Scheduling and (Integer) Linear Programming

Christian Artigues

LAAS - CNRS & Université de Toulouse, France

artigues@laas.fr

Master Class CPAIOR 2012 - Nantes

Outline

1. Introduction
2. Polyhedral studies and cutting plane generation
3. (Mixed) integer programming for scheduling problems
4. Column generation
5. A few Computational results

Outline

1. Introduction
2. Polyhedral studies and cutting plane generation
3. (Mixed) integer programming for scheduling problems
   - Time-indexed variables
   - Total unimodularity
   - Sequencing and natural date variables
   - Flow (TSP) and natural date variables
   - Positional date and assignment variables
   - Hybrid formulations
4. Column generation
5. A few Computational results

A bit of history

Since the beginning, linear programming has been used to solve scheduling problems.

> *"The military refer to their various plans or proposed schedules of training, logistical supply and deployment of combat units as a program. When I first analyzed the Air Force planning problem and saw that it could be formulated as a system of linear inequalities, I called my paper Programming in a Linear Structure" (Georges Dantzig)*

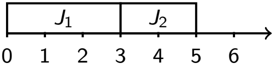
This presentation is a (non-exhaustive) survey of (integer) linear programming formulations and valid inequalities for scheduling problems

A simple scheduling example

One machine scheduling with release dates and deadlines

2 jobs J_1 and J_2 ($p_1 = 3$, $p_2 = 2$, $r_1 = 0$, $r_2 = 1$, $\tilde{d}_1 = 9$, $\tilde{d}_2 = 7$).
1 machine. Objective function $f(S) = C_1 + C_2 = S_1 + S_2 + p_1 + p_2$.

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 \\
S_1 \geq r_1 \\
S_2 \geq r_2 \\
S_1 + p_1 \leq \tilde{d}_1 \\
S_2 + p_2 \leq \tilde{d}_2 \\
S_2 \geq S_1 + p_1 \vee S_1 \geq S_2 + p_2 \\
S_1, S_2 \text{ integer}
\end{aligned}
$$



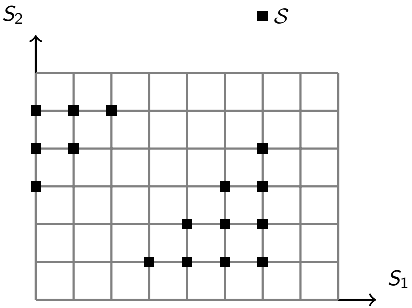
$\sum C_i = 8$ (optimal solution)

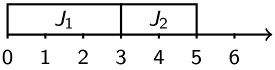
The scheduling polyhedron

Feasible set $\mathcal{S}$

The feasible set is $\mathcal{S}$ the set of points $S \in \mathbb{R}^n$ that satisfy the constraints

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 & \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 \geq S_1 + p_1 \vee S_1 &\geq S_2 + p_2 \\
S_1, S_2 &\text{ integer}
\end{aligned}
$$





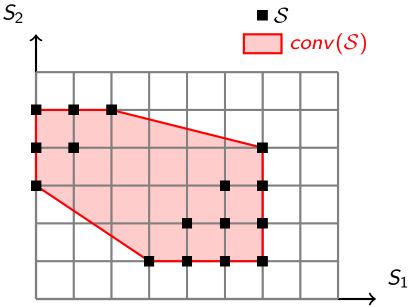
$\sum C_i = 8$

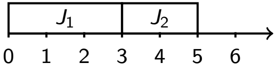
The scheduling polyhedron

Convex hull $conv(\mathcal{S})$

The *convex hull* of $\mathcal{S}$, i.e. the smallest convex set containing $\mathcal{S}$:
$conv(\mathcal{S}) = \left\{ x \in \mathbb{R}^n \middle| \exists \lambda_i \in \mathbb{R}^{n+}, x = \sum_{i=1}^{|\mathcal{S}|} \lambda_i S^i, \sum_{i=1}^{|\mathcal{S}|} \lambda_i = 1 \right\}$

$$\min S_1 + S_2 + p_1 + p_2$$
$$S_1 \geq r_1$$
$$S_2 \geq r_2$$
$$S_1 + p_1 \leq \tilde{d}_1$$
$$S_2 + p_2 \leq \tilde{d}_2$$
$$S_2 \geq S_1 + p_1 \vee S_1 \geq S_2 + p_2$$
$$S_1, S_2 \text{ integer}$$





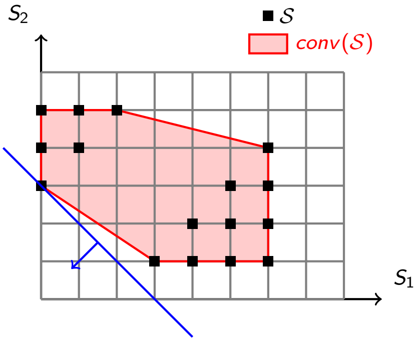
$\sum C_i = 8$

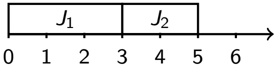
The scheduling polyhedron

Optimization on $\mathcal{S}$ and $conv(\mathcal{S})$

Let $f : \mathbb{R}^n \to \mathbb{R}$ a linear function, $\min_{S\in\mathcal{S}} f(S) = \min_{S\in conv(\mathcal{S})} f(S)$

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 & \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 \geq S_1 + p_1 &\vee S_1 \geq S_2 + p_2 \\
S_1, S_2 &\text{ integer}
\end{aligned}
$$





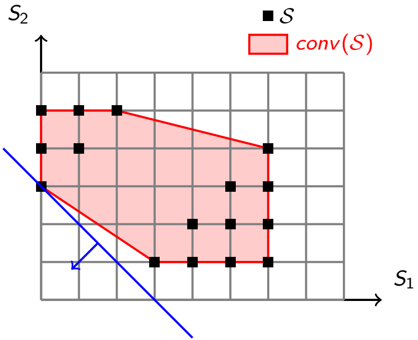
$\sum C_i = 8$

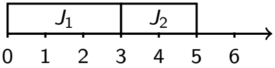
The scheduling polyhedron

$conv(\mathcal{S})$ is a polyhedron

There exists $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$ with m finite such that $conv(\mathcal{S}) = \{x \in \mathbb{R}^n | Ax \geq b\}$. Hence $\min\limits_{S \in conv(\mathcal{S})} f(S)$ is a LP.

$$
\begin{aligned}
\min S_1 + S_2 &+ p_1 + p_2 \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 \geq S_1 + p_1 &\vee S_1 \geq S_2 + p_2 \\
S_1, S_2 &\text{ integer}
\end{aligned}
$$





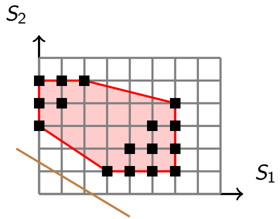
$\sum C_i = 8$

Find a complete description of $conv(\mathcal{S})$: hard in general

Definitions : valid inequalities, faces and facets

Let P denote a polyhedron.

- A valid inequality $\alpha x \geq \beta$ is such that $\forall S \in P$, S verifies the inequality (P is included in the halfspace induced by the inequality)
- A face is the intersection of P with the hyperplane $\{x \in \mathbb{R}^n | \alpha x = \beta\}$
- A vertex is a face of dimension 0
- A facet is a face of dimension $dim(P) - 1$

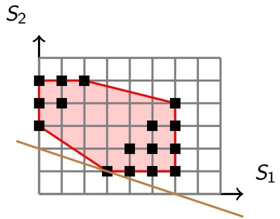


$3S_1 + 5S2 \geq 7$: a valid inequality

Definitions : valid inequalities, faces and facets

Let P denote a polyhedron.

- A valid inequality $\alpha x \geq \beta$ is such that $\forall S \in P$, S verifies the inequality (P is included in the halfspace induced by the inequality)
- A face is the intersection of P with the hyperplane $\{x \in \mathbb{R}^n | \alpha x = \beta\}$
- A vertex is a face of dimension 0
- A facet is a face of dimension $dim(P) - 1$

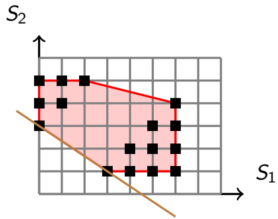


$P \cap \{S | S_1 + 3S_2 = 6\}$: a 0-dimensional face $(3, 1)$

Definitions : valid inequalities, faces and facets

Let P denote a polyhedron.

- A valid inequality $\alpha x \geq \beta$ is such that $\forall S \in P$, S verifies the inequality (P is included in the halfspace induced by the inequality)
- A face is the intersection of P with the hyperplane $\{x \in \mathbb{R}^n | \alpha x = \beta\}$
- A vertex is a face of dimension 0
- A facet is a face of dimension $dim(P) - 1$

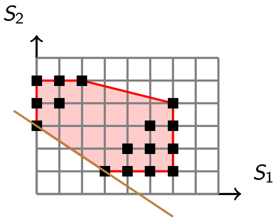


$P \cap \{S | 2S_1 + 3S_2 = 9\}$: a facet

Definitions : valid inequalities, faces and facets

Let P denote a polyhedron.

- A valid inequality $\alpha x \geq \beta$ is such that $\forall S \in P$, S verifies the inequality (P is included in the halfspace induced by the inequality)
- A face is the intersection of P with the hyperplane $\{x \in \mathbb{R}^n | \alpha x = \beta\}$
- A vertex is a face of dimension 0
- A facet is a face of dimension $dim(P) - 1$



$P \cap \{S | 2S_1 + 3S_2 = 9\}$: a facet

A polyhedron is fully described by the set of all facet-inducing valid inequalities

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

(Integer) linear programming-based scheduling

Given a scheduling problem and the logical description of $\mathcal{S}$

1. Perform a polyhedral study :
   - Find a complete description of $conv(\mathcal{S})$ with a polynomial number of linear inequalities ? $\rightarrow$ The problem can be solved by LP
   - Find a complete description of $conv(\mathcal{S})$ and show it has a supermodular structure ? $\rightarrow$ The problem can be solved by a greedy algorithm
   - Find a partial description of $conv(\mathcal{S})$? $\rightarrow$ This gives useful valid inequalities
2. Design a (mixed)-integer programming formulation
   - More polyhedral studies, solve with branch-and-cut, branch-and-price, branch-and-cut-and-price, heuristics, ... (only partly addressed here)

Outline

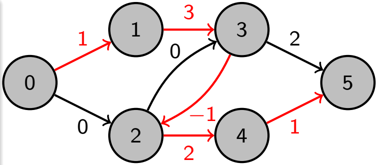
1. Introduction
2. Polyhedral studies and cutting plane generation
3. (Mixed) integer programming for scheduling problems
   - Time-indexed variables
   - Total unimodularity
   - Sequencing and natural date variables
   - Flow (TSP) and natural date variables
   - Positional date and assignment variables
   - Hybrid formulations
4. Column generation
5. A few Computational results

A scheduling problem that can be solved by LP

Project scheduling $\mathcal{S}^{PS}$

- Precedence constraints E.
- l_{ij} : minimum distance between the start of job i (S_i) and the start of job j (S_j).

$\mathcal{S}^{PS} = \{S \in \mathbb{R}^{+n} | S_j - S_i \geq l_{ij}, \forall (i,j) \in E\}$



find the shortest schedule

$$
\begin{aligned}
& \min S_{n+1} \\
& S_j - S_i \geq l_{ij} \quad \forall (i,j) \in E \\
& S_0 = 0 \\
& S_j \geq 0 \quad \forall j \in \mathcal{J}
\end{aligned}
$$

$$
\begin{aligned}
& \max \sum_{(i,j)\in E} l_{ij} x_{ij} \\
& \sum_{i \in \Gamma^{-1}(j)} x_{ij} = \sum_{i \in \Gamma(j)} x_{ji} \quad \forall j \in \mathcal{J} \setminus \{0\} \\
& \sum_{i \in \Gamma(0)} x_{0i} = 1 \\
& x_{0i} \geq 0
\end{aligned}
$$

(longest path model)

Supermodular polyhedron : definitions [Sch96]

- Consider a set $N = \{1, \ldots, n\}$ and its power set 2^N (set of all subsets of N) and a *supermodular* set function $f : 2^N \to \mathbb{R}$, i.e. a set function that verifies :

$$\begin{cases} f(\emptyset) = 0 \\ f(A \cup B) + f(A \cap B) \geq f(A) + f(B), \forall A, B \subseteq N \end{cases}$$

$\mathcal{P}(f) = \{x \in \mathbb{R}^{|N|} | \sum_{i \in A} x_i \geq f(A), \forall A \subseteq N\}$ *supermodular* polyhedron

- There is an $O(n \log n)$ greedy algorithm to find $x^* = \text{argmin}_{x \in \mathcal{P}(f)} cx$:
  - if $\exists i \in N$, $c_i < 0$, the problem is unbounded.
  - otherwise $x^* \in \mathcal{B}(f)$. Solve the problem with the following greedy algorithm

$$\begin{cases} x_1^* = f(\{1\}) \\ x_j^* = f(\{1, \ldots, j\}) - f(\{1, \ldots, j-1\}), \quad j = 2, \ldots, n. \end{cases}$$

Supermodular polyhedron : definitions [Sch96]

- Consider a set $N = \{1, \ldots, n\}$ and its power set 2^N (set of all subsets of N) and a *supermodular* set function $f : 2^N \to \mathbb{R}$, i.e. a set function that verifies :

$$\begin{cases} f(\emptyset) = 0 \\ f(A \cup B) + f(A \cap B) \geq f(A) + f(B), \forall A, B \subseteq N \end{cases}$$

$\mathcal{P}(f) = \{x \in \mathbb{R}^{|N|} | \sum_{i \in A} x_i \geq f(A), \forall A \subseteq N\}$ *supermodular* polyhedron

- There is an $O(n \log n)$ greedy algorithm to find $x^* = \mathrm{argmin}_{x \in \mathcal{P}(f)} cx$:
  - if $\exists i \in N$, $c_i < 0$, the problem is unbounded.
  - otherwise $x^* \in \mathcal{B}(f)$. Solve the problem with the following greedy algorithm

$$\begin{cases} x_1^* = f(\{1\}) \\ x_j^* = f(\{1, \ldots, j\}) - f(\{1, \ldots, j-1\}), \quad j = 2, \ldots, n. \end{cases}$$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \mid \sum_{j \in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j \in A} p_j\right)^2 + \sum_{j \in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
  - Optimum for $\sum_{i \in \mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i/p_i.
  - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
  - $\text{Opt} = p_{i_1} p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + p_{i_2} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \,|\, \sum_{j\in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j\in A} p_j\right)^2 + \sum_{j\in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
  - Optimum for $\sum_{i\in\mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i/p_i.
  - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
  - Opt $= p_{i_1}p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \,|\, \sum_{j \in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j\in A} p_j\right)^2 + \sum_{j \in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
    - Optimum for $\sum_{i \in \mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i/p_i.
    - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
    - Opt $= p_{i_1} p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + p_{i_2} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \,\middle|\, \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i, j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \,|\, \sum_{j \in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j \in A} p_j\right)^2 + \sum_{j \in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
  - Optimum for $\sum_{i \in \mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i/p_i.
  - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
  - $\text{Opt} = p_{i_1} p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + p_{i_2} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i, j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \,|\, \sum_{j \in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j \in A} p_j\right)^2 + \sum_{j \in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
  - Optimum for $\sum_{i \in \mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i / p_i.
  - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
  - Opt $= p_{i_1} p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + p_{i_2} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

