# Algorithms for the flexible cyclic job-shop problem

F. Quinton, I. Hamaz, L. Houssin

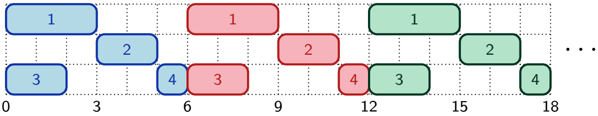
LAAS-CNRS, Toulouse, France

September 7, 2021

## Cyclic scheduling

### Scheduling repetitive tasks

What people[1] think it is

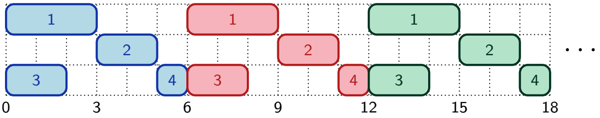



1 Of course not PMS audience.

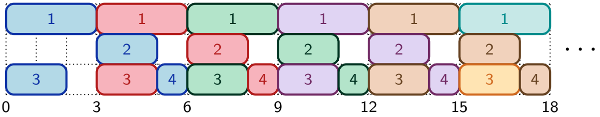
## Cyclic scheduling

### Scheduling repetitive tasks

What people[1] think it is




What it really is




[1] Of course not PMS audience.

# The cyclic job shop problem

Schedule elementary tasks $T = \{1, ..., n\}$ **infinitely repeated**.

- Each task $i$ has a processing time $p_i$ and belongs to a subset $T_J \subset T$ called job.

# The cyclic job shop problem

Schedule elementary tasks $T = \{1, ..., n\}$ **infinitely repeated**.

- Each task $i$ has a processing time $p_i$ and belongs to a subset $T_J \subset T$ called job.
- $t(i, k)$, $k \in \mathbb{N}$, is the start time of the $k-th$ occurrence of $i$

$$\forall i \in T, \forall k \in \mathbb{N} : \quad t(i, k+1) = \alpha + t(i, k)$$

where $\alpha$ is the cycle time.

# The cyclic job shop problem

Schedule elementary tasks $T = \{1, ..., n\}$ **infinitely repeated**.

- Each task $i$ has a processing time $p_i$ and belongs to a subset $T_J \subset T$ called job.
- $t(i, k)$, $k \in \mathbb{N}$, is the start time of the $k-th$ occurrence of $i$

$$\forall i \in T, \forall k \in \mathbb{N} : \quad t(i, k+1) = \alpha + t(i, k)$$

  where $\alpha$ is the cycle time.
- Each elementary task $i$ is executed using a machine $M(i) \in M = \{1, ..., m\}$, where $m < n$

# The cyclic job shop problem

Schedule elementary tasks $T = \{1, ..., n\}$ **infinitely repeated**.

- Each task $i$ has a processing time $p_i$ and belongs to a subset $T_J \subset T$ called job.
- $t(i,k)$, $k \in \mathbb{N}$, is the start time of the $k-th$ occurrence of $i$

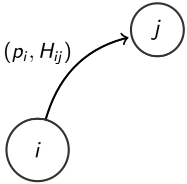
$$\forall i \in T, \forall k \in \mathbb{N} : \quad t(i, k+1) = \alpha + t(i,k)$$

  where $\alpha$ is the cycle time.
- Each elementary task $i$ is executed using a machine $M(i) \in M = \{1, ..., m\}$, where $m < n$
- Elementary tasks are connected by **uniform** constraints and **disjunctive** constraints.

# Uniform constraints

- Precedence constraint between tasks $i$ and $j$:

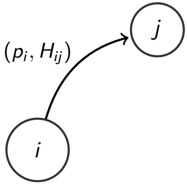
$$\forall i, j \in T, \forall k \in \mathbb{N} : \quad t(i,k) + p_i \leq t(j, k + H_{ij}).$$



# Uniform constraints

- Precedence constraint between tasks $i$ and $j$:

$$\forall i, j \in T, \forall k \in \mathbb{N} : \quad t(i,k) + p_i \leq t(j, k + H_{ij}).$$



