Solving an integrated Job-Shop problem with human resource constraints

PMS’10 - Tours (France)

O. Guyon\textsuperscript{1,2}, P. Lemaire\textsuperscript{3}, É. Pinson\textsuperscript{2} and D. Rivreau\textsuperscript{2}

\textsuperscript{1} École des Mines de Saint-Étienne
\textsuperscript{2} LISA - Institut de Mathématiques Appliquées d’Angers
\textsuperscript{3} Institut Polytechnique de Grenoble

April 26-28, 2010
# Table of contents

1. **Introduction**
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. **0-1 Branch and Bound**
   - Outline
   - Initial cuts
   - Characteristics

3. **Conclusion**
   - Experimental results
   - Concluding remarks
Plan

1. Introduction
   - Problem
     - Time-indexed ILP formulation
     - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
Nature of the problem (1/3)

Get a feasible production plan

JOB-SHOP

EMPLOYEE TIMETABLING

Minimize labor costs
Nature of the problem (2/3) - Employee Timetabling problem

Time horizon

- Timetabling horizon $H = \delta \cdot \pi$ where:
  - $\delta$ number of shifts
  - $\pi$ duration time of a shift

Nota: it can modelize, for example, a three-shift system

Employee Timetabling Problem for a set $E$ of $\mu$ employees

- $A_e$ set of machines employee $e$ masters
- $T_e$ set of shifts where employee $e$ is available
Job-Shop: Schedule a set $J$ of $n$ jobs on $m$ machines

- $\forall j \in J \{ O_{ji} \}_{i=1}^m$ chain of operations of job $j$
  - machine $m_{ji} \in \{1 \ldots m\}$
  - processing time $p_{ji}$

  $\leftarrow$ Notation $\rho_{jk}$: processing time of $j$ on machine $k$

- can not be interrupted
- requires a qualified employee to use machine $m_{ji}$

Feasible production plan

A schedule for which all operations are completed before a given scheduling completion time $C_{\text{max}} \leq H$. 
Nature of the problem (3/3)

Objective

Assigning at **minimum cost** employees to both machines and shifts in order to be able to provide a *feasible production plan*
Example (6 jobs - 4 machines - 15 employees)
Motivation: Extension of work

Continuation of a work

- Integrated employee timetabling and production scheduling problem

**Methods:**
- Decomposition and cut generation process
- Simplified production scheduling problem

**Motivation for this new study:**
- Is the decomposition and cut generation process also efficient with a harder production scheduling problem (Job-Shop)?
Motivation: Case treated in literature

Case treated in literature


- **Aim**: To experiment hybrid CP-ILP methods on an integrated job-shop scheduling and employee timetabling problem

- **Methods**: 
  - CP with a global additional constraint corresponding to the LP-relaxation of the employee timetabling problem

- **Our study**: specific case of mapping activities - machines
  - 8 of the 11 instances of [AGRV09] can be used
Plan

1. Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
ILP model (1/4)

Binary decision variables

- \( x_{eks} = 1 \) if employee \( e \) is assigned to the pair (machine \( k \); shift \( s \))
- \( y_{ikt} = 1 \) if job \( i \) starts its processing on the machine \( k \) at instant \( t \)
Objective function

\[ \text{min } \Theta = \sum_{e \in E} \sum_{k \in A_e} \sum_{s \in T_e} c_{eks} \cdot x_{eks} \]

Employee Timetabling Problem specific constraints

\[
\begin{align*}
\sum_{k \notin A_e} \sum_{s=0}^{\sigma} x_{eks} &= 0 \\
\sum_{k \in A_e} \sum_{s \notin T_e} x_{eks} &= 0 \\
\sum_{k \in A_e} (x_{eks} + x_{ek(s+1)} + x_{ek(s+2)}) &\leq 1 \\
x_{eks} &\in \{0, 1\}
\end{align*}
\]

\[
\begin{align*}
\text{e} &= 1, \ldots, \mu \\
\text{e} &= 1, \ldots, \mu \\
\text{e} &= 1, \ldots, \mu \\
\text{s} &= 0, \ldots, \sigma - 3 \\
\text{e} &= 1, \ldots, \mu \\
\text{k} &= 1, \ldots, m \\
\text{s} &= 0, \ldots, \sigma - 1
\end{align*}
\]
ILP model (3/4)