Single-machine scheduling $\mathcal{S}^{SM}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine :

$$\mathcal{S}^{SM} = \left\{ C \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} C_i \geq p_i, \quad \forall j \in \mathcal{J}, \\ C_i \geq C_j + p_i \vee C_j \geq C_i + p_j, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- Queyranne [Que93] showed that $conv(\mathcal{S}^{SM}) = \{C \in \mathbb{R}^{\mathcal{J}} \,|\, \sum_{j \in A} p_j C_j \geq g(A), \forall A \subseteq \mathcal{J}\}$ where $g(A) = \frac{1}{2}\left(\left(\sum_{j \in A} p_j\right)^2 + \sum_{j \in A} p_j^2\right)$.
- These valid inequalities can be obtained by Smith's rule (WSPT).
  - Optimum for $\sum_{i \in \mathcal{J}} w_i C_i$ is obtained by sorting jobs in non decreasing w_i / p_i.
  - For $A = \{i_1, \ldots, i_s\} \subseteq \mathcal{J}$, set $w_i = p_i$ for $i \in A$ and $w_i = 0$ for $i \in \mathcal{J} \setminus A$
  - Opt $= p_{i_1} p_{i_1} + p_{i_2}(p_{i_1} + p_{i_2}) + \ldots + p_{i_s}(p_{i_1} + p_{i_2} + \ldots + p_{i_s}) = g(A)$

A scheduling supermodular polyhedron [Que93]

- Each inequality $\sum_{j \in S} p_j C_j \geq g(S)$ defines a facet for $conv(\mathcal{Q})$.
- $conv(\mathcal{S}^{SM})$ is a supermodular polyhedron and the greedy algorithm for minimizing $\sum_{i \in \mathcal{J}} w_i C_i$ coincides with the Smith's rule WSPT
- This rediscovers the WSPT rule but also provides valid inequalities for other (NP-hard) scheduling problems. An $O(n \log n)$ separation algorithm is available to find an inequality violated by a given vector C.
- More scheduling supermodular polyhedra :
  - the convex hull of the feasible start time for unit jobs on parallel machines with nonstationary speeds [QS95]
  - the convex hull of mean busy time vectors of preemptive schedules for jobs with release dates on a single machine [GQS+02]

A scheduling supermodular polyhedron [Que93]

- Each inequality $\sum_{j \in S} p_j C_j \geq g(S)$ defines a facet for $conv(\mathcal{Q})$.
- $conv(\mathcal{S}^{SM})$ is a supermodular polyhedron and the greedy algorithm for minimizing $\sum_{i \in \mathcal{J}} w_i C_i$ coincides with the Smith's rule WSPT
- This rediscovers the WSPT rule but also provides valid inequalities for other (NP-hard) scheduling problems. An $O(n \log n)$ separation algorithm is available to find an inequality violated by a given vector C.
- More scheduling supermodular polyhedra :
  - the convex hull of the feasible start time for unit jobs on parallel machines with nonstationary speeds [QS95]
  - the convex hull of mean busy time vectors of preemptive schedules for jobs with release dates on a single machine [GQS+02]

A scheduling supermodular polyhedron [Que93]

- Each inequality $\sum_{j \in S} p_j C_j \geq g(S)$ defines a facet for $conv(\mathcal{Q})$.
- $conv(\mathcal{S}^{SM})$ is a supermodular polyhedron and the greedy algorithm for minimizing $\sum_{i \in \mathcal{J}} w_i C_i$ coincides with the Smith's rule WSPT
- This rediscovers the WSPT rule but also provides valid inequalities for other (NP-hard) scheduling problems. An $O(n \log n)$ separation algorithm is available to find an inequality violated by a given vector C.
- More scheduling supermodular polyhedra :
  - the convex hull of the feasible start time for unit jobs on parallel machines with nonstationary speeds [QS95]
  - the convex hull of mean busy time vectors of preemptive schedules for jobs with release dates on a single machine [GQS+02]

A scheduling supermodular polyhedron [Que93]

- Each inequality $\sum_{j \in S} p_j C_j \geq g(S)$ defines a facet for $conv(\mathcal{Q})$.
- $conv(\mathcal{S}^{SM})$ is a supermodular polyhedron and the greedy algorithm for minimizing $\sum_{i \in \mathcal{J}} w_i C_i$ coincides with the Smith's rule WSPT
- This rediscovers the WSPT rule but also provides valid inequalities for other (NP-hard) scheduling problems. An $O(n \log n)$ separation algorithm is available to find an inequality violated by a given vector C.
- More scheduling supermodular polyhedra :
  - the convex hull of the feasible start time for unit jobs on parallel machines with nonstationary speeds [QS95]
  - the convex hull of mean busy time vectors of preemptive schedules for jobs with release dates on a single machine [GQS+02]

A scheduling supermodular polyhedron [Que93]

- Each inequality $\sum_{j \in S} p_j C_j \geq g(S)$ defines a facet for $conv(\mathcal{Q})$.
- $conv(\mathcal{S}^{SM})$ is a supermodular polyhedron and the greedy algorithm for minimizing $\sum_{i \in \mathcal{J}} w_i C_i$ coincides with the Smith's rule WSPT
- This rediscovers the WSPT rule but also provides valid inequalities for other (NP-hard) scheduling problems. An $O(n \log n)$ separation algorithm is available to find an inequality violated by a given vector C.
- More scheduling supermodular polyhedra :
  - the convex hull of the feasible start time for unit jobs on parallel machines with nonstationary speeds [QS95]
  - the convex hull of mean busy time vectors of preemptive schedules for jobs with release dates on a single machine [GQS$^+$02]

Valid inequalities for NP-hard scheduling problems

Single-machine scheduling with release dates $\mathcal{S}^{SMR}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine with release dates :

$$\mathcal{S}^{SMR} = \left\{ S \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} S_i \geq r_i, \quad \forall j \in \mathcal{J}, \\ S_i \geq S_j + l_{ij} \vee S_j \geq S_i + l_{ji}, \forall i, j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- The problem of minimizing $\sum_{S \in \mathcal{S}} w_i S_i$ is NP-hard $\implies$ no chance to have a complete characterization of $conv(\mathcal{S}^{SMR})$
- However, Balas [Bal85] studied $P = conv(\mathcal{S})$ and derived facet-defining inequalities. He showed that a facet-defining inequality for a subset of jobs $\mathcal{K} \subseteq \mathcal{J}$ induces also a facet for $\mathcal{J}$.
  - $\forall i \in \mathcal{J}$, $S_i \geq r_i$ induces a facet of P
  - $\forall i, j \in \mathcal{J}$, $i \neq j$, $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$ induces a facet of P if and only if $-d_{ji} < r_j - r_i < d_{ij}$

Valid inequalities for NP-hard scheduling problems

Single-machine scheduling with release dates $\mathcal{S}^{SMR}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine with release dates :

$$\mathcal{S}^{SMR} = \left\{ S \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} S_i \geq r_i, \quad \forall j \in \mathcal{J}, \\ S_i \geq S_j + l_{ij} \vee S_j \geq S_i + l_{ji}, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- The problem of minimizing $\sum_{S \in \mathcal{S}} w_i S_i$ is NP-hard $\implies$ no chance to have a complete characterization of $conv(\mathcal{S}^{SMR})$
- However, Balas [Bal85] studied $P = conv(\mathcal{S})$ and derived facet-defining inequalities. He showed that a facet-defining inequality for a subset of jobs $\mathcal{K} \subseteq \mathcal{J}$ induces also a facet for $\mathcal{J}$.
  - $\forall i \in \mathcal{J}$, $S_i \geq r_i$ induces a facet of P
  - $\forall i,j \in \mathcal{J}$, $i \neq j$, $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$ induces a facet of P if and only if $-d_{ji} < r_j - r_i < d_{ij}$

Valid inequalities for NP-hard scheduling problems

Single-machine scheduling with release dates $\mathcal{S}^{SMR}$

Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine with release dates :

$$\mathcal{S}^{SMR} = \left\{ S \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} S_i \geq r_i, \quad \forall j \in \mathcal{J}, \\ S_i \geq S_j + l_{ij} \vee S_j \geq S_i + l_{ji}, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- The problem of minimizing $\sum_{S \in \mathcal{S}} w_i S_i$ is NP-hard $\implies$ no chance to have a complete characterization of $conv(\mathcal{S}^{SMR})$
- However, Balas [Bal85] studied $P = conv(\mathcal{S})$ and derived facet-defining inequalities. He showed that a facet-defining inequality for a subset of jobs $\mathcal{K} \subseteq \mathcal{J}$ induces also a facet for $\mathcal{J}$.
  - $\forall i \in \mathcal{J}$, $S_i \geq r_i$ induces a facet of P
  - $\forall i,j \in \mathcal{J}$, $i \neq j$, $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$ induces a facet of P if and only if $-d_{ji} < r_j - r_i < d_{ij}$

Valid inequalities for NP-hard scheduling problems

Single-machine scheduling with release dates $\mathcal{S}^{SMR}$

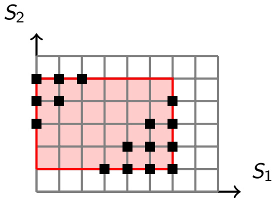
Set of feasible schedules for a set of jobs $\mathcal{J}$ on a single machine with release dates :

$$\mathcal{S}^{SMR} = \left\{ S \in \mathbb{R}^{\mathcal{J}} \;\middle|\; \begin{array}{l} S_i \geq r_i, \quad \forall j \in \mathcal{J}, \\ S_i \geq S_j + l_{ij} \vee S_j \geq S_i + l_{ji}, \forall i,j \in \mathcal{J}, i \neq j \end{array} \right\}$$

- The problem of minimizing $\sum_{S \in \mathcal{S}} w_i S_i$ is NP-hard $\implies$ no chance to have a complete characterization of $conv(\mathcal{S}^{SMR})$
- However, Balas [Bal85] studied $P = conv(\mathcal{S})$ and derived facet-defining inequalities. He showed that a facet-defining inequality for a subset of jobs $\mathcal{K} \subseteq \mathcal{J}$ induces also a facet for $\mathcal{J}$.
  - $\forall i \in \mathcal{J}$, $S_i \geq r_i$ induces a facet of P
  - $\forall i,j \in \mathcal{J}$, $i \neq j$, $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$ induces a facet of P if and only if $-d_{ji} < r_j - r_i < d_{ij}$

Cutting plane generation for NP-hard scheduling problems

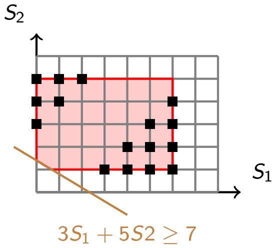
- From a partial descrition of $conv(\mathcal{S})$ (relaxation), iteratively solve the separation problem for families of valid inequalities.



- Optimum has been found after adding Balas inequality $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$
- Separation is NP-hard in general $\Longrightarrow$ the optimum cannot be found quickly by pure cutting plane generation.

Cutting plane generation for NP-hard scheduling problems

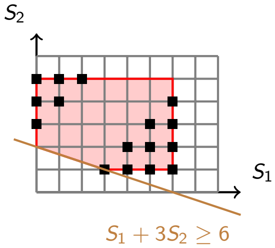
- From a partial descrition of $conv(\mathcal{S})$ (relaxation), iteratively solve the separation problem for families of valid inequalities.



- Optimum has been found after adding Balas inequality $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$
- Separation is NP-hard in general $\implies$ the optimum cannot be found quickly by pure cutting plane generation.

Cutting plane generation for NP-hard scheduling problems

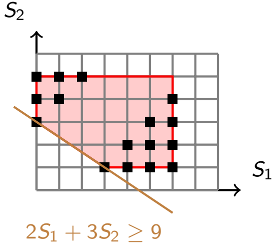
- From a partial descrition of $conv(\mathcal{S})$ (relaxation), iteratively solve the separation problem for families of valid inequalities.



- Optimum has been found after adding Balas inequality $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$
- Separation is NP-hard in general $\implies$ the optimum cannot be found quickly by pure cutting plane generation.

Cutting plane generation for NP-hard scheduling problems

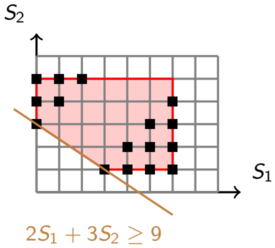
- From a partial descrition of $conv(\mathcal{S})$ (relaxation), iteratively solve the separation problem for families of valid inequalities.



- Optimum has been found after adding Balas inequality $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$
- Separation is NP-hard in general $\implies$ the optimum cannot be found quickly by pure cutting plane generation.

Cutting plane generation for NP-hard scheduling problems

- From a partial descrition of $conv(\mathcal{S})$ (relaxation), iteratively solve the separation problem for families of valid inequalities.

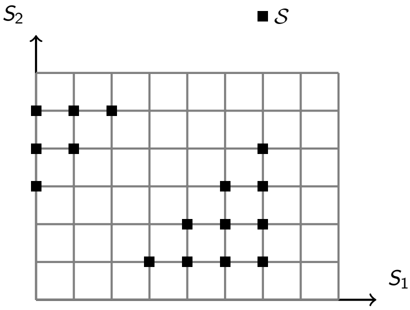


- Optimum has been found after adding Balas inequality $(l_{ij} + r_i - r_j)S_i + (l_{ji} + r_j - r_i)S_j \geq d_{ij}d_{ji} + L_i d_{ji} + L_j d_{ij}$
- Separation is NP-hard in general $\Longrightarrow$ the optimum cannot be found quickly by pure cutting plane generation.