## Disjunctive constraints

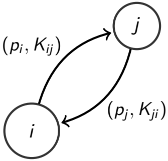
Resource constraints leads to
$\forall i, j \in T,\ \forall k \in \mathbb{N}$ such that $M(i) = M(j),\ i \neq j$ :

$$t(i,k) + p_i \leq t(j, k + K_{ij})$$
$$t(j,k) + p_j \leq t(i, k + K_{ji})$$

### Occurrence shift property

For all couple of elementary tasks $(i, j) \in T^2$ such that $M(i) = M(j)$ we know that

$$K_{ij} + K_{ji} = 1$$



# Solving a cyclic job-shop scheduling problem

**Goal**

Minimize the cycle time $\alpha$.

- find a cyclic schedule $w$ where all $t(i, k)$ minimize $\alpha$.

# Solving a cyclic job-shop scheduling problem

**Goal**

Minimize the cycle time $\alpha$.

- find a cyclic schedule $w$ where all $t(i,k)$ minimize $\alpha$.
- A cyclic schedule $w$ is totally defined by
  - a set $S_w = \{t(i,0) \in \mathbb{R}^+ \mid i \in T\}$
  - a cycle time $\alpha$

# CJSP Model 1

$$\min \alpha$$

s.t.

$$\alpha \geq p_i, \quad \forall i \in \mathcal{T} \tag{1}$$

$$t_j + \alpha H_{i,j} \geq t_i + p_i, \quad \forall (i,j) \in \mathcal{E} \tag{2}$$

$$t_j + \alpha \times K_{ij} \geq t_i + p_i, \quad \forall (i,j) \in \mathcal{D} \tag{3}$$

$$K_{ij} + K_{ji} = 1, \quad \forall (i,j) \in \mathcal{D} \tag{4}$$

$$K_{ij} \in \mathbb{Z}, \quad \forall (i,j) \in \mathcal{D} \tag{5}$$

$$t_i \geq 0, \quad \forall i \in \mathcal{T}. \tag{6}$$

# CJSP Model 1

$$\min \alpha$$

s.t.

$$\alpha \geq p_i, \quad \forall i \in \mathcal{T} \tag{1}$$

$$t_j + \alpha H_{i,j} \geq t_i + p_i, \quad \forall (i,j) \in \mathcal{E} \tag{2}$$

$$t_j + \alpha \times K_{ij} \geq t_i + p_i, \quad \forall (i,j) \in \mathcal{D} \tag{3}$$

$$K_{ij} + K_{ji} = 1, \quad \forall (i,j) \in \mathcal{D} \tag{4}$$

$$K_{ij} \in \mathbb{Z}, \quad \forall (i,j) \in \mathcal{D} \tag{5}$$

$$t_i \geq 0, \quad \forall i \in \mathcal{T}. \tag{6}$$

# CJSP Model 2

Define the variable $\tau = \frac{1}{\alpha}$ and for all $i \in T$, the variables $u_i = \frac{t_i}{\alpha}$.

# CJSP Model 2

$$\max \tau$$

s.t.

$$\tau \leq \frac{1}{p_i}, \quad \forall i \in \mathcal{T} \tag{7}$$

$$u_j + H_{i,j} \geq u_i + \tau p_i, \quad \forall (i,j) \in \mathcal{E} \tag{8}$$

$$u_j + K_{ij} \geq u_i + \tau p_i, \quad \forall (i,j) \in \mathcal{D} \tag{9}$$

