



# HIS<sup>3</sup>

Refresh your mind from stress ...

**Human-centered Interaction for Smarter Supervision  
of production Systems**

An iterative Constraint Programming approach  
to integrate **maximum workload constraints** in  
preemptive jobshop scheduling

**CP 2026** – Lisbon (*Tanguy TERRIEN, Cyrille BRIAND*)



# Introduction

*Context*



>300 000 elements  
3 years assembly  
Task diversity



# Introduction

Context



>300 000 elements  
3 years assembly  
Task diversity



DSS for dynamic  
project scheduling

\*Decision-Support System



# Introduction

Context



>300 000 elements  
3 years assembly  
Task diversity



Technocentrism



DSS for dynamic  
project scheduling

\*Decision-Support System



# Introduction

Context



>300 000 elements  
3 years assembly  
Task diversity



Technocentrism



Human-centered  
DSS



DSS for dynamic  
project scheduling

\*Decision-Support System



# Introduction

## Context



>300 000 elements  
3 years assembly  
Task diversity



DSS for dynamic  
project scheduling

\*Decision-Support System



Technocentrism



Human-centered  
DSS



Increased efficiency

Decreased  
cognitive load



# Summary

---

## ■ Considered problem

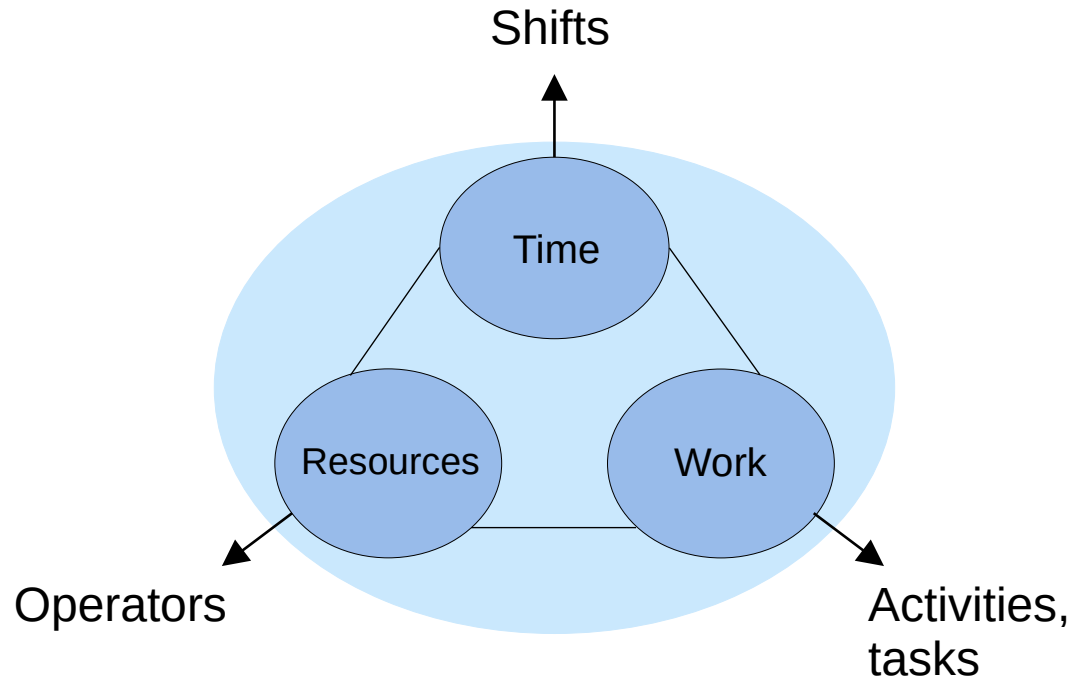
- ❑ Workload limitation
- ❑ Modelling
- ❑ Experimental results
- ❑ Perspectives