Outline

Principle of (mixed)-integer programming

- Design a *good* MIP formulation for the scheduling problem
- Solve by branch-and-bound

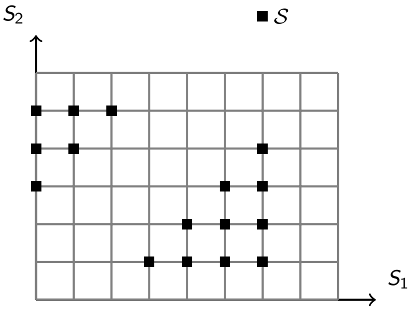


Remark : once x is fixed, extreme points are integral $\Longrightarrow$ no need for integer constraints on S

Principle of (mixed)-integer programming

- Design a *good* MIP formulation for the scheduling problem
- Solve by branch-and-bound

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 - S_1 + Mx &\geq p_1 \\
S_1 - S2 + M(1 - x) &\geq p_2 \\
S_1, S_2 &\text{ integer} \\
x &\in \{0, 1\}
\end{aligned}
$$

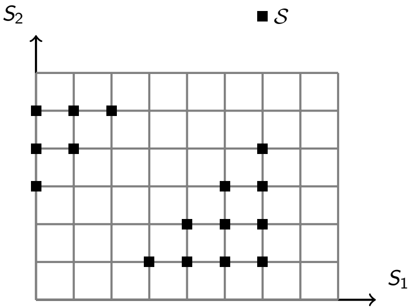


Remark : once x is fixed, extreme points are integral $\Longrightarrow$ no need for integer constraints on S

Principle of (mixed)-integer programming

- Design a *good* MIP formulation for the scheduling problem
- Solve by branch-and-bound

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 & \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 - S_1 + M{\color{red}x} &\geq p_1 \\
S_1 - S2 + M({\color{red}1 - x}) &\geq p_2 \\
S_1, S_2 & \text{ integer} \\
{\color{red}x \in \{0,1\}} &
\end{aligned}
$$

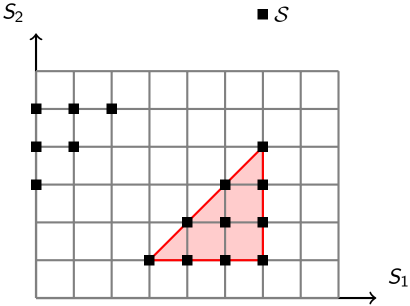


Remark : once x is fixed, extreme points are integral $\Longrightarrow$ no need for integer constraints on S

Principle of (mixed)-integer programming

- Design a *good* MIP formulation for the scheduling problem
- Solve by branch-and-bound

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 & \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 - S_1 + M{\color{red}x} &\geq p_1 \\
S_1 - S2 + M({\color{red}1 - x}) &\geq p_2 \\
S_1, S_2 &\text{ integer} \\
{\color{red}x \in \{0,1\}} &
\end{aligned}
$$



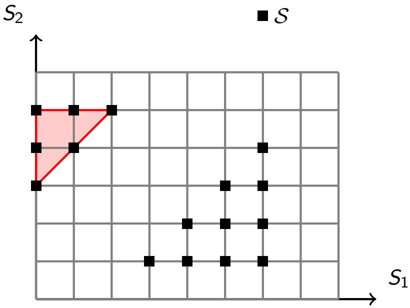
Left node $x = 1$

Remark : once x is fixed, extreme points are integral $\Longrightarrow$ no need for integer constraints on S

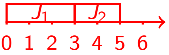
Principle of (mixed)-integer programming

- Design a *good* MIP formulation for the scheduling problem
- Solve by branch-and-bound

$$
\begin{aligned}
\min S_1 + S_2 + p_1 + p_2 & \\
S_1 &\geq r_1 \\
S_2 &\geq r_2 \\
S_1 + p_1 &\leq \tilde{d}_1 \\
S_2 + p_2 &\leq \tilde{d}_2 \\
S_2 - S_1 + Mx &\geq p_1 \\
S_1 - S2 + M(1 - x) &\geq p_2 \\
S_1, S_2 & \text{ integer} \\
x \in \{0, 1\} &
\end{aligned}
$$



Right node $x = 0$



$\sum C_i = 8$

Remark : once x is fixed, extreme points are integral $\implies$ no need for integer constraints on S

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \dots, T\}$.
- Precedence constraints E such that $(i,j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \dots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \dots, T\}$.
- Precedence constraints E such that $(i,j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \dots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \dots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \dots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Case study : the resource-constrained project scheduling problem

Resource-constrained project scheduling with irregular starting time costs

- n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- Precedence constraints E such that $(i,j) \in E \implies S_j \geq S_i + l_{ij}$.
- m resources (set $\mathcal{R}$). Constant availability B_k, $k \in \mathcal{R}$.
- For each job $i \in \mathcal{J}$: duration p_i and resource requirements b_{ik}, $k \in \mathcal{R}$.
- Resource constraints $\sum_{i \in \mathcal{J} | S_i \leq t \leq S_i + p_i - 1} b_{ik} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$.
- Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$.
- Find a schedule that minimizes $\sum_{i \in \mathcal{J}} w_j(S_j)$.

Remark 1 : $|\mathcal{R}| = 1, B_1 = 1$, and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ 1-machine problem.

Remark 2 : $|\mathcal{R}| = 1, B_1 \geq 2$ and $b_{i1} = 1, \forall i \in \mathcal{J} \implies$ parallel machine problem.

Scheduling objectives

Objective $\min \sum_{i \in \mathcal{J}} w_j(S_j)$ models most standard objectives

- **Makespan** $\min \max_{i \in \mathcal{J}} C_i$
  Let $\mathcal{J} = \{1, \ldots, n+1\}$ with $n+1$ dummy end jobs.
  $$w_i(t) = 0, \forall i \in \mathcal{J} \setminus \{n+1\}, \quad w_{n+1}(t) = t$$
- **Maximum lateness** $\min \max_{i \in \mathcal{J}} C_i - d_i$
  Same modeling by adding arc $(i, n+1)$ in E with $l_{i,n+1} = p_i - d_i$
- **Earliness-tardiness costs**
  $\min \sum_{i \in \mathcal{J}} (\alpha_i \max(0, d_i - C_i) + \beta_i \max(0, C_i - d_i)$
  $$w_i(t) = \alpha_i \max(0, d_i - t - p_i) + \beta_i \max(0, t + p_i - d_i)$$
  (weighted completion time if, in addition, $d_i = 0 \wedge \alpha_i = 0, \forall i \in \mathcal{J}$)
- **Weighted number of late jobs** $\sum_{i \in \mathcal{J}} w_i U_i$
  $$w_i(t) = \begin{cases} 0 \text{ if } t + p_i \leq d_i \\ w_i \text{ otherwise} \end{cases}$$

Scheduling objectives

Objective $\min \sum_{i \in \mathcal{J}} w_j(S_j)$ models most standard objectives

- **Makespan** $\min \max_{i \in \mathcal{J}} C_i$
  Let $\mathcal{J} = \{1, \ldots, n+1\}$ with $n+1$ dummy end jobs.
  $$w_i(t) = 0, \forall i \in \mathcal{J} \setminus \{n+1\}, \quad w_{n+1}(t) = t$$
- **Maximum lateness** $\min \max_{i \in \mathcal{J}} C_i - d_i$
  Same modeling by adding arc $(i, n+1)$ in E with $l_{i,n+1} = p_i - d_i$
- **Earliness-tardiness costs**
  $\min \sum_{i \in \mathcal{J}} (\alpha_i \max(0, d_i - C_i) + \beta_i \max(0, C_i - d_i))$
  $$w_i(t) = \alpha_i \max(0, d_i - t - p_i) + \beta_i \max(0, t + p_i - d_i)$$
  (weighted completion time if, in addition, $d_i = 0 \wedge \alpha_i = 0, \forall i \in \mathcal{J}$)
- **Weighted number of late jobs** $\sum_{i \in \mathcal{J}} w_i U_i$
  $$w_i(t) = \begin{cases} 0 \text{ if } t + p_i \leq d_i \\ w_i \text{ otherwise} \end{cases}$$

Scheduling objectives

Objective $\min \sum_{i \in \mathcal{J}} w_j(S_j)$ models most standard objectives

- **Makespan** $\min \max_{i \in \mathcal{J}} C_i$
  Let $\mathcal{J} = \{1, \ldots, n+1\}$ with $n+1$ dummy end jobs.
  $$w_i(t) = 0, \forall i \in \mathcal{J} \setminus \{n+1\}, \quad w_{n+1}(t) = t$$
- **Maximum lateness** $\min \max_{i \in \mathcal{J}} C_i - d_i$
  Same modeling by adding arc $(i, n+1)$ in E with $l_{i,n+1} = p_i - d_i$
- **Earliness-tardiness costs**
  $\min \sum_{i \in \mathcal{J}} (\alpha_i \max(0, d_i - C_i) + \beta_i \max(0, C_i - d_i))$
  $$w_i(t) = \alpha_i \max(0, d_i - t - p_i) + \beta_i \max(0, t + p_i - d_i)$$
  (weighted completion time if, in addition, $d_i = 0 \wedge \alpha_i = 0, \forall i \in \mathcal{J}$)
- **Weighted number of late jobs** $\sum_{i \in \mathcal{J}} w_i U_i$
  $$w_i(t) = \begin{cases} 0 \text{ if } t + p_i \leq d_i \\ w_i \text{ otherwise} \end{cases}$$

Scheduling objectives

Objective $\min \sum_{i \in \mathcal{J}} w_j(S_j)$ models most standard objectives

- **Makespan** $\min \max_{i \in \mathcal{J}} C_i$
  Let $\mathcal{J} = \{1, \ldots, n+1\}$ with $n+1$ dummy end jobs.

$$w_i(t) = 0, \forall i \in \mathcal{J} \setminus \{n+1\}, \quad w_{n+1}(t) = t$$

- **Maximum lateness** $\min \max_{i \in \mathcal{J}} C_i - d_i$
  Same modeling by adding arc $(i, n+1)$ in E with $l_{i,n+1} = p_i - d_i$
- **Earliness-tardiness costs**
  $\min \sum_{i \in \mathcal{J}} (\alpha_i \max(0, d_i - C_i) + \beta_i \max(0, C_i - d_i))$

$$w_i(t) = \alpha_i \max(0, d_i - t - p_i) + \beta_i \max(0, t + p_i - d_i)$$

  (weighted completion time if, in addition, $d_i = 0 \wedge \alpha_i = 0, \forall i \in \mathcal{J}$)
- **Weighted number of late jobs** $\sum_{i \in \mathcal{J}} w_i U_i$

$$w_i(t) = \begin{cases} 0 \text{ if } t + p_i \leq d_i \\ w_i \text{ otherwise} \end{cases}$$

Outline

MILP Formulations with time-indexed variables

time-indexed variables $x_{it} = 1 \Leftrightarrow S_i = t \Leftrightarrow S_i = \sum_{t=0}^{T} t x_{it}$

Single machine

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{t=0}^{T} t x_{jt} - \sum_{t=0}^{T} t x_{it} \geq l_{ij} \qquad \forall (i,j) \in E$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \qquad \forall t \in \mathcal{T}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

MILP Formulations with time-indexed variables

time-indexed variables $x_{it} = 1 \Leftrightarrow S_i = t \Leftrightarrow S_i = \sum_{t=0}^{T} t x_{it}$

Parallel machines

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{t=0}^{T} t x_{jt} - \sum_{t=0}^{T} t x_{it} \geq l_{ij} \qquad \forall (i,j) \in E$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq B \qquad \forall t \in \mathcal{T}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

MILP Formulations with time-indexed variables

time-indexed variables $x_{it} = 1 \Leftrightarrow S_i = t \Leftrightarrow S_i = \sum_{t=0}^{T} t x_{it}$

RCPSP

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{t=0}^{T} t x_{jt} - \sum_{t=0}^{T} t x_{it} \geq l_{ij} \qquad \forall (i,j) \in E$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} b_{jk} x_{js} \leq B_k \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

nT variables, $|E|$ precedence constraints, $|\mathcal{R}|T$ resource constraints.

MILP Formulations with time-indexed variables

time-indexed variables $x_{it} = 1 \Leftrightarrow S_i = t \Leftrightarrow S_i = \sum_{t=0}^{T} t x_{it}$
RCPSP (disaggregated precedence constraints)

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{s=t}^{T} x_{is} + \sum_{s=0}^{t+l_{ij}-1} x_{js} \leq 1 \qquad \forall (i,j) \in E, \forall t \in \mathcal{T}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} b_{jk} x_{js} \leq B_k \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

nT variables, $|E|T$ precedence constraints, $|\mathcal{R}|T$ resource constraints.

Outline

Total unimodularity

- A matrix A is totally unimodular (TU) if and only if every square submatrix has determinant 0, 1 or -1.
- if A is TU, $min_{Ax \geq b} cx$ has an integer optimal solution or there is no solution
- A is TU if (sufficient condition) rows can be partitionned into two disjoint sets B and C such that
  - each column of A has at most two non-zero entries,
  - each entry of A is 0, 1 or -1
  - if two non-zeros in a column have opposite signs, they are in the same subset of rows (both in B or both in C).
  - if two non-zeros in a column have the same sign, there is one in B and the other one in C
- A is also TU if its transpose is TU.

Total unimodularity and scheduling

Project scheduling with irregular starting time costs

n jobs (set $\mathcal{J}$) with integer start times $S_i \in \mathcal{T} = \{0, 1, \ldots, T\}$. Precedence constraints E such that $(i, j) \in E \implies S_j \geq S_i + l_{ij}$. Cost function $w_j : \{1, \ldots, T\} \to \mathbb{R}$. Find a schedule that minimizes $\sum_{j \in \mathcal{J}} w_j(S_j)$.

time-indexed variable $z_{it} = 1 \Leftrightarrow S_i \leq t$ $(z_{it} = \sum_{t=0}^{T} x_{it})$

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (w_j(t) - w_j(t+1))\, z_{jt}$$

$$\begin{aligned}
z_{jT} &= 1 && \forall j \in \mathcal{J} \\
z_{jt} - z_{j,t+1} &\leq 0 && \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \\
z_{j,t+l_{ij}} - z_{it} &\leq 0 && \forall (i,j) \in E, \forall t \in \mathcal{T} \\
z_{jt} &\in \{0, 1\} && \forall j \in \mathcal{J}, \forall t \in \mathcal{T}
\end{aligned}$$

The matrix totally unimodular $\implies$ no integer restriction needed!

Comparison between formulations

Other IP with time-indexed variables $x_{it} = 1 \Leftrightarrow S_i = t$ (nT variables)

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$
$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$
$$\sum_{s=t}^{T} x_{is} + \sum_{s=0}^{t+l_{ij}-1} x_{js} \leq 1 \qquad \forall (i,j) \in E, \forall t \in \mathcal{T}$$
$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Totally unimodular matrix, integer polyhedron

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$
$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$
$$\sum_{t=0}^{T} t x_{js} - \sum_{t=0}^{T} t x_{is} \geq l_{ij} \qquad \forall (i,j) \in E$$
$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Polyhedron is not integer !