$$K_{ij} + K_{ji} = 1, \quad \forall (i,j) \in \mathcal{D} \tag{10}$$

$$K_{ij} \in \mathbb{Z}, \quad \forall (i,j) \in \mathcal{D} \tag{11}$$

$$u_i \geq 0, \quad \forall i \in \mathcal{T}. \tag{12}$$

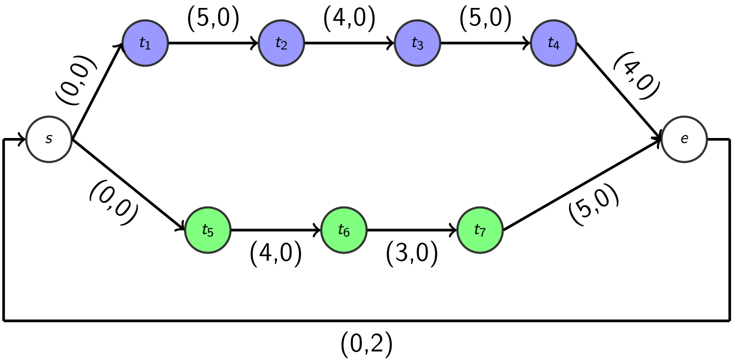
# Example

## CJSP Example

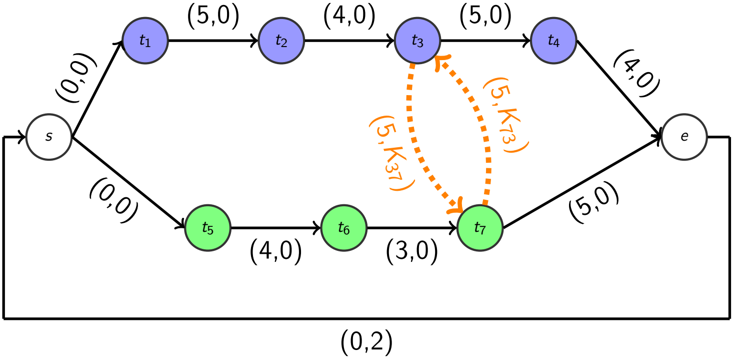
- Data:
  - 7 tasks
  - 4 machines

| Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Time | 5 | 4 | 5 | 4 | 4 | 3 | 5 |
| Machine | $M_1$ | $M_1$ | $M_3$ | $M_1$ | $M_2$ | $M_4$ | $M_3$ |

# Associated graph of the CJSP



# Associated graph of the CJSP



# Example - cyclic schedule with $WIP = 2$

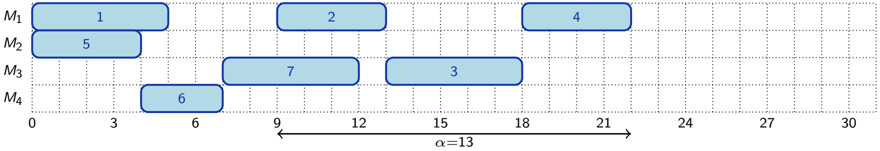



Figure: Periodic schedule example.

# Example - cyclic schedule with $WIP = 2$

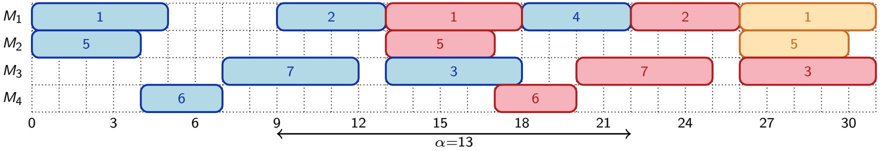



Figure: Periodic schedule example.

# Flexible cyclic job shop scheduling problem

The Flexible Cyclic Job shop Scheduling Problem (FCJSP) is a CJSP where the elementary tasks are **flexible**.

## Flexibility

For all elementary task $i \in T$, define the set $R(i) \subset M$ of machines on which $i$ can be assigned.

- The assignment of a task $i$ to a machine $r \in R(i)$ is a decision variable.

# Flexible cyclic job shop scheduling problem

The Flexible Cyclic Job shop Scheduling Problem (FCJSP) is a CJSP where the elementary tasks are **flexible**.

**Flexibility**

For all elementary task $i \in T$, define the set $R(i) \subset M$ of machines on which $i$ can be assigned.