Problem

Time-indexed ILP formulation

Solution methods

Job-Shop scheduling problem specific constraints

\[
\begin{align*}
\sum_{t=0}^{d_{ik} - \rho_{ik}} t \cdot y_{ikt} + \rho_{ik} & \leq C_{\max} \\
\sum_{t=r_{ik}}^{d_{ik} - \rho_{ik}} y_{ikt} & = 1 \\
\sum_{t=0}^{r_{ik}} y_{ikt} + \sum_{t=d_{ik} - \rho_{ik} + 1}^{C_{\max}} y_{ikt} & = 0 \\
\sum_{u=r_{ik} + \rho_{ik}}^{t} y_{ilu} - \sum_{u=r_{ik}}^{t - \rho_{ik}} y_{iku} & \leq 0 \\
\sum_{i=1}^{n} \sum_{u=\max(r_{ik}, t - \rho_{ik} + 1)}^{\min(d_{ik} - \rho_{ik}, t)} y_{iku} & \leq 1 \\
y_{ikt} & \in \{0, 1\}
\end{align*}
\]

\[i = 1, \ldots, n \quad k = m_{im} \]

\[i = 1, \ldots, n \quad k = 1, \ldots, m \]

\[i = 1, \ldots, n \quad k = 1, \ldots, m \]

\[i = 1, \ldots, n \quad j = 1, \ldots, m - 1 \]

\[k = m_{ij} \quad l = m_{i(j+1)} \]

\[t = \rho_{ik} + \rho_{ik}, \ldots, d_{il} - \rho_{il} \]

\[k = 1, \ldots, m \quad t = 0, \ldots, C_{\max} \]

\[i = 1, \ldots, n \quad k = 1, \ldots, m \]

\[t = 0, \ldots, C_{\max} \]
Coupling constraints

\[ \sum_{e \in E} x_{eks} - \sum_{i=1}^{n} \sum_{u=\max(r_{ik},t-\rho_{ik}+1)}^{\min(d_{ik}-\rho_{ik},t)} y_{iku} \geq 0 \quad k = 1, \ldots, m \quad t = 0, \ldots, C_{\text{max}} \]

\[ s = \lfloor t/\pi \rfloor \]
Plan

1 Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3 Conclusion
   - Experimental results
   - Concluding remarks
Solution methods

Three exact methods

- MIP
- Decomposition and cut generation process
- 0-1 Branch and Bound based on the work (or not) for each pair (machine, shift)
Plan

1. Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
Underlying ideas

- Splitting of $[P]$ (by relaxing coupling constraints)
  - $[Job \ – \ Shop]$ (non-coupling constraints; with variables $y_{jkt}$)
  - $[Employee]$ (non-coupling constraints; with variables $x_{eks}$)

- Fixing an assignment $\bar{z}$ of worked and unworked pairs (machine, shift)

- Checking the feasibility of $\bar{z}$ at two levels:
  - $[Job \ – \ Shop]$ which is solved with a dedicated Job-Shop solver
  - $[Employee]$ which is solved with an ILP solver

- To avoid an exhaustive search:
  $\leftrightarrow \ \bar{z}$ is generated through a 0-1 Branch and Bound coupled with a generation of
  - 3 sets of initial cuts
  - feasibility cuts (generated all along the process)
Plan

1 Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3 Conclusion
   - Experimental results
   - Concluding remarks
Probing initial cuts

What?
Test the obligation of work for each pair (machine $k$, shift $s$)

How?
- create a fictive job $j_f$ that has to be processed on $k$ over $s$ for a duration of $\pi$
- solve [Job-Shop] with a dedicated Job-Shop solver
- if [Job-Shop] is unfeasible ($\iff$ does not respect $C_{max}$):
  - $j_f$ ($\iff$ absence of work on $(k,s)$) can not be scheduled
  - an employee must be assigned to $(k,s)$

Result
- set of pairs $(k,s)$ fixed to work
- cuts for [Employee] $\Rightarrow \sum_{e=1}^{\mu} x_{eks} = 1 \ \forall \ (k,s)$ fixed to work
Capacitated cuts