For the RCPSP, the LP relaxation of the time-indexed model with disagreggated precedence constraints is tighter.

MILP Formulations with time-indexed variables : facets

Facet-inducing inequalities for the single-machine polyhedron (without precedence constraints) [SW92]

$$\sum_{t \in \mathcal{T}} x_{jt} \leq 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \qquad \forall t \in \mathcal{T}$$

$$\sum_{s=t-p_j+1}^{t+\Delta-1} x_{js} + \sum_{i \neq j} \sum_{s=t-p_i+\Delta}^{t} x_{is} \leq 1 \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T},$$

$$\forall \Delta \in \{2, \ldots, \max_{i \neq j} p_i\}$$

All facet-inducing inequalities with rhs $= 1$ [vHS99].

Sousa and Wolsey valid inequalities [SW92]

$$\sum_{u=1}^{T} \frac{x_{ju}}{2} = \frac{1}{2}$$

$$\sum_{i \in \mathcal{J}} \sum_{s=t-p_i+1}^{t} \frac{x_{is}}{2} \leq \frac{1}{2}$$

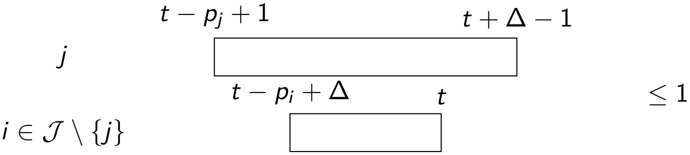
$$\sum_{i \in \mathcal{J}} \sum_{s=t-p_i+\Delta}^{t+\Delta-1} \frac{x_{is}}{2} \leq \frac{1}{2}$$

$$\lfloor \sum_{u=1}^{T} \frac{x_{ju}}{2} + \sum_{i=1}^{n} \sum_{s=t-p_i+1}^{t} \frac{x_{is}}{2} + \sum_{i \in \mathcal{J}} \sum_{s=t-p_i+\Delta}^{t+\Delta-1} \frac{x_{is}}{2} \rfloor \leq \lfloor \frac{3}{2} \rfloor$$

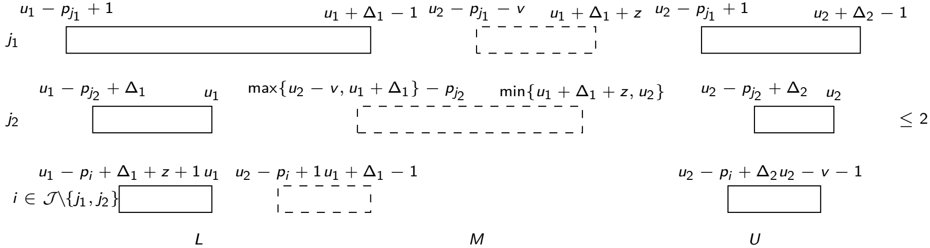
$$\sum_{s=t-p_j+1}^{t+\Delta-1} x_{js} + \sum_{i \neq j} \sum_{s=t-p_i+\Delta}^{t} x_{is} \leq 1$$

Sousa and Wolsey valid inequalities [SW92]

For each time period $t \in \mathcal{T}$, for each task $j \in \mathcal{T}$ and for each $\Delta \in \{2, \ldots, \max_{i \neq j} p_i\}$



van den Akker et al. valid ineqalities [vHS99]



With the same principle all facet-inducing inequalities with rhs $= 2$ are derived.

In [WJNS02], relation with valid ineqalities for the node packing problem is established.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Valid inequalities for the time-indexed RCPSP ILP

Hardin et al. [HNS08] extend the Sousa and Wolsey 1-machine cuts to the RCPSP.

Let F be a forbidden set (or cover), i.e. $\sum_{j \in F} b_i > B$.
$\sum_{j \in F} \sum_{s=t-p_j+1}^{t} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

Consider now a "special" task $j \in F$ and an interval $v \geq 0$
$\sum_{i \in F \setminus \{j\}} \sum_{s=t-p_j+1+v}^{t} x_{iq} + \sum_{s=t-p_j+1}^{t+v} x_{js} \leq |F| - 1$ is a valid inequality $\forall t \in \mathcal{T}$.

- The inequality defines a facet for the polyhedron reduced to jobs in C if C is a minimal forbidden set, i.e. $\sum_{C \setminus \{i\}} b_i \leq B, \forall i \in C$.
- Hardin et al. propose lifting procedures and conditions for the resulting inequalities to be facet-inducing for the complete polyhedron.

Outline

Sequencing variables

Time-indexed models yield good LP relaxation but the number of variables can be huge $\implies$ need to consider more compact models.

- Binary variable $y_{ij} = 1$ if and only if $S_j \geq S_i + p_i$
- Disjunctive case : $y_{ij} = 1 - y_{ji}, \forall (i,j) \in D$ (half of the variables can be dropped) and y_{ij} is the incidence vector of linear orderings :
- If there are no release dates and in the 1-machine case $C_j = \sum_{i \in D \setminus \{j\}} y_{ij} p_i + p_j$
- Not possible to consider start dependent costs $\implies$ objective $\sum_{i \in \mathcal{J}} w_i C_i$

Sequencing variables

Time-indexed models yield good LP relaxation but the number of variables can be huge $\Longrightarrow$ need to consider more compact models.

- Binary variable $y_{ij} = 1$ if and only if $S_j \geq S_i + p_i$
- Disjunctive case : $y_{ij} = 1 - y_{ji}, \forall (i,j) \in D$ (half of the variables can be dropped) and y_{ij} is the incidence vector of linear orderings :
- If there are no release dates and in the 1-machine case $C_j = \sum_{i \in D \setminus \{j\}} y_{ij} p_i + p_j$
- Not possible to consider start dependent costs $\Longrightarrow$ objective $\sum_{i \in \mathcal{J}} w_i C_i$

Sequencing variables

Time-indexed models yield good LP relaxation but the number of variables can be huge $\Longrightarrow$ need to consider more compact models.

- Binary variable $y_{ij} = 1$ if and only if $S_j \geq S_i + p_i$
- Disjunctive case : $y_{ij} = 1 - y_{ji}$, $\forall (i,j) \in D$ (half of the variables can be dropped) and y_{ij} is the incidence vector of linear orderings :
- If there are no release dates and in the 1-machine case $C_j = \sum_{i \in D \setminus \{j\}} y_{ij} p_i + p_j$
- Not possible to consider start dependent costs $\Longrightarrow$ objective $\sum_{i \in \mathcal{J}} w_i C_i$

Sequencing variables

Time-indexed models yield good LP relaxation but the number of variables can be huge $\implies$ need to consider more compact models.

- Binary variable $y_{ij} = 1$ if and only if $S_j \geq S_i + p_i$
- Disjunctive case : $y_{ij} = 1 - y_{ji}$, $\forall (i,j) \in D$ (half of the variables can be dropped) and y_{ij} is the incidence vector of linear orderings :
- If there are no release dates and in the 1-machine case $C_j = \sum_{i \in D \setminus \{j\}} y_{ij} p_i + p_j$
- Not possible to consider start dependent costs $\implies$ objective $\sum_{i \in \mathcal{J}} w_i C_i$

Sequencing variables

Time-indexed models yield good LP relaxation but the number of variables can be huge $\Longrightarrow$ need to consider more compact models.

- Binary variable $y_{ij} = 1$ if and only if $S_j \geq S_i + p_i$
- Disjunctive case : $y_{ij} = 1 - y_{ji}$, $\forall (i,j) \in D$ (half of the variables can be dropped) and y_{ij} is the incidence vector of linear orderings :
- If there are no release dates and in the 1-machine case $C_j = \sum_{i \in D \setminus \{j\}} y_{ij} p_i + p_j$
- Not possible to consider start dependent costs $\Longrightarrow$ objective $\sum_{i \in \mathcal{J}} w_i C_i$

Sequencing variables : simplest case

Simple case (one-machine, no release dates, no precedence constraints)

- Consider y_{ij} for $i < j \implies C_j = \sum_{i<j} y_{ij} p_i + \sum_{i>j} (1 - y_{ji}) p_i + p_j$
- $\sum_j w_j C_j = \sum_{1 \leq i < j \leq n} (w_j p_i - w_i p_j) y_{ij} + \sum_{1 \leq i < j \leq n} w_i p_j + \sum_{j \in \mathcal{J}} w_j p_j$
- The objective is easily optimized by LP :
  $\min \sum_{1 \leq i < j \leq n} (w_j p_i - w_i p_j) y_{ij}$
  $0 \leq y_{ij} \leq 1 \quad \forall i \in \mathcal{J}, \forall \in \mathcal{J}, j > i$
- It suffices to fix $y_{ij} = 1$ if $w_j p_i - w_i p_j \leq 0$ and to 0 otherwise $\implies$ WSPT rule [QS94].

Sequencing variables : simplest case

Simple case (one-machine, no release dates, no precedence constraints)

- Consider y_{ij} for $i < j \implies C_j = \sum_{i<j} y_{ij} p_i + \sum_{i>j} (1 - y_{ji}) p_i + p_j$
- $\sum_j w_j C_j = \sum_{1 \le i < j \le n} (w_j p_i - w_i p_j) y_{ij} + \sum_{1 \le i < j \le n} w_i p_j + \sum_{j \in \mathcal{J}} w_j p_j$
- The objective is easily optimized by LP :
  $\min \sum_{1 \le i < j \le n} (w_j p_i - w_i p_j) y_{ij}$
  $0 \le y_{ij} \le 1 \quad \forall i \in \mathcal{J}, \forall \in \mathcal{J}, j > i$
- It suffices to fix $y_{ij} = 1$ if $w_j p_i - w_i p_j \le 0$ and to 0 otherwise $\implies$ WSPT rule [QS94].

Sequencing variables : simplest case

Simple case (one-machine, no release dates, no precedence constraints)

- Consider y_{ij} for $i < j \implies C_j = \sum_{i<j} y_{ij} p_i + \sum_{i>j}(1 - y_{ji}) p_i + p_j$
- $\sum_j w_j C_j = \sum_{1 \leq i < j \leq n}(w_j p_i - w_i p_j) y_{ij} + \sum_{1 \leq i < j \leq n} w_i p_j + \sum_{j \in \mathcal{J}} w_j p_j$
- The objective is easily optimized by LP :
  $\min \sum_{1 \leq i < j \leq n}(w_j p_i - w_i p_j) y_{ij}$
  $0 \leq y_{ij} \leq 1 \quad \forall i \in \mathcal{J}, \forall \in \mathcal{J}, j > i$
- It suffices to fix $y_{ij} = 1$ if $w_j p_i - w_i p_j \leq 0$ and to 0 otherwise $\implies$ WSPT rule [QS94].

Sequencing variables : simplest case

Simple case (one-machine, no release dates, no precedence constraints)

- Consider y_{ij} for $i < j \implies C_j = \sum_{i<j} y_{ij} p_i + \sum_{i>j} (1 - y_{ji}) p_i + p_j$
- $\sum_j w_j C_j = \sum_{1 \leq i < j \leq n} (w_j p_i - w_i p_j) y_{ij} + \sum_{1 \leq i < j \leq n} w_i p_j + \sum_{j \in \mathcal{J}} w_j p_j$
- The objective is easily optimized by LP :
  $\min \sum_{1 \leq i < j \leq n} (w_j p_i - w_i p_j) y_{ij}$
  $0 \leq y_{ij} \leq 1 \quad \forall i \in \mathcal{J}, \forall \in \mathcal{J}, j > i$
- It suffices to fix $y_{ij} = 1$ if $w_j p_i - w_i p_j \leq 0$ and to 0 otherwise $\implies$ WSPT rule [QS94].

Sequencing variables : one machine, precedence constraints (1)

Set E of precedence constraints
(MILP due to [Pot80])

min $\sum_{i,j \in \mathcal{J}} p_i w_j y_{ij} + \sum_{j \in \mathcal{J}} p_j w_j$

$$
\begin{aligned}
y_{ij} + y_{ji} &= 1 & \forall i, j \in \mathcal{J}, i \neq j \\
y_{ij} + y_{jk} - y_{ik} &\leq 1 & \forall i, j, k \in \mathcal{J}, i \neq j \neq k \\
y_{ij} &= 1 & \forall (i, j) \in E \\
y_{ij} &= 0 & \forall (j, i) \in E \\
y_{ij} &\in \{0, 1\} & \forall i, j \in \mathcal{J}, i \neq j
\end{aligned}
$$

(NP-hard)

Sequencing variables : one machine, precedence constraints (2)

A vertex cover LP-relaxation. Let $i \leftrightarrow$j denote $\{(i,j),(j,i)\} \cap E = \emptyset$

min $\sum_{i,j \in \mathcal{J}} p_i w_j y_{ij} + \sum_{j \in \mathcal{J}} p_j w_j$

$$\begin{aligned}
y_{ij} + y_{ji} &\geq 1 & \forall i,j \in \mathcal{J}, i \neq j, i \leftrightarrow j \\
y_{ik} + y_{kj} &\geq 1 & \forall (i,j) \in E, \forall k \in \mathcal{J}, i \leftrightarrow k, k \leftrightarrow j \\
y_{il} + y_{kj} &\geq 1 & \forall (i,j),(k,l) \in E, i \leftrightarrow l, j \leftrightarrow k \\
y_{ij} &\geq 0 & \forall i,j \in \mathcal{J}, i \neq j, i \leftrightarrow j
\end{aligned}$$

For Series-Parallel precedence constraints, the polyhedron is integral and yields a feasible solution [CS04]

Sequencing and natural date variables : one machine, release dates

Jobs have release dates r_i, assuming $r_1 \leq r_2 \leq \ldots \leq r_n$
Idle times $\implies$ need to consider natural date variables S_i

Sequencing and natural date variables : one machine, release dates

Jobs have release dates r_i, assuming $r_1 \leq r_2 \leq \ldots \leq r_n$
Idle times $\implies$ need to consider natural date variables S_i

MILP by [NS92] (improving [DW90] inequalities) linearizing :

$$S_j \geq (r_i + p_i) y_{ij} + \sum_{k \neq i,j} p_k y_{ik} y_{kj} \quad \forall i,j \in \mathcal{J}, i \neq j$$

Sequencing and natural date variables : one machine, release dates

Jobs have release dates r_i, assuming $r_1 \leq r_2 \leq \ldots \leq r_n$
Idle times $\implies$ need to consider natural date variables S_i