- The assignment of a task $i$ to a machine $r \in R(i)$ is a decision variable.
- A flexible cyclic schedule $w$ is totally defined by
  - a set $S_w = \{t(i,0) \in \mathbb{R}^+ \mid i \in T\}$
  - a set $R_w = \{m(i) \in R(i) \mid i \in T\}$
  - a cycle time $\alpha$

# Example

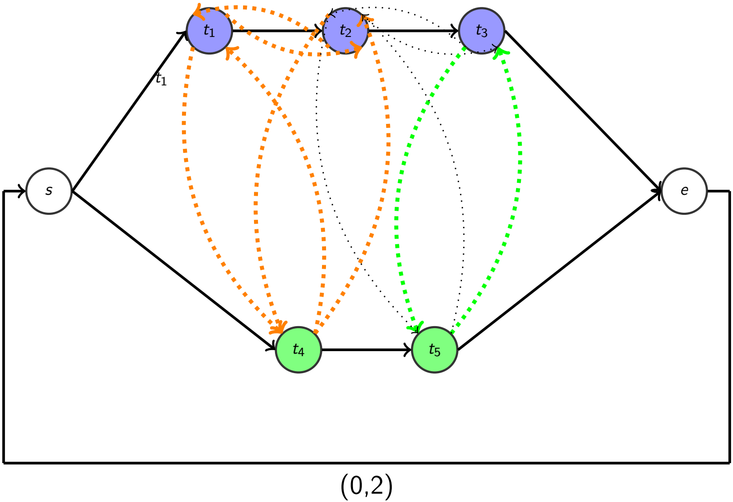
## FCJSP Example

- Data:
  - 5 tasks
  - 2 machines
  - Task 2 can be assigned to machines $M_1$ or $M_2$.

| Task | 1 | 2 | | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Machine | $M_1$ | $M_1$ | $M_2$ | $M_2$ | $M_1$ | $M_2$ |

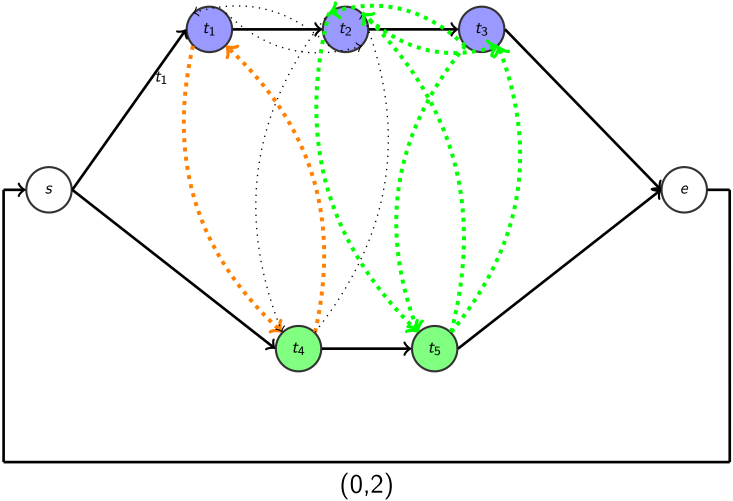
# Effect of machine assignment on the set of constraints

With task 2 assigned to machine 1:



# Effect of machine assignment on the set of constraints

With task 2 assigned to machine 2:



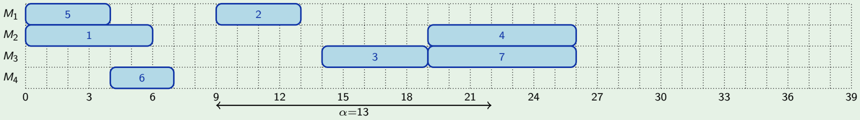
# Example

- Data:
  - 7 tasks
  - 4 machines
  - Each task can be assigned to one of three machines.

| Task | 1 | | | 2 | | | 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Machine | $M_1$ | $M_2$ | $M_4$ | $M_1$ | $M_2$ | $M_3$ | $M_2$ | $M_3$ | $M_4$ |
| Time | 5 | 6 | 6 | 4 | 5 | 6 | 7 | 5 | 6 |
| **Task** | **4** | | | **5** | | | **6** | | |
| Machine | $M_1$ | $M_2$ | $M_3$ | $M_1$ | $M_2$ | $M_3$ | $M_1$ | $M_3$ | $M_4$ |
| Time | 4 | 7 | 5 | 4 | 4 | 5 | 3 | 4 | 3 |
| **Task** | **7** | | | | | | | | |
| Machine | $M_1$ | $M_2$ | $M_3$ | | | | | | |
| Time | 5 | 7 | 5 | | | | | | |

