Atom Generation

1. Atoms and domain clauses are generated during conflict analysis
2. There is a redundancy issue
3. Suppose that $[\![x \leq 2]\!]$, $[\![x \leq 4]\!]$ are already generated with $\neg[\![x \leq 2]\!] \vee [\![x \leq 4]\!]$ and we will generate $[\![x \leq 3]\!]$.
4. Add $\neg[\![x \leq 2]\!] \vee [\![x \leq 3]\!]$, $\neg[\![x \leq 3]\!] \vee [\![x \leq 4]\!]$.
5. $\neg[\![x \leq 2]\!] \vee [\![x \leq 4]\!]$ is useless (redundant)
6. For a domain of size k, $k - 2$ redundant clauses.

Avoiding redundancy via DomainFaithfulness

Avoiding redundancy via DomainFaithfulness

Key Idea

- Use a single constraint responsible for the consistency of the domain.
- Whenever an atom is generated, we update the internal structure of the constraint

Avoiding redundancy via DomainFaithfulness

Key Idea

- Use a single constraint responsible for the consistency of the domain.
- Whenever an atom is generated, we update the internal structure of the constraint

Definition

$\textsc{DomainFaithfulness}(x, [b_1 \ldots b_n], [v_1, \ldots, v_n]) : \forall i, b_i \leftrightarrow x \leq v_i$

Avoiding redundancy via DomainFaithfulness

Key Idea

- Use a single constraint responsible for the consistency of the domain.
- Whenever an atom is generated, we update the internal structure of the constraint

Definition

$\textsc{DomainFaithfulness}(x, [b_1 \dots b_n], [v_1, \dots, v_n]) : \forall i, b_i \leftrightarrow x \leq v_i$

Arc consistency

Can be enforced in constant amortized time complexity ($O(1)$) down a branch of the search tree

DISJUNCTIVE-based Learning

DISJUNCTIVE-based Learning

- Branch on the reified Boolean variables
- $\rightarrow$ There exists an explanation for every bound literal $[\![x \leq u]\!]$

Disjunctive-based Learning

- Branch on the reified Boolean variables
- $\rightarrow$ There exists an explanation for every bound literal $[\![x \leq u]\!]$

Disjunctive-based Learning

Two phases:

1. 1-UIP cut
2. Apply resolution for every bound literal until having a nogood with only reified Boolean variables

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge a \wedge [\![x \geq 0]\!] \wedge [\![y \geq 4]\!] \Rightarrow \bot$

Disjunctive-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \perp$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \perp$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \perp$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge a \wedge [\![x \geq 0]\!] \wedge [\![y \geq 4]\!] \Rightarrow \perp$
- Nogood Reduction $a \wedge \neg b \wedge c \wedge [\![y \geq 4]\!] \Rightarrow \perp$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge a \wedge [\![x \geq 0]\!] \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- Nogood Reduction $a \wedge \neg b \wedge c \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- $\neg b \wedge [\![z \geq 0]\!] \Rightarrow [\![y \geq 4]\!]$

Disjunctive-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge a \wedge [\![x \geq 0]\!] \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- Nogood Reduction $a \wedge \neg b \wedge c \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- $\neg b \wedge [\![z \geq 0]\!] \Rightarrow [\![y \geq 4]\!]$
- Resolution $a \wedge \neg b \wedge c \Rightarrow \bot$

DISJUNCTIVE-based Learning

Example

- 1-UIP nogood: $a \wedge \neg b \wedge [\![x \leq 7]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $c \wedge [\![z \leq 13]\!] \Rightarrow [\![x \leq 7]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![y \leq 9]\!] \Rightarrow \bot$
- $[\![x \geq 4]\!] \Rightarrow [\![y \leq 9]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge [\![z \leq 13]\!] \wedge [\![x \geq 4]\!] \Rightarrow \bot$
- $a \wedge [\![x \geq 0]\!] \Rightarrow [\![z \leq 13]\!]$
- Resolution $a \wedge \neg b \wedge c \wedge a \wedge [\![x \geq 0]\!] \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- Nogood Reduction $a \wedge \neg b \wedge c \wedge [\![y \geq 4]\!] \Rightarrow \bot$
- $\neg b \wedge [\![z \geq 0]\!] \Rightarrow [\![y \geq 4]\!]$
- Resolution $a \wedge \neg b \wedge c \Rightarrow \bot$

- ⊕ No domain encoding
- ⊕ Scheduling horizon does not manner in size
- ⊖ Language of literals is restricted compared to standard approaches

Experimental results

Protocol

- **Mistral-Hybrid:** backward explanations, semantic reductions, lazy generation, DISJUNCTIVE-based learning
- **http://siala.github.io/jssp/details.pdf**
- Lawrence and Taillard Job Shop benchmarks
- Global cutoff: 1h

Experimental results

Protocol

- **Mistral-Hybrid:** backward explanations, semantic reductions, lazy generation, DISJUNCTIVE-based learning
- **http://siala.github.io/jssp/details.pdf**
- Lawrence and Taillard Job Shop benchmarks
- Global cutoff: 1h

How hard are Taillard instances?