MILP by [NS92] (improving [DW90] inequalities) linearizing :

$$S_j \geq (r_i + p_i)y_{ij} + \sum_{k \neq i,j} p_k y_{ik} y_{kj} \quad \forall i, j \in \mathcal{J}, i \neq j$$

$$\min \textstyle\sum_{j \in \mathcal{J}} w_j S_j$$

$$\begin{aligned}
S_j \geq (r_i+p_i)y_{ij}+\textstyle\sum_{k<i,k\neq j} p_k(y_{ik}-y_{jk})+\sum_{k>i,k\neq j} p_k y_{kj} & \quad \forall i,j \in \mathcal{J}, i \neq j \\
S_j \geq r_i & \quad \forall i \in \mathcal{J} \\
y_{ij} + y_{ji} = 1 & \quad \forall 1 \leq i < j \leq n \\
y_{ij} + y_{jk} - y_{ik} \leq 1 & \quad \forall i,j,k \in \mathcal{J}, i \neq j \neq k \\
\delta_{jk} \in \{0,1\} & \quad \forall i,j \in \mathcal{J}, i \neq j
\end{aligned}$$

Sequencing and natural date variables : the general disjunctive scheduling problem

$\min \sum_{j \in \mathcal{J}} w_j S_j$

$$
\begin{aligned}
S_j &\geq r_i & \forall i \in \mathcal{J} \\
S_j - S_i + M_{ij}(1 - y_{ij}) &\geq p_i & \forall (i,j) \in D \\
y_{ij} + y_{ji} &= 1 & \forall (i,j) \in D \\
y_{ij} + y_{jk} - y_{ik} &\leq 1 & \forall (i,j),(j,k),(k,i) \in D, i \neq j \neq k \\
S_j - S_i &\geq l_{ij} & \forall (i,j) \in E \\
y_{ij} &\in \{0,1\} & \forall (i,j) \in D
\end{aligned}
$$

with M_{ij} an upper bound on $S_i + p_i - S_j$, e.g. $\tilde{d}_i - r_j$ in case of deadlines.
($n(n-1)/2$ integer variables, n continuous variables)

Sequencing and natural date variables : RCPSP

Forbidden set $F \in \mathcal{F}$: $\exists k \in \mathcal{R}$, $\sum_{j \in F} b_{jk} > B_k$. $\min \sum_{j \in \mathcal{J}} w_j S_j$
MIP issued from [AVT93]

$$
\begin{aligned}
S_j &\geq r_i && \forall i \in \mathcal{J} \\
S_j - S_i + M_{ij}(1 - y_{ij}) &\geq p_i && \forall (i,j) \in D \\
\sum_{i,j \in F, i \neq j} y_{ij} &\geq 1 && \forall F \in \mathcal{F} \\
y_{ij} + y_{ji} &\leq 1 && \forall i,j \in \mathcal{J} \\
y_{ij} + y_{ji} &= 1 && \forall (i,j) \in D \\
y_{ij} + y_{jk} - y_{ik} &\leq 1 && \forall (i,j),(j,k),(ki) \in D, i \neq j \neq k \\
S_j - S_i &\geq l_{ij} && \forall (i,j) \in E \\
y_{ij} &\in \{0,1\} && \forall (i,j) \in D
\end{aligned}
$$

Let $\mathcal{SY}$ the set of feasible (S,y) vectors.
($n(n-1)$ binary variables, n continuous variables, $|\mathcal{F}|$??)

Sequencing and natural date variables : compact model for the RCPSP

Replace exponential number of constraints
$\sum_{i,j \in F, i \neq j} y_{ij} \geq 1 \quad \forall F \in \mathcal{F}$
by resource flow networks (0 dummy source job and $n+1$ dummy sink job)

Sequencing and natural date variables : compact model for the RCPSP

Replace exponential number of constraints
$\sum_{i,j \in F, i \neq j} y_{ij} \geq 1 \quad \forall F \in \mathcal{F}$
by resource flow networks (0 dummy source job and $n+1$ dummy sink job)

$$\sum_{j \in \mathcal{J}} f_{0jk} = \sum_{j \in \mathcal{J}} f_{j,n+1,k} = B_k \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{R}$$

$$\sum_{j \neq i} f_{ijk} = \sum_{j \neq i} f_{jik} = b_{ik} \qquad \forall i \in \mathcal{J}, \forall k \in \mathcal{R}$$

$$0 \leq f_{ijk} \leq \min(b_{ik}, b_{jk}) y_{ij} \qquad \forall i, j \in \mathcal{J}, i \neq j, \forall k \in \mathcal{R}$$

Sequencing and natural date variables : compact model for the RCPSP

Replace exponential number of constraints
$\sum_{i,j \in F, i \neq j} y_{ij} \geq 1 \quad \forall F \in \mathcal{F}$
by resource flow networks (0 dummy source job and $n+1$ dummy sink job)

$$\sum_{j \in \mathcal{J}} f_{0jk} = \sum_{j \in \mathcal{J}} f_{j,n+1,k} = B_k \qquad \forall j \in \mathcal{J}, \forall k \in \mathcal{R}$$

$$\sum_{j \neq i} f_{ijk} = \sum_{j \neq i} f_{jik} = b_{ik} \qquad \forall i \in \mathcal{J}, \forall k \in \mathcal{R}$$

$$0 \leq f_{ijk} \leq \min(b_{ik}, b_{jk}) y_{ij} \qquad \forall i,j \in \mathcal{J}, i \neq j, \forall k \in \mathcal{R}$$

A compact model is obtained A. et al. [AMR00].

(Facet-inducing) valid inequalities for the RCPSP polyhedron

Polyhedral study for the forbidden set-based formulation, Alvarez-Valdés et al. [AVT93]

- let $Q = conv(\{(S, y) \in \mathbb{R}^{n^2} | (S, y) \in \mathcal{SY}\})$ (the RCPSP polyhedron)
- The dimension of Q is $d_Q = n^2 - |E^*| - |D|$.
- Example : $y_{ij} + y_{ji} \leq 1$ (a) induces a facet of Q if $(i, j) \notin |E^*| \cup D$
  - (a) is a face of Q as (i) we can find a solution $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = 1$, $\tilde{y}_{ji} = 0$ and $(\tilde{S}, \tilde{y}) \in Q \cap \{(S, y) | y_{ij} + y_{ji} = 1\}$ (possible if the time horizon is sufficiently large) (ii) we can find solutions $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = \tilde{y}_{ji} = 0$ with $(\tilde{S}, \tilde{y}) \notin \{(S, y) | y_{ij} + y_{ji} = 1\}$.
  - (a) is of dimension $d_Q - 1$ as by setting $y_{ij} + y_{ji} = 1$ we increment $|D|$

(Facet-inducing) valid inequalities for the RCPSP polyhedron

Polyhedral study for the forbidden set-based formulation, Alvarez-Valdés et al. [AVT93]

- let $Q = conv(\{(S,y) \in \mathbb{R}^{n^2} | (S,y) \in \mathcal{SY}\})$ (the RCPSP polyhedron)
- The dimension of Q is $d_Q = n^2 - |E^*| - |D|$.
- Example : $y_{ij} + y_{ji} \leq 1$ (a) induces a facet of Q if $(i,j) \notin |E^*| \cup D$
  - (a) is a face of Q as (i) we can find a solution $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = 1$, $\tilde{y}_{ji} = 0$ and $(\tilde{S}, \tilde{y}) \in Q \cap \{(S,y) | y_{ij} + y_{ji} = 1\}$ (possible if the time horizon is sufficiently large) (ii) we can find solutions $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = \tilde{y}_{ji} = 0$ with $(\tilde{S}, \tilde{y}) \notin \{(S,y) | y_{ij} + y_{ji} = 1\}$.
  - (a) is of dimension $d_Q - 1$ as by setting $y_{ij} + y_{ji} = 1$ we increment $|D|$

(Facet-inducing) valid inequalities for the RCPSP polyhedron

Polyhedral study for the forbidden set-based formulation, Alvarez-Valdés et al. [AVT93]

- let $Q = conv(\{(S,y) \in \mathbb{R}^{n^2} | (S,y) \in \mathcal{SY}\})$ (the RCPSP polyhedron)
- The dimension of Q is $d_Q = n^2 - |E^*| - |D|$.
- Example : $y_{ij} + y_{ji} \leq 1$ (a) induces a facet of Q if $(i,j) \notin |E^*| \cup D$
  - (a) is a face of Q as (i) we can find a solution $(\tilde{S},\tilde{y})$ with $\tilde{y}_{ij} = 1$, $\tilde{y}_{ji} = 0$ and $(\tilde{S},\tilde{y}) \in Q \cap \{(S,y) | y_{ij} + y_{ji} = 1\}$ (possible if the time horizon is sufficiently large) (ii) we can find solutions $(\tilde{S},\tilde{y})$ with $\tilde{y}_{ij} = \tilde{y}_{ji} = 0$ with $(\tilde{S},\tilde{y}) \notin \{(S,y) | y_{ij} + y_{ji} = 1\}$.
  - (a) is of dimension $d_Q - 1$ as by setting $y_{ij} + y_{ji} = 1$ we increment $|D|$

(Facet-inducing) valid inequalities for the RCPSP polyhedron

Polyhedral study for the forbidden set-based formulation, Alvarez-Valdés et al. [AVT93]

- let $Q = conv(\{(S,y) \in \mathbb{R}^{n^2} | (S,y) \in \mathcal{SY}\})$ (the RCPSP polyhedron)
- The dimension of Q is $d_Q = n^2 - |E^*| - |D|$.
- Example : $y_{ij} + y_{ji} \leq 1$ (a) induces a facet of Q if $(i,j) \notin |E^*| \cup D$
  - (a) is a face of Q as (i) we can find a solution $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = 1$, $\tilde{y}_{ji} = 0$ and $(\tilde{S}, \tilde{y}) \in Q \cap \{(S,y) | y_{ij} + y_{ji} = 1\}$ (possible if the time horizon is sufficiently large) (ii) we can find solutions $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = \tilde{y}_{ji} = 0$ with $(\tilde{S}, \tilde{y}) \notin \{(S,y) | y_{ij} + y_{ji} = 1\}$.
  - (a) is of dimension $d_Q - 1$ as by setting $y_{ij} + y_{ji} = 1$ we increment $|D|$

(Facet-inducing) valid inequalities for the RCPSP polyhedron

Polyhedral study for the forbidden set-based formulation, Alvarez-Valdés et al. [AVT93]

- let $Q = conv(\{(S, y) \in \mathbb{R}^{n^2} | (S, y) \in \mathcal{SY}\})$ (the RCPSP polyhedron)
- The dimension of Q is $d_Q = n^2 - |E^*| - |D|$.
- Example : $y_{ij} + y_{ji} \leq 1$ (a) induces a facet of Q if $(i, j) \notin |E^*| \cup D$
  - (a) is a face of Q as (i) we can find a solution $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = 1$, $\tilde{y}_{ji} = 0$ and $(\tilde{S}, \tilde{y}) \in Q \cap \{(S, y) | y_{ij} + y_{ji} = 1\}$ (possible if the time horizon is sufficiently large) (ii) we can find solutions $(\tilde{S}, \tilde{y})$ with $\tilde{y}_{ij} = \tilde{y}_{ji} = 0$ with $(\tilde{S}, \tilde{y}) \notin \{(S, y) | y_{ij} + y_{ji} = 1\}$.
  - (a) is of dimension $d_Q - 1$ as by setting $y_{ij} + y_{ji} = 1$ we increment $|D|$

More valid inequalities for disjunctive scheduling and the RCPSP

- Let $\mathcal{C}$ denote the set of cliques in D. Applegate and Cook [AC91] derived cuts for the job-shop problem from one-machine scheduling valid inequalities (especially [DW90] inequalities) such as

$$S_j \geq r_j + \sum_{i \in C} y_{ij} p_i \quad \forall C \in \mathcal{C}, \forall j \in C$$

$$S_j \geq r_k + \sum_{i \in C \setminus \{j\}} y_{ij} p_i - \sum_{i \in C} y_{ik}(r_l - r_j) \quad \forall C \in \mathcal{C}, \forall j \in C,$$

- Alvarez-Valdés et al. [AVT93] also proposed such cuts for the RCPSP. Let $\Gamma^{-1}(i)$ ($\Gamma(i)$) be the set of ancestors (descendants) of i in E.

$$S_j \geq S_i + p_i + \sum_{k \in \Gamma(i)} p_k y_{kj} + \sum_{k \in \Gamma^{-1}(j)} p_k y_{ik} + \sum_{\substack{k \in D(i) \cup D(j) \\ k \notin \Gamma(i) \cup \Gamma^{-1}(j)}} p_k (y_{ik} + y_{kj} - 1)$$

More valid inequalities for disjunctive scheduling and the RCPSP

- Let $\mathcal{C}$ denote the set of cliques in D. Applegate and Cook [AC91] derived cuts for the job-shop problem from one-machine scheduling valid inequalities (especially [DW90] inequalities) such as

$$S_j \geq r_j + \sum_{i \in C} y_{ij} p_i \quad \forall C \in \mathcal{C}, \forall j \in C$$

$$S_j \geq r_k + \sum_{i \in C \setminus \{j\}} y_{ij} p_i - \sum_{i \in C} y_{ik}(r_l - r_j) \quad \forall C \in \mathcal{C}, \forall j \in C,$$

- Alvarez-Valdés et al. [AVT93] also proposed such cuts for the RCPSP. Let $\Gamma^{-1}(i)$ ($\Gamma(i)$) be the set of ancestors (descendants) of i in E.

$$S_j \geq S_i + p_i + \sum_{k \in \Gamma(i)} p_k y_{kj} + \sum_{k \in \Gamma^{-1}(j)} p_k y_{ik} + \sum_{\substack{k \in D(i) \cup D(j) \\ k \notin \Gamma(i) \cup \Gamma^{-1}(j)}} p_k (y_{ik} + y_{kj} - 1)$$

Constraint-propagation-based cutting planes for the RCPSP

- So far the studied valid inequalities ignore the existence of a tight upper bound.
- Compute UB with an efficient heuristic and update distance matrix d_{ij} through constraint propagation (where d_{ij} is a lower bound of $S_j - S_i$).
- Cuts can be derived from such updates. d^c_{ij} denotes the value of d_{ij} if constraint c is satisfied. We consider constraints such as $k||h$, $k \prec h$, $k \succ h$.
  - Example 1 : $S_j - S_i \geq d_{ij}^{k||h} + d_{ij}^{k \prec h} y_{kh} + d_{ij}^{k \succ h} y_{hk}$, for any 4 jobs i, j, k, h.
  - Example 2 : $x_{ij} \geq x_{hk}$, $\forall i, j, h, i$, $d_{ij}^{h \prec k} \geq p_i$

Demassey, A. and Michelon [DAM05]