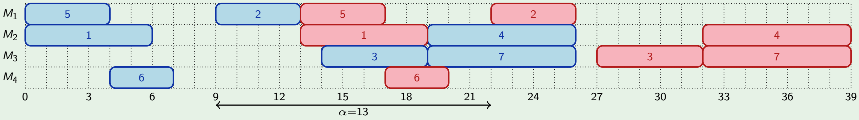
# Example - flexible cyclic schedule

## Example



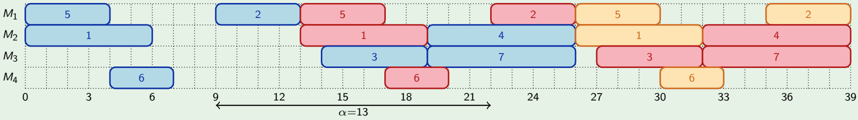
# Example - flexible cyclic schedule

## Example



# Example - flexible cyclic schedule

## Example



# Model for the FCJSP

Let us define

- $\forall i \in T, \forall r \in R(i), m_{i,r} = \begin{cases} 1 \text{ if task i is assigned to machine r} \\ 0 \text{ otherwise} \end{cases}$
- $\forall i, j \in T, \quad R(i,j) = R(i) \cap R(j)$ the set of common machines for tasks $i$ and $j$.
- $E = \{(i,j) \in T^2 | i \text{ precedes } j\}$ the set of precedence constraints.
- $D = \{(i,j) | R(i,j) \neq \emptyset\}$ the set of disjunction constraints.

# Non-linear intuitive model for the FCJSP

The FCJSP can be modelized as follows:

$$\min \ \alpha$$

s.t.

$$\begin{aligned}
\alpha \geq p_{i,r} - (1 - m_{i,r}) \times M_1, &\quad \forall i = 1, ..., n; \forall r \in M(i) \\
t_j + \alpha \times H_{i,j} \geq t_i + p_{i,r} \times m_{i,r}, &\quad \forall (i,j) \in E; \forall r \in M(i) \\
t_j + \alpha \times K_{ij} \geq t_i + p_{i,r} \times m_{i,r}, &\quad \forall (i,j) \in D; \forall r \in M(i) \\
\sum_{r \in M(i)} m_{i,r} = 1, &\quad \forall i = 1, ..., n
\end{aligned}$$

$$t_i \geq 0, \alpha \geq 0 \quad \forall i = 1, ..., n$$

$$K_{i,j} + K_{j,i} = 1, \quad \forall (i,j) \in D$$

$$K_{ij} \in \mathbb{Z}$$

$$m_{i,r} \in \{0, 1\}, \quad \forall i = 1, ..., n$$

# Non-linear intuitive model for the FCJSP

The FCJSP can be modelized as follows:

$$\min \ \alpha$$

s.t.

$$\alpha \geq p_{i,r} - (1 - m_{i,r}) \times M_1, \quad \forall i = 1, ..., n; \forall r \in M(i)$$
$$t_j + \alpha \times H_{i,j} \geq t_i + p_{i,r} \times m_{i,r}, \quad \forall (i,j) \in E; \forall r \in M(i)$$
$$t_j + \alpha \times K_{ij} \geq t_i + p_{i,r} \times m_{i,r}, \quad \forall (i,j) \in D; \forall r \in M(i)$$
$$\sum_{r \in M(i)} m_{i,r} = 1, \quad \forall i = 1, ..., n$$
$$t_i \geq 0, \alpha \geq 0 \quad \forall i = 1, ..., n$$
$$K_{i,j} + K_{j,i} = 1, \quad \forall (i,j) \in D$$
$$K_{ij} \in \mathbb{Z}$$
$$m_{i,r} \in \{0, 1\}, \quad \forall i = 1, ..., n$$

# Linearization

For linearization purpose we introduce:

- $\tau = \frac{1}{\alpha}$
- $\forall i \in T, u_i = \frac{t_i}{\alpha}$

# MIP model the the FCJSP

$$\max \tau$$

$$\tau \leq \sum_{r \in R(i)} \frac{m_{i,r}}{p_{i,r}},, \quad \forall i \in T, \forall r \in R(i)$$

$$u_j + H_{i,j} \geq u_i + \sum_{r \in R(i)} p_{i,r} y_{i,r}, \quad \forall (i,j) \in E, \forall r \in R(i)$$

$$P_1(2 - m_{i,r} - m_{j,r}) + u_j + K_{i,j} \geq u_i + p_{i,r} y_{i,r}, \quad \forall (i,j) \in E, \forall r \in R(i,j)$$

$$\sum_{r \in R(i)} y_{i,r} = \tau, \quad \forall i \in T$$

$$y_{i,r} \leq m_{i,r}, \quad \forall i \in \mathcal{T}, \forall r \in R(i)$$

$$\sum_{r \in R(i)} m_{i,r} = 1, \quad \forall i \in T$$

$$K_{i,j} + K_{j,i} = 1, \quad \forall (i,j) \in D$$

# MIP model the the FCJSP

$$K_{ij} \in \mathbb{Z}, \quad \forall (i,j) \in D$$
$$u_i \geq 0, \quad \forall i \in T$$
$$m_{i,r} \in \{0,1\}, \quad \forall i \in T \forall r \in R(i)$$
$$y_{i,r} \geq 0, \quad \forall i \in T \forall r \in R(i)$$

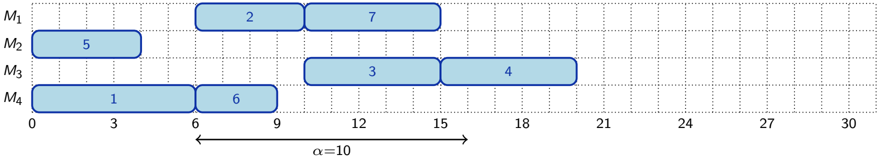



Figure: FCJSP example solved by the MIP.

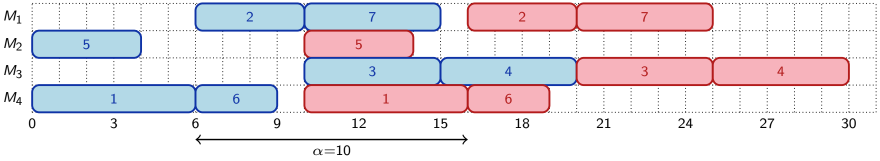



Figure: FCJSP example solved by the MIP.

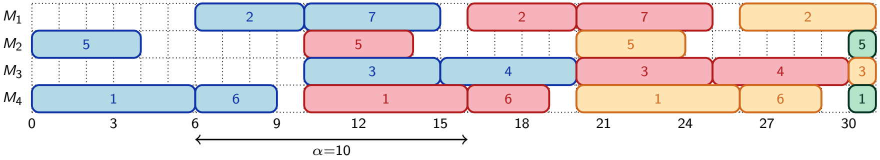



Figure: FCJSP example solved by the MIP.

# Numerical Results

Table: Numerical results with the MIP model

| Instance parameters | | Solving time (sec) | Gap to optima | Number of solved instances |
|---|---|---|---|---|
| 15 tasks | 3 machines | 25.09 | - | 10 |
| 15 tasks | 4 machines | 36.58 | 11.11% | 9 |
| 15 tasks | 5 machines | 5.13 | 16.67% | 9 |
| 20 tasks | 3 machines | 111.87 | 17.29% | 8 |
| 20 tasks | 4 machines | 115.25 | 13.06% | 8 |
| 20 tasks | 5 machines | time out | 23.81% | 0 |
| 30 tasks | 3 machines | time out | 32.64% | 0 |
| 30 tasks | 4 machines | time out | 123.61% | 0 |
| 30 tasks | 5 machines | time out | 121.95% | 0 |

# Benders decomposition

## First approach

- Corresponding Master Problem (MP) is composed of the constraints involving only the integer variables and the optimality cuts and the feasibility cuts.
- Subproblem involves only continuous variables. Can be written as a LP.