Capacitated cut for machine $k$

Find a close lower bound $b_k$ of the minimal number of worked shifts on $k$

How? $b_k$ is the maximum of 3 valid lower bounds

1. **direct**: number of pairs $(k, s)$ fixed by probing
2. **direct**: $\left\lceil \frac{\sum_{j=1}^{n} \rho_{jk}}{\pi} \right\rceil$
3. **Fix** (by probing) the maximum number of schedulable fictive jobs

Result: Cuts for $[\text{Employee}]$

$$\sum_{e=1}^{\mu} \sum_{s \in T_e} x_{eks} \geq b_k \quad k = 1, \ldots, m$$
Non-overlapping cuts for each pair (machine \( k \), shift \( s \))

To ensure each feasible solution to get at most one employee assigned to \((k, s)\)

Result: Cuts for \([\text{Employee}]\)

\[
\sum_{e \in E | (k \in A_e) \land (s \in T_e)} x_{eks} \leq 1 \quad k = 1, \ldots, m \quad s = 0, \ldots, \delta - 1
\]
Plan

1. Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
Characteristics (1/4)

Binary separation scheme
- left child: impose a given pair (machine $\bar{k}$, shift $\bar{s}$) to be not worked
- right child: impose the same given pair ($\bar{k}$, $\bar{s}$) to be worked

Exploration strategy
depth-first

Branching pair (machine $\bar{k}$, shift $\bar{s}$)
- $\bar{k}$: machine with a maximum gap between the number of shifts fixed (or which can still be fixed) to work and the lower bound $LB_k$ of the minimum number of worked shifts on $\bar{k}$
- $\bar{s}$: latest shift such that ($\bar{k}$, $\bar{s}$) has not been fixed yet
Definition

- **Relaxed assignment**
  Assignment of pairs \((k, s)\) such that each fixed pair \((k, s)\) is fixed to its value and all the other ones are free

- **Strict assignment**
  Assignment of pairs \((k, s)\) such that each fixed pair \((k, s)\) is fixed to its value and all the other ones are imposed to be unworked

Evaluation

LP-relaxation of [Employee] for the relaxed assignment
Implications rules

If the decision is: *the pair* \((\tilde{k}, \tilde{s})\) *must be unworked*

We try to fix pairs (machine, shift) by:

- using probing techniques for each non-fixed pair \((\tilde{k}, s)\)
- checking the respect of the lower bound \(b_{\tilde{k}}\) of the minimal number of worked shifts on \(\tilde{k}\)

Elimination rules

A branching node can be pruned if one of these two conditions is fulfilled:

1. *Employee* with *relaxed assignment* is unfeasible (evaluation) or has a cost greater than the best current known
2. *Relaxed assignment* is unfeasible for *Job – Shop*
If the current node has not been pruned

- We get the optimum \((\bar{x}^*, \bar{\theta}^*)\) for \([Employee]\) that respects the strict assignment
  - if \(\bar{x}^*\) exists, we check if the strict assignment is feasible for \([Job − Shop]\)
    - if it is, \(UB\) is updated: \(UB \leftarrow \bar{\theta}^*\)
  - If \(UB\) is not updated: we add a feasibility cut to \([Employee]\)

Feasibility cut for \([Employee]\)

- permits to eliminate solutions similar to \(\bar{x}^*\)
- impose the work on at least one of the unworked pairs \((k, s)\) of the strict assignment
- \[ \text{Cut} \colon \sum_{(k,s)\text{not fixed}} \sum_{e=1}^{\mu} x_{eks} \geq 1 \]
Plan

1. Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
## Test bed

### 8 instances of [AGRV09]

<table>
<thead>
<tr>
<th>Instance</th>
<th>$n$</th>
<th>$m$</th>
<th>$\mu$</th>
<th>$\mu_{\text{extra}}$</th>
<th>$\delta$</th>
<th>$\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ejs$</td>
<td>6</td>
<td>4</td>
<td>25</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$ejs8X8$</td>
<td>8</td>
<td>8</td>
<td>40</td>
<td>20</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$ejs10X10$</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