- Proposed 2 decades ago
- State-of-the art method recently proposed in [Vilím et al., 2015]
  - IBM CP-Optimizer studio
  - 8h20min per instance
  - Parallelization
  - Start search with best known bounds as an additional information.

Experimental results: Summary

Experimental results: Summary

Mostly proven optimal

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013

Hard instances

Instances	CP(*task*)		H(*vsids*, *disj*)		H(*vsids*, *lazy*)		H(*task*, *disj*)		H(*task*, *lazy*)	
	PRD	nds/s	PRD	nds/s	PRD	nds/s	PRD	nds/s	PRD	nds/s
tai-11-20	1.8432	4908	**1.1564**	3583	1.3725	531	1.2741	2544	1.2824	489
tai-21-30	1.6131	3244	0.9150	2361	1.0841	438	0.9660	1694	**0.8745**	409
tai-31-40	5.4149	3501	4.0210	2623	**3.7350**	580	4.0536	1497	3.8844	510
tai-41-50	7.0439	2234	4.8362	1615	**4.6800**	436	4.9305	1003	5.0136	390
tai-51-60	3.0346	1688	3.2449	2726	3.7809	593	**1.1156**	1099	1.1675	575
tai-61-70	6.8598	1432	6.5890	2414	5.4264	578	3.9637	866	**3.6617**	533

Experimental results: Summary

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
Mostly proven optimal															
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013
Hard instances															
	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s
tai-11-20	1.8432		4908	**1.1564**		3583	1.3725		531	1.2741		2544	1.2824		489
tai-21-30	1.6131		3244	0.9150		2361	1.0841		438	0.9660		1694	**0.8745**		409
tai-31-40	5.4149		3501	4.0210		2623	**3.7350**		580	4.0536		1497	3.8844		510
tai-41-50	7.0439		2234	4.8362		1615	**4.6800**		436	4.9305		1003	5.0136		390
tai-51-60	3.0346		1688	3.2449		2726	3.7809		593	**1.1156**		1099	1.1675		575
tai-61-70	6.8598		1432	6.5890		2414	5.4264		578	3.9637		866	**3.6617**		533

- Hybrid models outperform CP

Experimental results: Summary

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
Mostly proven optimal															
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013

Instances	CP(*task*)		H(*vsids*, *disj*)		H(*vsids*, *lazy*)		H(*task*, *disj*)		H(*task*, *lazy*)	
Hard instances										
	PRD	nds/s	PRD	nds/s	PRD	nds/s	PRD	nds/s	PRD	nds/s
tai-11-20	1.8432	4908	**1.1564**	3583	1.3725	531	1.2741	2544	1.2824	489
tai-21-30	1.6131	3244	0.9150	2361	1.0841	438	0.9660	1694	**0.8745**	409
tai-31-40	5.4149	3501	4.0210	2623	**3.7350**	580	4.0536	1497	3.8844	510
tai-41-50	7.0439	2234	4.8362	1615	**4.6800**	436	4.9305	1003	5.0136	390
tai-51-60	3.0346	1688	3.2449	2726	3.7809	593	**1.1156**	1099	1.1675	575
tai-61-70	6.8598	1432	6.5890	2414	5.4264	578	3.9637	866	**3.6617**	533

- The impact of clause learning is more visible when the size of the instance grows

Experimental results: Summary

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
Mostly proven optimal															
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013
Hard instances															
	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s
tai-11-20	1.8432		4908	**1.1564**		3583	1.3725		531	1.2741		2544	1.2824		489
tai-21-30	1.6131		3244	0.9150		2361	1.0841		438	0.9660		1694	**0.8745**		409
tai-31-40	5.4149		3501	4.0210		2623	**3.7350**		580	4.0536		1497	3.8844		510
tai-41-50	7.0439		2234	4.8362		1615	**4.6800**		436	4.9305		1003	5.0136		390
tai-51-60	3.0346		1688	3.2449		2726	3.7809		593	**1.1156**		1099	1.1675		575
tai-61-70	6.8598		1432	6.5890		2414	5.4264		578	3.9637		866	**3.6617**		533

- DISJUNCTIVE-based learning outperforms the other models on medium sized instances

Experimental results: Summary

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
Mostly proven optimal															
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013
Hard instances															
	PRD	nds/s		PRD	nds/s		PRD	nds/s		PRD	nds/s		PRD	nds/s	
tai-11-20	1.8432	4908		**1.1564**	3583		1.3725	531		1.2741	2544		1.2824	489	
tai-21-30	1.6131	3244		0.9150	2361		1.0841	438		0.9660	1694		**0.8745**	409	
tai-31-40	5.4149	3501		4.0210	2623		**3.7350**	580		4.0536	1497		3.8844	510	
tai-41-50	7.0439	2234		4.8362	1615		**4.6800**	436		4.9305	1003		5.0136	390	
tai-51-60	3.0346	1688		3.2449	2726		3.7809	593		**1.1156**	1099		1.1675	575	
tai-61-70	6.8598	1432		6.5890	2414		5.4264	578		3.9637	866		**3.6617**	533	

- wdeg is the best choice with the largest instances.