# Benders decomposition

## First approach: Master Problem

$$\max \ z$$

s.t.

$$\sum_{r \in r(i)} m_{i,r} = 1, \quad \forall i \in \mathcal{T} \tag{13a}$$

$$K_{ij} + K_{ji} = 1, \quad \forall (i,j) \in \mathcal{D} \tag{13b}$$

$$\text{Feasibility Cuts } \mathcal{F} \tag{13c}$$

$$\text{Optimality Cuts } \mathcal{O} \tag{13d}$$

$$m_{i,r} \in \{0,1\}, \quad \forall i \in \mathcal{T}, \forall r \in R(i) \tag{13e}$$

$$z \in \mathbb{R}, \quad K_{ij} \in \mathbb{Z}, \quad \forall (i,j) \in \mathcal{D}. \tag{13f}$$

# Benders decomposition

## First approach: SubProblem

$$\max \ \tau$$

s.t.

$$\tau \leq \sum_{r \in R(i)} \frac{\overline{m}_{i,r}}{p_{i,r}}, \quad \forall i \in \mathcal{T} \tag{14}$$

$$u_j + H_{ij} \geq u_i + \sum_{r \in R(i)} y_{i,r} \ p_{i,r}, \quad \forall (i,j) \in \mathcal{E} \tag{15}$$

$$P_1(2 - \overline{m}_{i,r} - \overline{m}_{j,r}) + u_j + \overline{K}_{i,j} \geq u_i + y_{i,r} p_{i,r}, \quad \forall (i,j) \in \mathcal{D}, \forall r \in R(i,j) \tag{16}$$

$$\sum_{r \in R(i)} y_{i,r} = \tau, \quad \forall i \in \mathcal{T} \tag{17}$$

$$y_{i,r} \leq \overline{m}_{i,r}, \quad \forall i \in \mathcal{T}, \forall r \in R(i) \tag{18}$$

$$\tau \geq 0, \quad u_i \geq 0, \forall i \in \mathcal{T}, \quad y_{i,r} \geq 0, \forall i \in \mathcal{T}, \forall r \in R(i). \tag{19}$$

# Benders decomposition

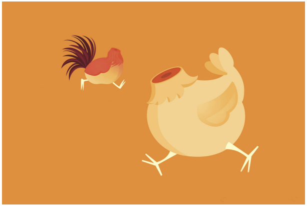
## First approach: Results

- very slow
- produces tons of infeasibility cuts

# Benders decomposition

## First approach: Why is it so bad?

- Not enough connection between the MP and the SP.
- The master runs around like a headless chicken.



▸ more

# Benders decomposition

## Second approach

- Add cuts in the MP to ensure the existence of feasible starting times fulfilling the disjunction constraints and precedence constraints implied by the integer variables.
- Even a naive UB of the cycle time is sufficient to ensure feasibility of the SP.

# Numerical experimentations

Time to optimality

| Instance name | MILP | Benders decomposition | 2-Flex heuristic |
|---|---|---|---|
| inst0_10tasks_4machines | 29.37 | 505.57 | 1.78(10.49%) |
| inst1_10tasks_4machines | 461.93 | 668.82 | 0.63(-) |
| inst2_10tasks_4machines | 52.68 | 356.89 | 0.91(8.25%) |
| inst3_10tasks_4machines | 132.53 | 703.93 | 1.04(2.53%) |
| inst4_10tasks_4machines | 93.66 | 458.89 | 1.01(1.40%) |
| inst5_10tasks_4machines | 19.78 | 364.25 | 0.43(5.44%) |
| inst6_10tasks_4machines | 9.15 | 514.52 | 0.66(9.57%) |
| inst7_10tasks_4machines | 49.81 | 455.74 | 0.31(2.97%) |
| inst8_10tasks_4machines | 124.83 | 659.4 | 0.86(2.46%) |
| inst9_10tasks_4machines | 14.65 | 317.62 | 0.87(2.97%) |