# Introduction

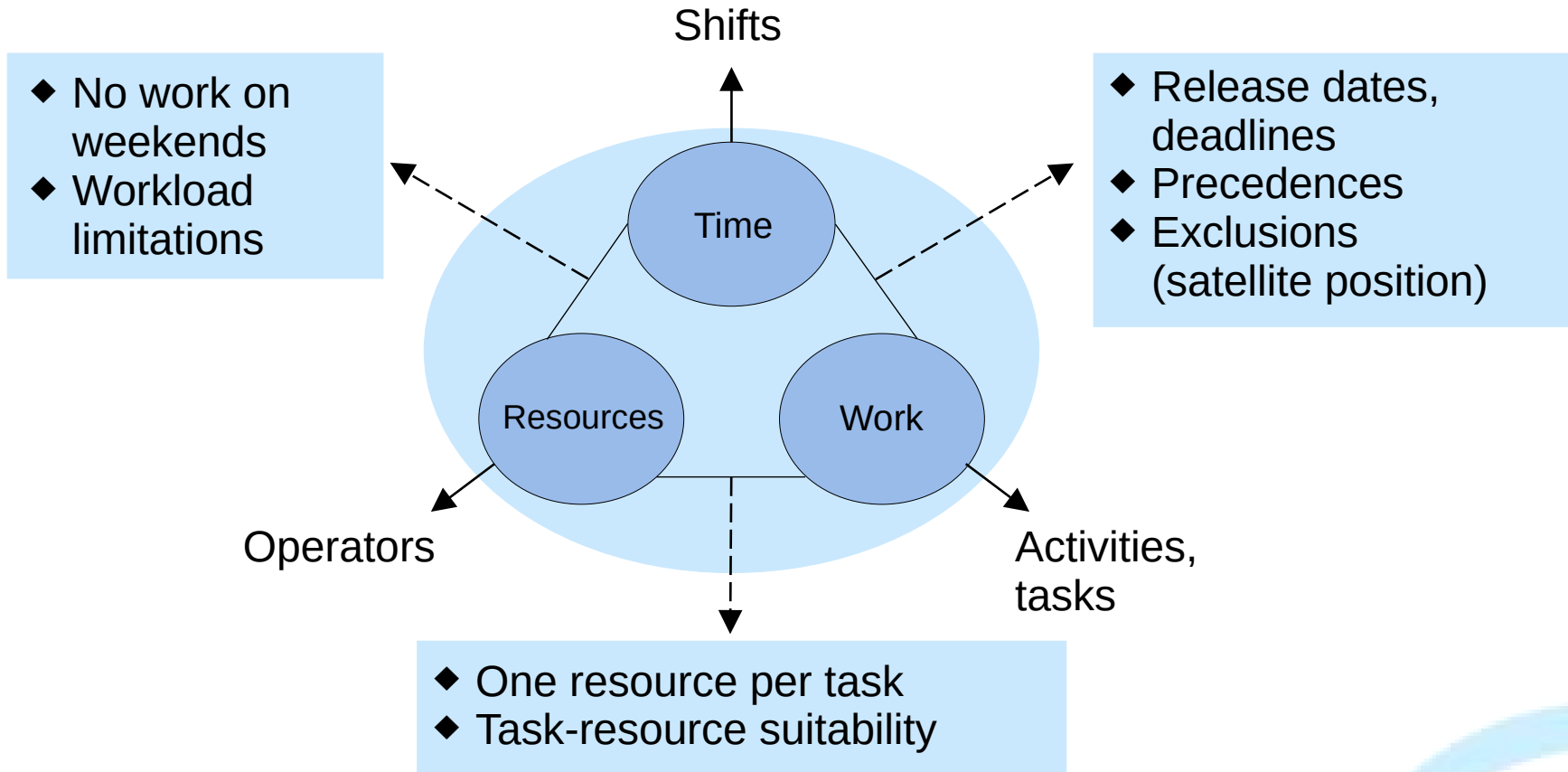
*Considered problem*





# Introduction

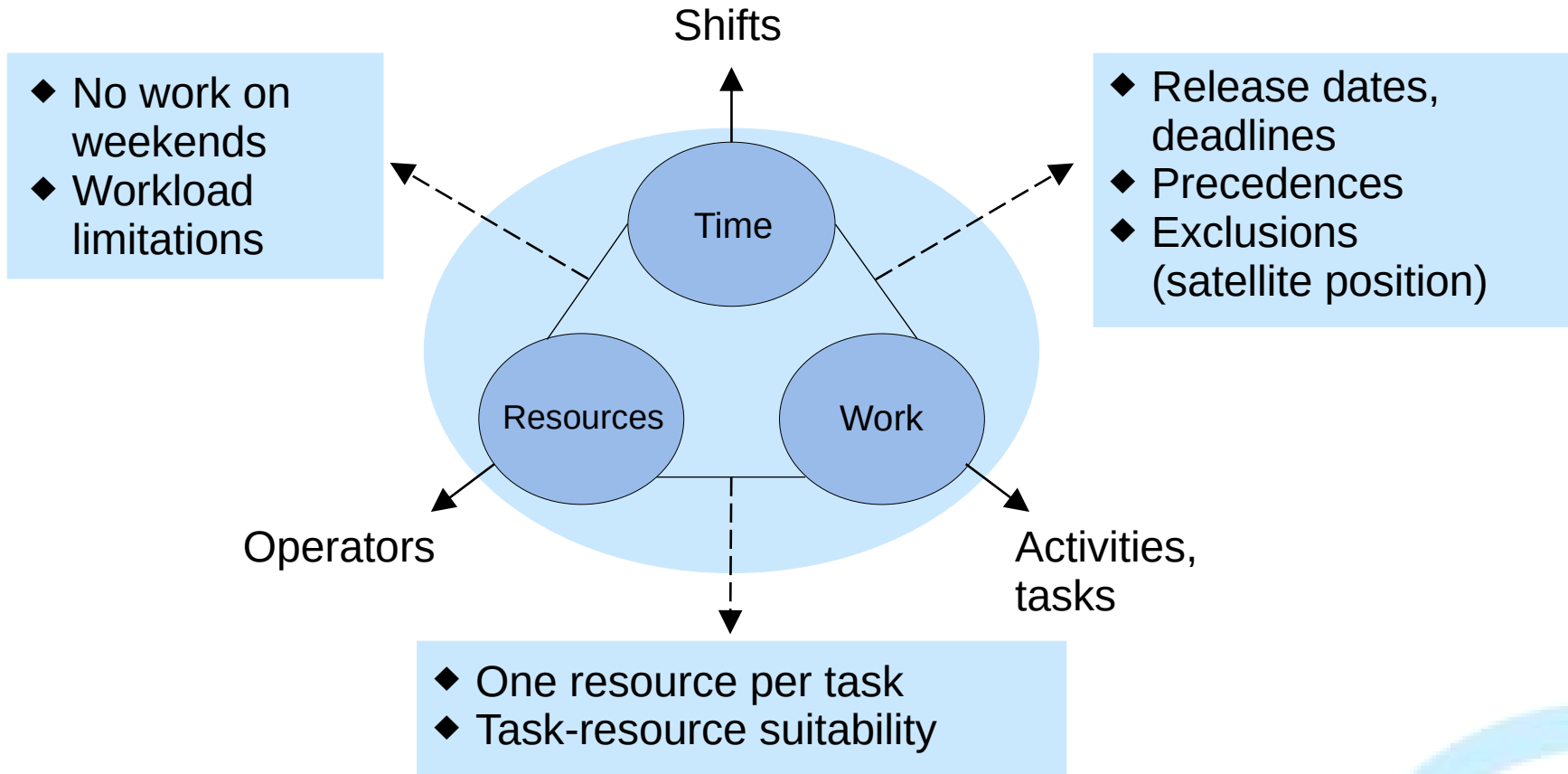
*Considered problem*





# Introduction

*Considered problem*