Constraint-propagation-based cutting planes for the RCPSP

- So far the studied valid inequalities ignore the existence of a tight upper bound.
- Compute UB with an efficient heuristic and update distance matrix d_{ij} through constraint propagation (where d_{ij} is a lower bound of $S_j - S_i$).
- Cuts can be derived from such updates. d_{ij}^{c} denotes the value of d_{ij} if constraint c is satisfied. We consider constraints such as $k||h$, $k \prec h$, $k \succ h$.
  - Example 1 : $S_j - S_i \geq d_{ij}^{k||h} + d_{ij}^{k \prec h} y_{kh} + d_{ij}^{k \succ h} y_{hk}$, for any 4 jobs i, j, k, h.
  - Example 2 : $x_{ij} \geq x_{hk}$, $\forall i, j, h, i$, $d_{ij}^{h \prec k} \geq p_i$

Demassey, A. and Michelon [DAM05]

Constraint-propagation-based cutting planes for the RCPSP

- So far the studied valid inequalities ignore the existence of a tight upper bound.
- Compute UB with an efficient heuristic and update distance matrix d_{ij} through constraint propagation (where d_{ij} is a lower bound of $S_j - S_i$).
- Cuts can be derived from such updates. d_{ij}^c denotes the value of d_{ij} if constraint c is satisfied. We consider constraints such as $k||h$, $k \prec h$, $k \succ h$.
  - Example 1 : $S_j - S_i \geq d_{ij}^{k||h} + d_{ij}^{k \prec h} y_{kh} + d_{ij}^{k \succ h} y_{hk}$, for any 4 jobs i, j, k, h.
  - Example 2 : $x_{ij} \geq x_{hk}$, $\forall i, j, h, i$, $d_{ij}^{h \prec k} \geq p_i$

Demassey, A. and Michelon [DAM05]

Constraint-propagation-based cutting planes for the RCPSP

- So far the studied valid inequalities ignore the existence of a tight upper bound.
- Compute UB with an efficient heuristic and update distance matrix d_{ij} through constraint propagation (where d_{ij} is a lower bound of $S_j - S_i$).
- Cuts can be derived from such updates. d_{ij}^c denotes the value of d_{ij} if constraint c is satisfied. We consider constraints such as $k||h$, $k \prec h$, $k \succ h$.
    - Example 1 : $S_j - S_i \geq d_{ij}^{k||h} + d_{ij}^{k\prec h} y_{kh} + d_{ij}^{k \succ h} y_{hk}$, for any 4 jobs i, j, k, h.
    - Example 2 : $x_{ij} \geq x_{hk}$, $\forall i, j, h, i$, $d_{ij}^{h \prec k} \geq p_i$

Demassey, A. and Michelon [DAM05]

Constraint-propagation-based cutting planes for the RCPSP

- So far the studied valid inequalities ignore the existence of a tight upper bound.
- Compute UB with an efficient heuristic and update distance matrix d_{ij} through constraint propagation (where d_{ij} is a lower bound of $S_j - S_i$).
- Cuts can be derived from such updates. d_{ij}^c denotes the value of d_{ij} if constraint c is satisfied. We consider constraints such as $k||h$, $k \prec h$, $k \succ h$.
  - Example 1 : $S_j - S_i \geq d_{ij}^{k||h} + d_{ij}^{k \prec h} y_{kh} + d_{ij}^{k \succ h} y_{hk}$, for any 4 jobs i, j, k, h.
  - Example 2 : $x_{ij} \geq x_{hk}$, $\forall i, j, h, i$, $d_{ij}^{h \prec k} \geq p_i$

Demassey, A. and Michelon [DAM05]

Outline

Flow and natural date variables

For one-machine and parallel machine problem, sequencing variables y_{ij} may be replaced by flow variables $f_{ij} \in \{0,1\}$ stating that i is an immediate predecessor of j in the sequence.

min $\sum_{i \in \mathcal{J}} w_i S_i$

$$\begin{aligned}
\sum_{j \in \mathcal{J}} f_{0j} = \sum_{j \in \mathcal{J}} f_{j,n+1} &= m && \\
\sum_{j \in \mathcal{J} \setminus \{i\}} f_{ij} = \sum_{j \in \mathcal{J} \setminus \{i\}} f_{ji} &= 1 && \forall i \in \mathcal{J} \\
S_i &\geq r_i && \forall i \in \mathcal{J} \\
S_j - S_i + M_{ij}(1 - f_{ij}) &\geq p_i && \forall i,j \in \mathcal{J}, i \neq j \\
S_j - S_i &\geq l_{ij} && \forall (i,j) \in E \\
f_{ij} &\in \{0,1\} && \forall i,j \in \mathcal{J}, i \neq j
\end{aligned}$$

When there are no relase dates, machine assignment and subtour elimitation constraints can be used to avoid big$-M$ coefficients.

Flow and natural date variables (variant)

Big-M constraints can be removed, by considering continuous variable $S_{ij} \geq 0$ which equals 0 unless i is the immediate predecessor of j.

$$\min \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J} \setminus \{i\}} w_i S_{ij}$$

$$\sum_{j \in \mathcal{J}} f_{0j} = \sum_{j \in \mathcal{J}} f_{j,n+1} = m$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} f_{ij} = \sum_{j \in \mathcal{J} \setminus \{i\}} f_{ji} = 1 \qquad \forall i \in \mathcal{J}$$

$$r_i f_{ij} \leq S_{ij} \leq \tilde{d}_i f_{ij} \qquad \forall i, j \in \mathcal{J}, i \neq j$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} S_{ij} - \sum_{k \in \mathcal{J} \setminus \{i\}} (S_{ki} + p_k f_{ki}) \geq 0 \qquad \forall i \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J} \setminus \{j\}} S_{jk} - \sum_{k \in \mathcal{J} \setminus \{i\}} S_{ik} \geq l_{ij} \qquad \forall (i,j) \in E$$

$$f_{ij} \in \{0,1\}, S_{ij} \geq 0 \qquad \forall i, j \in \mathcal{J}, i \neq j$$

Outline

Positional date and assignment variables (1 mach)

Compact formulation without $big - M$ coefficients? Lasserre and Queyranne [LQ92]

- Time horizon is divided in a set $\mathcal{E}$ of n positions (or events).

Positional date and assignment variables (1 mach)

Compact formulation without $big - M$ coefficients ? Lasserre and Queyranne [LQ92]

- Time horizon is divided in a set $\mathcal{E}$ of n positions (or events).
- Positional dates $t_e, f_e \geq 0$, for each $e \in \mathcal{E}$ start, end of event e (n var.)

Positional date and assignment variables (1 mach)

Compact formulation without $big - M$ coefficients ? Lasserre and Queyranne [LQ92]

- Time horizon is divided in a set $\mathcal{E}$ of n positions (or events).
- Positional dates $t_e, f_e \geq 0$, for each $e \in \mathcal{E}$ start, end of event e (n var.)
- Assignment variable $a_{ie} \in \{0, 1\}$ of job j to event j (n^2 var.)

Positional date and assignment variables (1 mach)

Compact formulation without $big - M$ coefficients ? Lasserre and Queyranne [LQ92]

- Time horizon is divided in a set $\mathcal{E}$ of n positions (or events).
- Positional dates $t_e, f_e \geq 0$, for each $e \in \mathcal{E}$ start, end of event e (n var.)
- Assignment variable $a_{ie} \in \{0,1\}$ of job j to event j (n^2 var.)

$$\min \sum_{e \in \mathcal{E}} w_i C_i$$

$$\sum_{e \in \mathcal{E}} a_{je} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} a_{je} = 1 \qquad \forall e \in \mathcal{E}$$

$$t_e + \sum_{j \in \mathcal{J}} p_i a_{ie} = f_e \qquad \forall e \in \mathcal{E}$$

$$t_e - \sum_{j \in \mathcal{J}} r_j a_{je} \geq 0 \qquad \forall e \in \mathcal{E}$$

$$t_e \geq t_{e-1} \qquad \forall e \in \mathcal{E}$$

$$C_i + {\color{red}M}(1 - a_{ie}) \geq f_e \qquad \forall i \in \mathcal{J}, \in \forall e \in \mathcal{E}$$

$$t_e \geq 0 \qquad \forall e \in \mathcal{E}$$

Positional date and assignment variables (RCPSP)

How many events needed ?

$\forall i \in \mathcal{J}$ either $S_i = 0$ or $\exists j \in \mathcal{J}$, $S_i = S_j + p_j \implies |\mathcal{E}| \leq n+1$

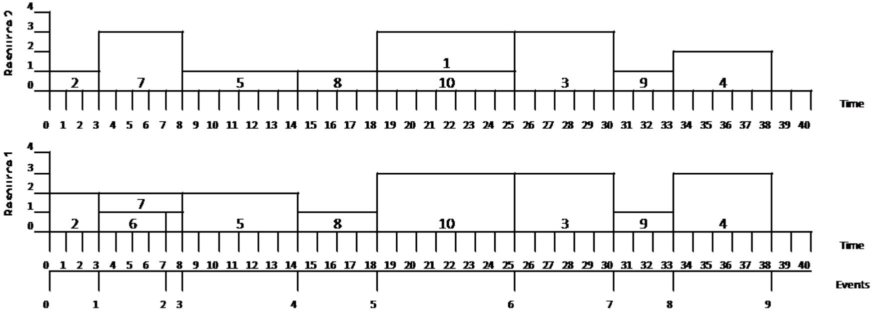
Start/End Event-based formulation (SEE) Koné, A., Lopez, Mongeau [KALM11]

- Variable $x_{ie} \in \{0,1\}$: job i starts at event e.
- Variable $y_{ie} \in \{0,1\}$: job i ends at event e.
- t_e time of event e
- $2n^2 + 2n$ binary variables, $(n+1)$ continuous variables

On/Off Event-based formulation (OOE) Koné, A., Lopez, Mongeau [KALM11]

- Variable $z_{ie} \in \{0,1\}$: z_{ie} is set to 1 if job i starts at event e or if it still being processed immediately after event e
- n^2 binary variables, $(n+1)$ continuous variables

Example



e	0	1	2	3
x_{6e}	0	1	0	0
y_{6e}	0	0	1	0
z_{6e}	0	1	0	0
x_{7e}	0	1	0	0
y_{7e}	0	0	0	1
z_{7e}	0	1	1	0

t	2	3	4	5	6	7	8
x_{6t}	0	1	0	0	0	0	0
x_{7t}	0	1	0	0	0	0	0

Start/End Event-based formulation (SEE)

$$\min t_n$$

$$t_0 = 0$$

$$t_f \geq t_e + p_i x_{ie} - p_i(1 - y_{if}) \qquad \forall (e,f) \in \mathcal{E}^2, f > e, \forall i \in \mathcal{J}$$

$$t_{e+1} \geq t_e \qquad \forall e \in \mathcal{E}, e < n$$

$$\sum_{e \in \mathcal{E}} x_{ie} = 1, \quad \sum_{e \in \mathcal{E}} y_{ie} = 1 \qquad \forall i \in \mathcal{J}$$

$$\sum_{v=0}^{e} y_{iv} + \sum_{v=e}^{n} x_{iv} \leq 1 \qquad \forall i \in \mathcal{J}, \forall e \in \mathcal{E}$$

$$\sum_{e'=e}^{n} y_{ie'} + \sum_{e'=0}^{e-1} x_{je'} \leq 1 \qquad \forall (i,j) \in E, \forall e \in \mathcal{E}$$

$$r_{0k} = \sum_{i \in A} b_{ik} x_{i0} \qquad \forall k \in \mathcal{R}$$

$$r_{ek} = r_{(e-1)k} + \sum_{i \in \mathcal{J}} b_{ik} x_{ie} - \sum_{i \in \mathcal{J}} b_{ik} y_{ie} \qquad \forall e \in \mathcal{E}, e \geq 1, k \in R$$

$$r_{ek} \leq B_k \qquad \forall e \in \mathcal{E}, k \in \mathcal{R}$$

$$x_{ie} \in \{0,1\}, y_{ie} \in \{0,1\} \qquad \forall i \in \mathcal{J} \cup \{0, n+1\}, \forall e \in \mathcal{E}$$

$$t_e \geq 0, r_{ek} \geq 0 \qquad \forall e \in \mathcal{E}, k \in \mathcal{R}.$$

On/Off Event-based formulation (OOE)

$$\min C_{\max}$$

$$C_{\max} \geq t_e + (z_{ie} - z_{i(e-1)})p_i \qquad \forall e \in \mathcal{E}, \forall i \in \mathcal{J}$$

$$t_0 = 0, t_{e+1} \geq t_e \qquad \forall e \neq n-1 \in \mathcal{E}$$

$$t_f \geq t_e + ((z_{i-e} - z_{i(e-1)}) - (z_{if} - z_{i(f-1)}) - 1)p_i \qquad \forall (e,f,i) \in \mathcal{E}^2 \times \mathcal{J}, f > e \neq 0$$

$$\sum_{e'=0}^{e-1} z_{ie'} \geq e(1 - (z_{ie} - z_{i(e-1)})), \ \sum_{e'=e}^{n-1} z_{ie'} \geq e(1 + (z_{ie} - z_{i(e-1)})) \qquad \forall e \neq 0 \in \mathcal{E}$$

$$\sum_{e \in \mathcal{E}} z_{ie} \geq 1 \qquad \forall i \in \mathcal{J}$$

$$z_{ie} + \sum_{e'=0}^{e} z_{je'} \leq 1 + (1 - z_{ie})e \qquad \forall e \in \mathcal{E}, \forall (i,j) \in E$$

$$\sum_{i=0}^{n-1} b_{ik} z_{ie} \leq B_k \qquad \forall e \in \mathcal{E}, \forall k \in \mathcal{R}$$

$$t_e \geq 0 \qquad \forall e \in \mathcal{E}$$

$$z_{ie} \in \{0,1\} \qquad \forall i \in \mathcal{J}, \forall e \in \mathcal{E}$$

Outline

Arc-time indexed formulations

Mixing flow-based and time-indexed formulations.

Arc-time indexed formulations

Mixing flow-based and time-indexed formulations.

$x_{ij}^t = 1$ if job j starts immediately at the end of job i at time t.