### Tools

- Computer programming language: Java except for [AGRV09] (C++)
- MIP solver: Ilog Cplex 12.1
- CP solvers (for [AGRV09]): Ilog Solver 6.7; Ilog Scheduler 6.7
- Job-Shop solver: Branch & Bound (*Carlier, Péridy, Pinson, Rivreau*)
- Processor: Intel Core 2 Quad Q6600 @ 2,40 GHz - 3 GB RAM
- CPU time limit: 300 seconds
## Results: initial cuts

### Impact of the initial cuts

<table>
<thead>
<tr>
<th>Instance</th>
<th>LP relaxation of $[P]$</th>
<th>LP′ relaxation of $[P]$ with initial cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$LP/\Theta^*$</td>
<td>Time</td>
</tr>
<tr>
<td>ejs4</td>
<td>92.4%</td>
<td>0.2s</td>
</tr>
<tr>
<td>ejs9</td>
<td>87.3%</td>
<td>0.3s</td>
</tr>
<tr>
<td>ejs10</td>
<td>97.1%</td>
<td>0.5s</td>
</tr>
<tr>
<td>ejs8 × 8₁</td>
<td>60.1%</td>
<td>1.6s</td>
</tr>
<tr>
<td>ejs8 × 8₂</td>
<td>68.0%</td>
<td>2.5s</td>
</tr>
<tr>
<td>ejs8 × 8₃</td>
<td>55.1%</td>
<td>1.6s</td>
</tr>
<tr>
<td>ejs10 × 10₁</td>
<td>55.7%</td>
<td>11.4s</td>
</tr>
<tr>
<td>ejs10 × 10₃</td>
<td>69.1%</td>
<td>18.1s</td>
</tr>
</tbody>
</table>
# Results: exact methods

## Results of the exact methods

<table>
<thead>
<tr>
<th>Instance</th>
<th>$\Theta^*$</th>
<th>$\Theta$ Time</th>
<th>$\Theta$ Time</th>
<th>$\Theta$</th>
<th>Preprocessing time</th>
<th>Total time</th>
<th>#initial cuts + #cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ejs4</td>
<td>23</td>
<td>23 0.2s</td>
<td>23 0.9s</td>
<td>23</td>
<td>0.0s</td>
<td>0.0s</td>
<td>43 + 0</td>
</tr>
<tr>
<td>ejs9</td>
<td>24</td>
<td>24 11.4s</td>
<td>24 87.9s</td>
<td>24</td>
<td>1.5s</td>
<td>5.0s</td>
<td>67 + 44</td>
</tr>
<tr>
<td>ejs10</td>
<td>23</td>
<td>23 1.0s</td>
<td>23 3.5s</td>
<td>23</td>
<td>0.4s</td>
<td>0.6s</td>
<td>50 + 2</td>
</tr>
<tr>
<td>ejs8 × 8</td>
<td>78</td>
<td>84 TL</td>
<td>78 64.8s</td>
<td>78</td>
<td>9.5s</td>
<td>37.8s</td>
<td>94 + 66</td>
</tr>
<tr>
<td>ejs8 × 8</td>
<td>96</td>
<td>96 TL</td>
<td>96 98.9s</td>
<td>96</td>
<td>7.9s</td>
<td>43.3s</td>
<td>103 + 108</td>
</tr>
<tr>
<td>ejs8 × 8</td>
<td>83</td>
<td>83 TL</td>
<td>83 29.7s</td>
<td>83</td>
<td>4.5s</td>
<td>9.0s</td>
<td>93 + 15</td>
</tr>
<tr>
<td>ejs10 × 10</td>
<td>124</td>
<td>-1 TL</td>
<td>137 TL</td>
<td>124</td>
<td>17.5s</td>
<td>33.0s</td>
<td>158 + 31</td>
</tr>
<tr>
<td>ejs10 × 10</td>
<td>95</td>
<td>-1 TL</td>
<td>150 TL</td>
<td>102</td>
<td>218.8s</td>
<td>TL</td>
<td>164 + 251</td>
</tr>
</tbody>
</table>
1. Introduction
   - Problem
   - Time-indexed ILP formulation
   - Solution methods

2. 0-1 Branch and Bound
   - Outline
   - Initial cuts
   - Characteristics

3. Conclusion
   - Experimental results
   - Concluding remarks
Concluding remarks

- Strong initial cuts (especially probing cuts)
- Interesting decomposition approach
- 0-1 Branch and Bound really competitive