# Numerical experimentations

Time to optimality

| Instance name | MILP | Benders decomposition | 2-Flex heuristic |
|---|---|---|---|
| inst0_10tasks_3machines | timeout | 819.95 | 66.29(-) |
| inst1_10tasks_3machines | 946.08 | 614.28 | 2.93(-) |
| inst2_10tasks_3machines | 1640.75 | 516.42 | 6.74(-) |
| inst3_10tasks_3machines | 823.53 | 589.63 | 27.07(-) |
| inst4_10tasks_3machines | 658.47 | 385.83 | 34.43(2.58%) |
| inst5_10tasks_3machines | 765.6 | 424.89 | 21.01(2.58%) |
| inst6_10tasks_3machines | timeout | 954.92 | 7.62(-) |
| inst7_10tasks_3machines | 2383.02 | 945.99 | 34.74(-) |
| inst8_10tasks_3machines | 3215.08 | 769.91 | 9.81(-) |
| inst9_10tasks_3machines | 2337.64 | 473.86 | 13.17(-) |

# Conclusion and perspectives

Conclusion

- An original model for the flexible CJSP.
- A Benders decomposition.
- We propose a heuristic procedure and perform some numerical experimentations. heuristic
- Full paper in Annals of OR 2020

# Heuristics for the FCJSP

The FCJSP is highly combinatory. For large instances, MIP does not give satisfying results. To tacle this issue, we propose two heuristic procedures:

- For every $i \in T$, consider a wisely chosen subset of $R(i)$.
- Solve the problem where only a subset of elementary tasks $i \in T$ are flexible.

# Reduce the $R(i)$ sets

Define $R^H(i) \subset R(i) \forall i \in T$ the reduced subset of machines to which task $i$ can be assigned.

- $R^H(i)$ can be defined has the set of the $k$ machine that produce task $i$ the fastest, with $k < card(R(i))$ .
- Solving with a reduced number of machines for every tasks drastically reduces the combinatory of the problem.

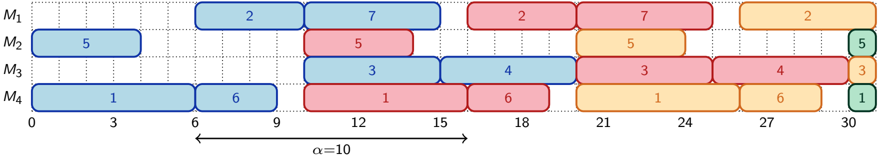



Figure: FCJSP example solved by the MIP.

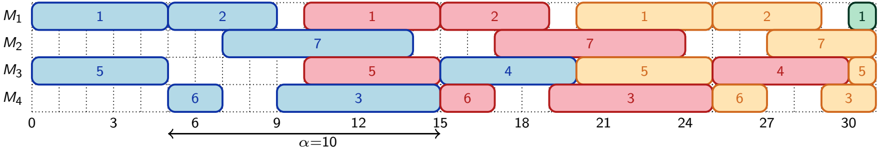



Figure: FCJSP example solved by the Benders algorithm.

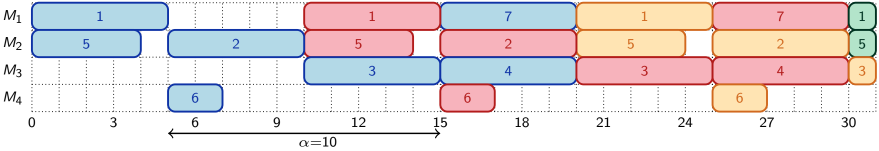
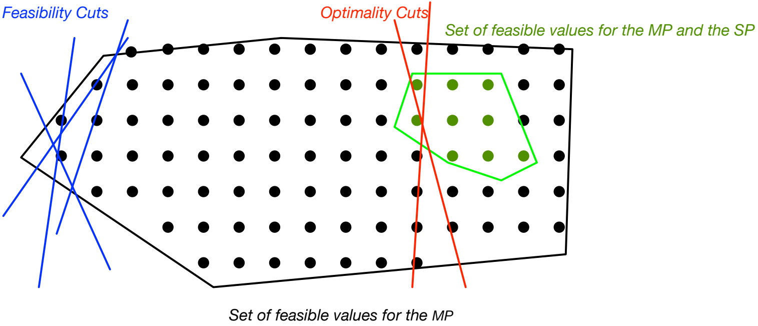



Figure: FCJSP example solved "2-Flex" heuristic.



Feasibility Cuts
Optimality Cuts
Set of feasible values for the MP and the SP
Set of feasible values for the MP


back