Arc-time indexed formulations

Mixing flow-based and time-indexed formulations.
$x_{ij}^t = 1$ if job j starts immediately at the end of job i at time t.

$$\min \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J} \setminus \{i\}} \sum_{t \in \mathcal{T}} w_j(t) x_{ij}^t$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} \sum_{t \in \mathcal{T}} x_{ij}^t = 1 \qquad \forall i \in \mathcal{J}$$

$$\sum_{i \in \mathcal{J}} x_{0i}^0 = m$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} x_{ji}^t - \sum_{j \in \mathcal{J} \setminus \{i\}} x_{ij}^{t+p_i} = 0 \qquad \forall i \in \mathcal{J}, \forall t \in \mathcal{T}$$

$$x_{ij}^t \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{J} \setminus \{i\}, \forall t \in \mathcal{T}$$

Arc-time indexed formulations

Mixing flow-based and time-indexed formulations.
$x_{ij}^t = 1$ if job j starts immediately at the end of job i at time t.

$$\min \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J} \setminus \{i\}} \sum_{t \in \mathcal{T}} w_j(t) x_{ij}^t$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} \sum_{t \in \mathcal{T}} x_{ij}^t = 1 \qquad \forall i \in \mathcal{J}$$

$$\sum_{i \in \mathcal{J}} x_{0i}^0 = m$$

$$\sum_{j \in \mathcal{J} \setminus \{i\}} x_{ji}^t - \sum_{j \in \mathcal{J} \setminus \{i\}} x_{ij}^{t+p_i} = 0 \qquad \forall i \in \mathcal{J}, \forall t \in \mathcal{T}$$

$$x_{ij}^t \in \{0,1\} \quad \forall i \in \mathcal{J}, \forall j \in \mathcal{J} \setminus \{i\}, \forall t \in \mathcal{T}$$

At least as strong LP relaxation as time-indexed one if variables x_{ii}^t are omitted except for $i = 0$ (dummy "depot" job) [PUPR10]

Outline

Column generation for time-indexed formulations

How to deal with large-horizons when using time-indexed formulations ? $\Longrightarrow$ Dantzig-Wolfe decomposition.

Column generation for time-indexed formulations

How to deal with large-horizons when using time-indexed formulations ? $\Longrightarrow$ Dantzig-Wolfe decomposition.

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt} \text{ (one-machine problem)}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \qquad \forall t \in \mathcal{T}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Column generation for time-indexed formulations

How to deal with large-horizons when using time-indexed formulations ? $\Longrightarrow$ Dantzig-Wolfe decomposition.

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt} \text{ (one-machine problem)}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \qquad \forall t \in \mathcal{T}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Let $\mathcal{PS} = \{x \in [0,1]^{n+T} \mid \sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1, \forall t \in \mathcal{T}\}$.

Column generation for time-indexed formulations

How to deal with large-horizons when using time-indexed formulations ? $\Longrightarrow$ Dantzig-Wolfe decomposition.

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt} \text{ (one-machine problem)}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1 \qquad \forall t \in \mathcal{T}$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Let $\mathcal{PS} = \{x \in [0,1]^{n+T} | \sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} x_{js} \leq 1, \forall t \in \mathcal{T}\}$.
$\mathcal{PS}$ is integral as the matrix is totally unimodular [vHS00]. Set of *pseudo-schedules*

Column generation for time-indexed formulations

We have $x = \sum_{q=1}^{r} \lambda_q a^q$, $\forall x$ extreme point of $conv(\{\mathcal{PS}\})$ where $\sum_{q=1}^{r} \lambda_q = 1$, $\lambda \geq 0$ and a^q is the q^{st} point of $\mathcal{PS}$.

Column generation for time-indexed formulations

We have $x = \sum_{q=1}^{r} \lambda_q a^q$, $\forall x$ extreme point of $conv(\{\mathcal{PS}\})$ where $\sum_{q=1}^{r} \lambda_q = 1$, $\lambda \geq 0$ and a^q is the q^{st} point of $\mathcal{PS}$.

Equivalent LP (of equal LP relaxation value)

$$\min \sum_{q=1}^{r} \left(\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q \right) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q \right) \lambda_q = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{q=1}^{r} \lambda_q = 1$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Column generation for time-indexed formulations

We have $x = \sum_{q=1}^{r} \lambda_q a^q$, $\forall x$ extreme point of $conv(\{\mathcal{PS}\})$ where $\sum_{q=1}^{r} \lambda_q = 1$, $\lambda \geq 0$ and a^q is the q^{st} point of $\mathcal{PS}$.

Equivalent LP (of equal LP relaxation value)

$$\min \sum_{q=1}^{r} \left(\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q \right) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q \right) \lambda_q = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{q=1}^{r} \lambda_q = 1$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Start with a restricted set of pseudo schedules and solve the LP relaxation by column generation.

Column generation for time-indexed formulations

$$\min \sum_{q=1}^{r} (\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q \right) \lambda_q = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{q=1}^{r} \lambda_q = 1$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Reduced cost of a pseudo schedule

$$\tilde{c}^q = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q$$

Column generation for time-indexed formulations

$$\min \sum_{q=1}^{r} \left(\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q\right) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q \right) \lambda_q = 1 \qquad \forall j \in \mathcal{J} (\pi_j)$$

$$\sum_{q=1}^{r} \lambda_q = 1 \qquad (\gamma)$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Reduced cost of a pseudo schedule

$$\tilde{c}^q = \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q - \sum_{j \in \mathcal{J}} \pi_j \left(\sum_{t \in \mathcal{T}} a_{jt}^q \right) - \gamma$$

Column generation for time-indexed formulations

$$\min \sum_{q=1}^{r} \left(\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q\right) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q\right) \lambda_q = 1 \qquad \forall j \in \mathcal{J} \ (\pi_j)$$

$$\sum_{q=1}^{r} \lambda_q = 1 \qquad (\gamma)$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Reduced cost of a pseudo schedule

$$\begin{aligned} \tilde{c}^q &= \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q - \sum_{j \in \mathcal{J}} \pi_j \left(\sum_{t \in \mathcal{T}} a_{jt}^q\right) - \gamma \\ &= \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (w_j(t) - \pi_j) a_{jt}^q - \gamma \end{aligned}$$

Column generation for time-indexed formulations

$$\min \sum_{q=1}^{r} \left(\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q\right) \lambda_q$$

$$\sum_{q=1}^{r} \left(\sum_{t \in \mathcal{T}} a_{jt}^q\right) \lambda_q = 1 \qquad \forall j \in \mathcal{J} \, (\pi_j)$$

$$\sum_{q=1}^{r} \lambda_q = 1 \qquad (\gamma)$$

$$0 \leq \lambda_q \leq 1 \qquad q = 1, \ldots, |\mathcal{T}|$$

Reduced cost of a pseudo schedule

$$\begin{aligned}\tilde{c}^q &= \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) a_{jt}^q - \sum_{j \in \mathcal{J}} \pi_j \left(\sum_{t \in \mathcal{T}} a_{jt}^q\right) - \gamma \\ &= \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (w_j(t) - \pi_j) a_{jt}^q - \gamma\end{aligned}$$

Finding a negative reduced cost variable amounts to find a shortest path in an acyclic graph with $O(nT)$ arcs.

Column generation for time-indexed RCPSP (1)

How to strengthen the time-indexed formulation ? Mingozzi et al. [MM98]

$$\min \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} w_j(t) x_{jt}$$

$$\sum_{t \in \mathcal{T}} x_{jt} = 1 \qquad \forall j \in \mathcal{J}$$

$$\sum_{j \in \mathcal{J}} \sum_{s=t-p_j+1}^{t} b_{jk} x_{js} \leq B_k \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{R}$$

$$\sum_{t=0}^{T} t x_{js} - \sum_{t=0}^{T} t x_{is} \geq l_{ij} \qquad \forall (i,j) \in E$$

$$x_{jt} \in \{0,1\} \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}$$

Integer Dantzig-Wolfe decomposition of resource constraints

Column generation for time-indexed RCPSP (2)

- Introduce additional variables

$$y \in \{0,1\}^{n+T} \begin{cases} \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \\ \sum_{s=0}^{T} y_{js} = p_j, \forall j \in \mathcal{J} \\ x_{jt} \geq y_{jt} - y_{j,t-1}, \forall j \in \mathcal{J} \end{cases}$$

- Perform an integer Danzig-Wolfe decomposition of $\mathcal{PS} = \left\{ y \in \{0,1\}^{n+T} \middle| \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \right\}$
- Subproblem is decomposable for each $t \in \mathcal{T}$ and each $t \in \mathcal{R}$n yielding mT multidimentional knapsack problems.
- Better LP relaxation than the time-indexed formulation, but practically intractable.
- Best known lower bounds for the RCPSP (before "SAT" results [Hor10]) where obtained by computing relaxation of this formulation (Brucker and Knust [BK00], Baptiste, A., Demassey Michelon [DABM04], Baptiste and Demassey[BD04]) integrating CP-based filtering and cuts.

Column generation for time-indexed RCPSP (2)

- Introduce additional variables

$$y \in \{0,1\}^{n+T} \begin{cases} \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \\ \sum_{s=0}^{T} y_{js} = p_j, \forall j \in \mathcal{J} \\ x_{jt} \geq y_{jt} - y_{j,t-1}, \forall j \in \mathcal{J} \end{cases}$$

- Perform an integer Danzig-Wolfe decomposition of

$$\mathcal{PS} = \left\{ y \in \{0,1\}^{n+T} \middle| \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \right\}$$

- Subproblem is decomposable for each $t \in \mathcal{T}$ and each $t \in \mathcal{R}$n yielding mT multidimentional knapsack problems.
- Better LP relaxation than the time-indexed formulation, but practically intractable.
- Best known lower bounds for the RCPSP (before "SAT" results [Hor10]) where obtained by computing relaxation of this formulation (Brucker and Knust [BK00], Baptiste, A., Demassey Michelon [DABM04], Baptiste and Demassey[BD04]) integrating CP-based filtering and cuts.

Column generation for time-indexed RCPSP (2)

- Introduce additional variables
$$y \in \{0,1\}^{n+T} \begin{cases} \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \\ \sum_{s=0}^{T} y_{js} = p_j, \forall j \in \mathcal{J} \\ x_{jt} \geq y_{jt} - y_{j,t-1}, \forall j \in \mathcal{J} \end{cases}$$
- Perform an integer Danzig-Wolfe decomposition of
$\mathcal{PS} = \left\{ y \in \{0,1\}^{n+T} \middle| \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \right\}$
- Subproblem is decomposable for each $t \in \mathcal{T}$ and each $t \in \mathcal{R}$n yielding mT multidimentional knapsack problems.
- Better LP relaxation than the time-indexed formulation, but practically intractable.
- Best known lower bounds for the RCPSP (before "SAT" results [Hor10]) where obtained by computing relaxation of this formulation (Brucker and Knust [BK00], Baptiste, A., Demassey Michelon [DABM04], Baptiste and Demassey[BD04]) integrating CP-based filtering and cuts.

Column generation for time-indexed RCPSP (2)

- Introduce additional variables
$$y \in \{0,1\}^{n+T} \begin{cases} \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \\ \sum_{s=0}^{T} y_{js} = p_j, \forall j \in \mathcal{J} \\ x_{jt} \geq y_{jt} - y_{j,t-1}, \forall j \in \mathcal{J} \end{cases}$$
- Perform an integer Danzig-Wolfe decomposition of
$\mathcal{PS} = \left\{ y \in \{0,1\}^{n+T} \middle| \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \right\}$
- Subproblem is decomposable for each $t \in \mathcal{T}$ and each $t \in \mathcal{R}$n yielding mT multidimentional knapsack problems.
- Better LP relaxation than the time-indexed formulation, but practically intractable.
- Best known lower bounds for the RCPSP (before "SAT" results [Hor10]) where obtained by computing relaxation of this formulation (Brucker and Knust [BK00], Baptiste, A., Demassey Michelon [DABM04], Baptiste and Demassey[BD04]) integrating CP-based filtering and cuts.

Column generation for time-indexed RCPSP (2)

- Introduce additional variables
$$y \in \{0,1\}^{n+T} \begin{cases} \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \\ \sum_{s=0}^{T} y_{js} = p_j, \forall j \in \mathcal{J} \\ x_{jt} \geq y_{jt} - y_{j,t-1}, \forall j \in \mathcal{J} \end{cases}$$
- Perform an integer Danzig-Wolfe decomposition of
$\mathcal{PS} = \left\{ y \in \{0,1\}^{n+T} \middle| \sum_{j\in\mathcal{J}} \sum_{s=0}^{T} b_{jk} y_{js} \leq B_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{R} \right\}$
- Subproblem is decomposable for each $t \in \mathcal{T}$ and each $t \in \mathcal{R}$n yielding mT multidimentional knapsack problems.
- Better LP relaxation than the time-indexed formulation, but practically intractable.
- Best known lower bounds for the RCPSP (before "SAT" results [Hor10]) where obtained by computing relaxation of this formulation (Brucker and Knust [BK00], Baptiste, A., Demassey Michelon [DABM04], Baptiste and Demassey[BD04]) integrating CP-based filtering and cuts.