Experimental results: Summary

Instances	CP(*task*)			H(*vsids*, *disj*)			H(*vsids*, *lazy*)			H(*task*, *disj*)			H(*task*, *lazy*)		
Mostly proven optimal															
	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s	%O	T	nds/s
la-01-40	87	522	8750	**91.5**	437	6814	88	632	2332	90.50	434	5218	88.75	509	2694
tai-01-10	89	768	5875	**90**	517	4975	88	1060	1033	90	634	3572	84	1227	1013
Hard instances															
	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s	PRD		nds/s
tai-11-20	1.8432		4908	**1.1564**		3583	1.3725		531	1.2741		2544	1.2824		489
tai-21-30	1.6131		3244	0.9150		2361	1.0841		438	0.9660		1694	**0.8745**		409
tai-31-40	5.4149		3501	4.0210		2623	**3.7350**		580	4.0536		1497	3.8844		510
tai-41-50	7.0439		2234	4.8362		1615	**4.6800**		436	4.9305		1003	5.0136		390
tai-51-60	3.0346		1688	3.2449		2726	3.7809		593	**1.1156**		1099	1.1675		575
tai-61-70	6.8598		1432	6.5890		2414	5.4264		578	3.9637		866	**3.6617**		533

- Surprisingly DISJUNCTIVE-based learns shorter clauses

Experimental results: lower bounds computation

Experimental results: lower bounds computation

Open instances from Taillard benchmark before [Vilím et al., 2015]

- 7 new bounds found with DISJUNCTIVE-based learning and VSIDS

tai13		tai21		tai23		tai25		tai26		tai29		tai30	
new	old	new	old	new	old	new	old	new	old	new	old	new	old
1305	1282	**1613**	1573	**1514**	1474	**1544**	1518	**1561**	1558	**1576**	1525	**1515**	1485

Experimental results: lower bounds computation

Open instances from Taillard benchmark before [Vilím et al., 2015]

- 7 new bounds found with DISJUNCTIVE-based learning and VSIDS

tai13		tai21		tai23		tai25		tai26		tai29		tai30	
new	old	new	old	new	old	new	old	new	old	new	old	new	old
1305	1282	**1613**	1573	**1514**	1474	**1544**	1518	**1561**	1558	**1576**	1525	**1515**	1485
1342		**1642**		**1518**		**1558**		**1591**		**1573**		**1519**	

- 8h20min per instance
- Parallelization
- Start search with best known bounds as an additional information.

Experimental results: lower bounds computation

Open instances from Taillard benchmark before [Vilím et al., 2015]

- 7 new bounds found with DISJUNCTIVE-based learning and VSIDS

tai13		tai21		tai23		tai25		tai26		tai29		tai30	
new	old	new	old	new	old	new	old	new	old	new	old	new	old
1305	1282	**1613**	1573	**1514**	1474	**1544**	1518	**1561**	1558	**1576**	1525	**1515**	1485
1342		**1642**		**1518**		**1558**		**1591**		**1573**		**1519**	

- 8h20min per instance
- Parallelization
- Start search with best known bounds as an additional information.

Relaunch with 2h

- tai-29: 1583 (1573 in [Vilím et al., 2015])
- tai-30: 1528 (1519 in [Vilím et al., 2015])

Summary

- Alternative lazy (atom) generation approach avoiding a redundancy issue
- Novel conflict analysis mechanism
- Efficient in practice, specially for finding proofs

Future Research

- Applications to other scheduling problems?
- Learning with global constraints?
- Hand-crafted learning?

Thank you.

References I

Beck, J. C. (2007).
Solution-guided multi-point constructive search for job shop scheduling.
Journal of Artificial Intelligence Research, 29(1):49–77.

Carlier, J. and Pinson, E. (1989).
An algorithm for solving the job-shop problem.
Management Science, 35(2):164–176.

Grimes, D. and Hebrard, E. (2015).
Solving Variants of the Job Shop Scheduling Problem through Conflict-Directed Search.
INFORMS Journal on Computing.

Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S. (2001).
Chaff: Engineering an Efficient SAT Solver.
In *Proceedings of the 38th Annual Design Automation Conference, DAC'01, Las Vegas, Nevada, USA*, pages 530–535.

Sadeh, N. M. (1991).
Lookahead techniques for micro-opportunistic job-shop scheduling.
PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

References II

Schutt, A., Feydy, T., Stuckey, P. J., and Wallace, M. G. (2013).
Solving rcpsp/max by lazy clause generation.
Journal of Scheduling, 16(3):273–289.

Vilím, P., Laborie, P., and Shaw, P. (2015).
Failure-directed Search for Constraint-based Scheduling.
In *Proceedings of the 12th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR'15, Barcelona, Spain*, page to appear.