More on Column generation for time-indexed RCPSP

- Can we combine strengthening the TI formulation and reduction of the number of variables ? (Cut and Column generation directly on the time-indexed formulation Sadykov and Vanderbeck [SV11] ?)
- Other decomposition schemes : partition the job in m subsets so that there is a feasible single machine schedule for each subset. van den Akker et al. [vHv99, vHv06, Mv07]

More on Column generation for time-indexed RCPSP

- Can we combine strengthening the TI formulation and reduction of the number of variables ? (Cut and Column generation directly on the time-indexed formulation Sadykov and Vanderbeck [SV11] ?)
- Other decomposition schemes : partition the job in m subsets so that there is a feasible single machine schedule for each subset. van den Akker et al. [vHv99, vHv06, Mv07]

Outline

Machine scheduling results (1)

- Efficiency of Sousa and Wolsey/van den Akker et al. cuts for lower bound computation
  (from van den Akker et al. [vHS99])

	LP		1		2	
$(n, p_{\max})$	G_{LP}^{av}	$G_{LP}^{\max}$	G_1^{av}	$G_1^{\max}$	G_2^{av}	$G_2^{\max}$
(20, 5)	0.379	1.346	0.157	1.228	0.058	0.572
(20,10)	0.64	1.959	0.233	1.337	0.054	0.407
(20,20)	0.507	1.657	0.126	0.966	0.047	0.385
(30, 5)	0.390	1.309	0.179	0.664	0.121	0.599
(30,10)	0.478	1.099	0.121	0.934	0.096	0.592

$(n, p_{\max})$	n_{LP}	n_1	n_2
(20, 5)	5	12	18
(20,10)	0	6	16
(20,20)	4	13	17
(30, 5)	5	6	8
(30,10)	0	5	9

Columns 1(2) : average or maximal gap for inequalities with rhs 1(2)
Columns n1(n2) : number of integer solutions found by cutting plane generation with inequalities with rhs 1 (2)

Machine scheduling results (2)

- Column-and-cut generation (Sadykov and Vanderbeck 2011)

n	Cplex for [R] cpu	Column generation for [M] it	sp	cpu	Column-and-row generation for [R] it	sp	$\%z$	cpu
25	11.2	343	343	2.1	208	69	5.8%	1.5
50	153.0	1270	1270	39.4	339	106	4.5%	16.9
100	2233.0	8784	8784	2891.5	466	139	4.5%	169.1

[R] Time-indexed formulation
[M] Dantzig-Wolfe decomposition of [vHS00]
Column cpu : time to solve the LP relaxation

Machine scheduling results results (3)

- Positional and sequencing based formulations cannot be discarded but have to be combined with heuristics (Hoogeveen and van de Velde [Hv95], Danna et al. [DRL05])
- Comparison of formulations for parallel machine scheduling Unlu and Mason [UM10]
  Recommendation is to use time-indexed models for small durations and flow-based model for other cases

RCPSP results : comparison of ILP formulations vs CP

Instances	Formulations	%Integer	%Opt	%Gap	%ΔCPM	Time Opt (s)
KSD30	DDT	91	82	0.47	8.91	10.45
	DT	86	78	0.55	6.74	12.76
	FCT	67	62	0.16	3.76	22.66
	OOE_Prec	46	30	1.69	13.65	52.31
	OOE	33	24	1.22	7.00	112.62
	SEE	3.1	2.9	0.24	0.61	123.62
	MCS	-	**97**	0.00	11.48	7.39
PACK	DDT	95	76	1.08	199.02	63.39
	DT	85	55	0.49	203.58	48.24
	OOE_Prec	55	5	3.25	227.19	18.92
	OOE	49	9	2.89	231.29	61.78
	FCT	2	0	1.28	14.49	-
	SEE	0	0	-	-	-
	MCS	-	25	0.00	149.81	115.88
BL	DDT	100	100	0.00	32.40	13.68
	DT	100	100	0.00	32.40	37.93
	OOE_Prec	54	0	7.26	40.30	-
	OOE	49	0	7.90	41.65	-
	FCT	21	3	6.14	30.64	310.58
	SEE	8	0	12.81	29.96	-
	MCS	-	**100**	0.00	32.40	3.29
KSD15_d	OEE_Prec	99.8	86	0.00	10.02	6.49
	FCT	99	94	0.02	9.02	12.06
	OEE	99	83	0.01	10.14	4.68
	SEE	92	76	0.15	9.86	13.04
	DT	55	54	0.23	4.31	12.10
	DDT	1	1	0.00	2.63	3.34
	MCS	-	**100**	0.00	10.18	0.07
PACK_d	OEE	60	18	1.26	120.13	75.58
	OOE_Prec	60	14	1.62	117.56	54.35
	FCT	7	7	0.00	0.00	60.88
	SEE	4	4	0.00	0.00	215.08
	DT	0	0		-	-
	DDT	0	0		-	-
	MCS	-	**38**	0.00	50.59	72.34

(from Kone et al.[KALM11])

- No formulation dominates the other, the accurate formulation has to be chosen depending on instance charactristics
- ILP formulations are not dominated by CP

RCPSP results : a cyclic example, MIP vs CP

Instances	n	DSP	hybrid/HD		hybrid/GS		CP	ILP+ (P_λ^+)		CG (DW_λ^+)		λ^0
		λ^{dsp}	λ^{hyb}	CPU_s	λ^{hyb}	CPU_s	λ^{CP}	λ^{ILP+}	CPU_s	λ^{CG}	CPU_s	
adpcm-st231.1	86	**80**	-	-	-	-	**80**	-	-	55	301	52
adpcm-st231.2	142	**139**	-	-	-	-	**139**	-	-	82	305	82
gsm-st231.1	30	30	29	2	**28**	2	**28**	**28***	256	25	8	24
gsm-st231.2	101	**93**	-	-	-	-	**93**	-	-	61	301	59
gsm-st231.5	44	**36**	**36**	10	**36**	17	-	**36***	3343	36	37	26
gsm-st231.6	30	**27**	**27**	3	**27**	4	**27**	**27***	7	27	3	17
gsm-st231.7	44	**41**	**41**	13	**41**	17	**41**	**41***	256	41	66	28
gsm-st231.8	14	**12**	**12**	0.3	**12**	0.3	-	**12***	0.6	12	$<$0.1	9
gsm-st231.9	34	**32**	**32**	2	34	4	**32**	**32***	62	31	12	28
gsm-st231.10	10	**8**	**8**	0.2	**8**	0.1	**8***	**8***	0.2	8	$<$0.1	6
gsm-st231.11	26	**24**	**24**	1	**24**	1	**24**	**24***	5	24	1.5	20
gsm-st231.12	15	**13**	**13**	0.3	**13**	0.4	**13**	**13***	0.7	13	$<$0.1	10
gsm-st231.13	46	43	43	168	**42**	440	43	41	11265	41	125	27
gsm-st231.14	39	34	34	6	34	10	34	**33***	3766	33	17	20
gsm-st231.15	15	**12**	**12**	0.3	**12**	0.3	**12**	**12***	0.9	12	$<$0.1	9
gsm-st231.16	65	**59**	**59**	145	**59**	144	60	58	8656	48	300	38
gsm-st231.17	38	**33**	**33**	202	-	-	**33**	32	12786	33	19	23
gsm-st231.18	214	194	-	-	-	-	**193**	-	-	-	-	120
gsm-st231.19	19	**15**	**15**	0.4	**15**	0.6	**15**	**15***	1.6	15	0.2	12
gsm-st231.20	23	**20**	**20**	1	**20**	1.4	**20**	**20***	17	20	0.9	13
gsm-st231.21	33	**30**	**30**	6	**30**	6	31	**30***	6105	29	7	20
gsm-st231.22	31	**29**	**29**	3	**29**	4	**29**	**29***	51	28	7	18
gsm-st231.25	60	**55**	-	-	**55**	75	57	**55***	11589	48	300	37
gsm-st231.29	44	**42**	**42**	13	**42**	15	**42**	**42***	63	42	68	28
gsm-st231.30	30	**25**	**25**	3	**25**	6	**25**	**25***	14	25	6	16
gsm-st231.31	44	**39**	**39**	13	**39**	17	**39**	**39***	833	39	59	26
gsm-st231.32	32	**30**	**30**	4	**30**	5	**30**	**30***	11	30	5	21
gsm-st231.33	59	52	-	-	-	-	**46**	45	9697	46	300	33
gsm-st231.34	10	8	**7**	$<$0.1	**7**	0.1	**7***	**7***	0.2	7	$<$0.1	6
gsm-st231.35	18	16	**14**	0.4	**14**	0.5	**14**	**14***	4	14	0.2	11
gsm-st231.36	31	29	**24**	2	**24**	10	**24**	**24***	321	24	4	18
gsm-st231.39	26	23	**21**	1.5	**21**	3	**21**	**21***	376	20	2	15
gsm-st231.40	21	17	**16**	0.5	18	1.3	17	**16***	28	16	0.6	12
gsm-st231.41	60	50	-	-	**47**	286	49	46	10069	46	300	34
gsm-st231.42	23	19	**18**	0.7	19	4	**18**	**18***	20	18	1	14
gsm-st231.43	26	23	21	2	22	2	**20**	**20***	101	20	2	15
#best(opt)		23	25		25		27(2)	27(27)				

From [AABH12], on a cyclic RCPSP with unit duration tasks, CP is able to find very good solutions but ILP is better at proving optimality.

Conclusion

Problems of increasing size are solved by MILP models with the help of

- Strong valid inequalities
- Efficient heuristics
- Benefits from hybrid-methods (CP/MILP/LS)

LP Lower bounds are often still too slow : cut-and-column generation perspective.

> *"In retrospect it is interesting to note that the original problem that started my research is still outstanding– namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty. If such a problem could be successfully solved it could (eventually through better planning) contribute to the well-being and stability of the world." (George Dantzig)*

References I

C. Artigues, M. A. Ayala, A. Benabid, and Claire Hanen, *Lower and upper bounds for the resource-constrained modulo scheduling problem*, 13th International Conference on Project Management and Scheduling, 2012, pp. 82–85.

D. Applegate and W. Cook, *A computational study of job-shop scheduling*, ORSA Journal on Computing **3** (1991), no. 2, 149–156.

C. Artigues, Ph. Michelon, and S. Reusser, *Insertion techniques for static and dynamic resource-constrained project scheduling*, European Journal of Operational Research **149** (2000), no. 2, 249–267.

References II

- R. Alvarez-Valdés and J. Tamarit, *The project scheduling polyhedron : Dimension, facets and lifting theorems*, European Journal of Operational Research **67** (1993), no. 2, 204–220.
- E. Balas, *On the facial structure of scheduling polyhedra*, Mathematical Programming Essays in Honor of George B. Dantzig Part I, Mathematical Programming Studies, vol. 24, Springer, 1985, pp. 179–218.
- P Baptiste and S Demassey, *Tight LP bounds for resource constrained project scheduling*, OR Spectrum **26** (2004), no. 2, 251–262.
- P. Brucker and S. Knust, *A linear programming and constraint propagation-based lower bound for the rcpsp*, European Journal of Operational Research **127** (2000), 355–362.

References III

J. R. Correa and A. S. Shulz, *Single machine scheduling with precedence constraints*, Integer Programming and Combinatoiral Optimization (D. Bienstock and G. Nemhauser, eds.), Lecture Notes in Computer Science, vol. 3064, 2004, pp. 145–168.

S. Demassey, C. Artigues, P. Baptiste, and P. Michelon, *Lagrangean relaxation-based lower bounds for the RCPSP*, 8th International Workshop on Project Management and Scheduling (Nancy,France), 2004, pp. 76–79.

S. Demassey, C. Artigues, and P. Michelon, *Constraint propagation-based cutting planes : an application to the rcpsp*, INFORMS Journal on Computing **17** (2005), no. 1, 52–65.

References IV

E. Danna, E. Rothberg, and C. Le Pape, *Exploring relaxation induced neighborhoods to improve mip solutions*, Mathematical Programming **102** (2005), no. 1, 71–90.

M.E. Dyer and L.A. Wolsey, *Formulating the single machine problem with release dates as a mixed integer program*, Discrete Applied Mathematics **26** (1990), 255–270.

M. X. Goemans, M. Queyranne, A. S. Shulz, M. Skutella, and Y. Wang, *Single machine scheduling with release dates*, SIAM Journal on Discrete Mathematics **15** (2002), 165–192.

J. R. Hardin, G. L. Nemhauser, and M. W. P. Savelsbergh, *Strong valid inequalities for the resource-constrained scheduling problem with uniform resource requirements*, Discrete Optimization **5** (2008), 19–35.

References V

- A. Horbach, *A boolean satisfiability approach to the resource-constrained project scheduling problem*, Annals of Operations Research **181** (2010), 89–107.
- H. Hoogeveen and S. L. van de Velde, *Formulating a scheduling problem with almost identical jobs by using positional completion times*, IPCO, 1995, pp. 292–306.
- O. Koné, C. Artigues, P. Lopez, and M. Mongeau, *Event-based milp models for resource-constrained project scheduling problems*, Computers and Operations Research **38** (2011), no. 1, 3–13.
- J.B. Lasserre and M. Queyranne, *Generic scheduling polyhedral and a new mixed integer formulation for single-machine scheduling*, Proceedings of the second IPCO conference (Carnegie-Mellon University Pittsburgh), 1992, pp. 136–149.

References VI

A. Mingozzi and V. Maniezzo, *An exact algorithm for the resource constrained project scheduling problem based on a new mathematical formulation*, Management Science **44** (1998), 714–729.

J. A. Hoogeveen M. van den Akker, Guido Diepen, *A column generation based destructive lower bound for resource constrained project scheduling problems*, CPAIOR, 2007, pp. 376–390.

G. Nemhauser and M. Savelsbergh, *A cutting plane algorithm for the single machine scheduling problem with release times*, Combinatorial Optimization : New Frontiers in the Theory and Practice (M. Akgul, H. Hamacher, and S. Tufecki, eds.), NATO ASI Series F : Computer and Systems Sciences, vol. 82, Springer-Verlag, 1992, pp. 63–84.

References VII

C.N. Potts, *An algorithm for the single machine sequencing problem with precedence constraints*, Mathematical Programming Study **13** (1980), 78–87.

A. Pessoa, E. Uchoa, M. Poggi de Aragão, and R. Rodrigues, *Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems*, Mathematical Programming Computation **2** (2010), 259–290.

M. Queyranne and A. S. Schulz, *Polyhedral approaches to machine scheduling*, Tech. Report 408/1994, TechnischeUniversitätBerlin, 1994.

References VIII

- ———, *Scheduling unit jobs with compatible release dates on parallel machines with nonstationary speeds*, nteger Programming and Combinatorial Optimization (IPCO 1995) (E. Balas and J. Clausen, eds.), Lecture Notes in Computer Science, vol. 920, 1995, pp. 307–320.
- M. Queyranne, *Structure of a simple scheduling polyhedron*, Mathematical Programming **58** (1993), 263–285.
- A. S. Schulz, *Polytopes and scheduling*, Ph.D. thesis, Technischen Universität Berlin, 1996.

References IX

- R. Sadykov and F. Vanderbeck, *Machine scheduling by column-and-row generation on the time-indexed formulation*, 10th International Workshop on Models and Algorithms for Planning and Scheduling Problems (Nymburk, Czech Republic), 2011, pp. 55–57.
- J. P. Sousa and L. A. Wolsey, *A time indexed formulation of non-preemptive single machine scheduling problems*, Mathematical Programming **54** (1992), 353–367.
- Y. Unlu and S. J. Mason, *Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems*, Computers & Industrial Engineering **58** (2010), no. 4, 785–800.

References X

J. M. van den Akker, C.P.M. Van Hoesel, and M.W.P. Savelsbergh, *A polyhedral approach to single-machine scheduling problems*, Mathematical Programming **85** (1999), 541–572.

M. van den Akker, C. A. J. Hurkens, and M. W. P. Savelsbergh, *Time-indexed formulations for machine scheduling problems : Column generation*, INFORMS Journal on Computing **12** (2000), no. 2, 111–124.

M. van den Akker, J. A. Hoogeveen, and S. L. van de Velde, *Parallel machine scheduling by column generation*, Operations Research **47** (1999), no. 6, 862–872.

M. van den Akker, J. A. Hoogeveen, and J. W. van Kempen, *Parallel machine scheduling through column generation : Minimax objective functions*, ESA, 2006, pp. 649–659.

References XI

H. Waterer, E.L. Johnson, P. Nobili, and M.W.P. Savelsbergh, *The relation of time indexed formulations of single machine scheduling problems to the node packing problem*, Mathematical Programming (2002), no. 93, 477–494.