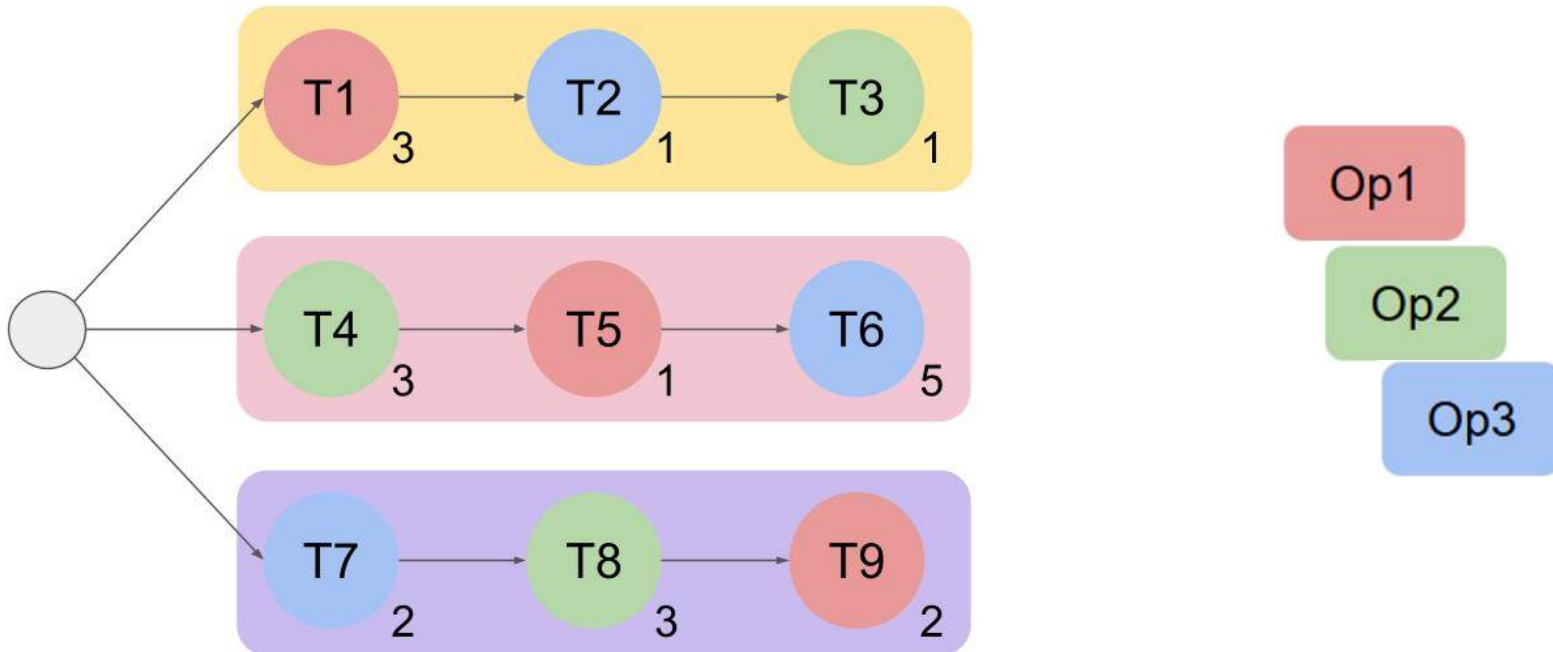


→ Preemptive Jobshop Scheduling Problem with additional constraints



# Illustrative example

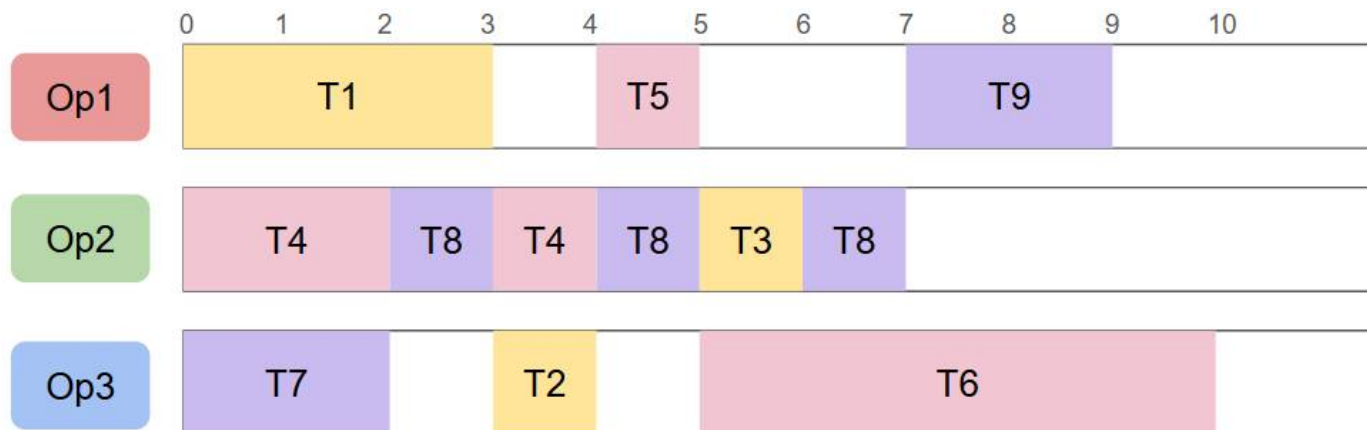
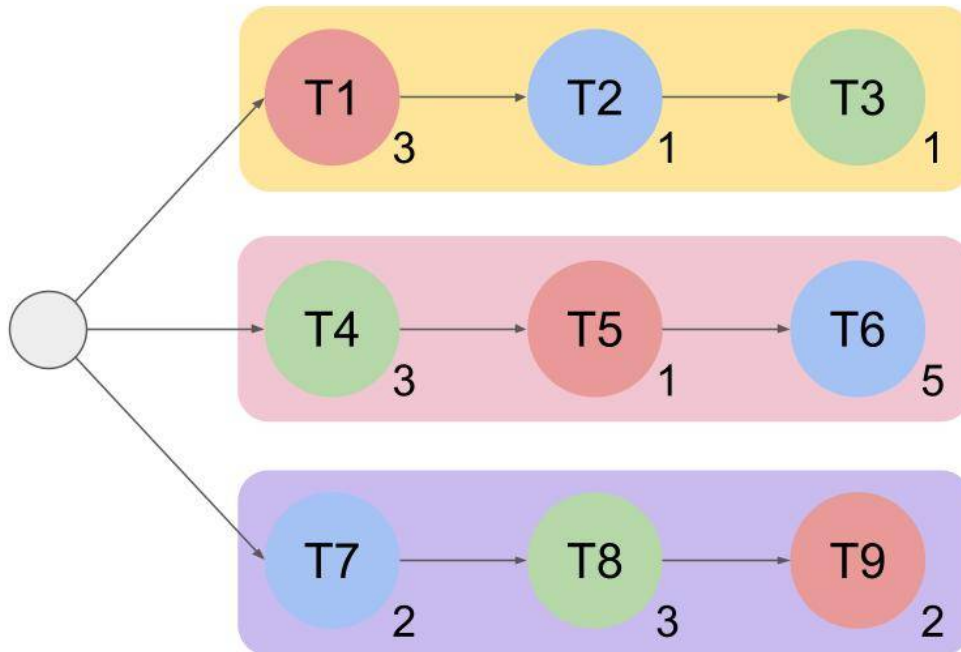
.. of the *preemptive Jobshop Scheduling Problem (pJSP)*





# Illustrative example

.. of the *preemptive Jobshop Scheduling Problem (pJSP)*



Solution example



# Summary

---

- ❑ Considered problem

## ■ Workload limitation

- ❑ Modelling

- ❑ Experimental results

- ❑ Perspectives



# Workload limitation

*Introducing a new constraint*

**MaxW** constraint :

$$\left( \delta_{uv}^k, k, [u, v] \right)$$

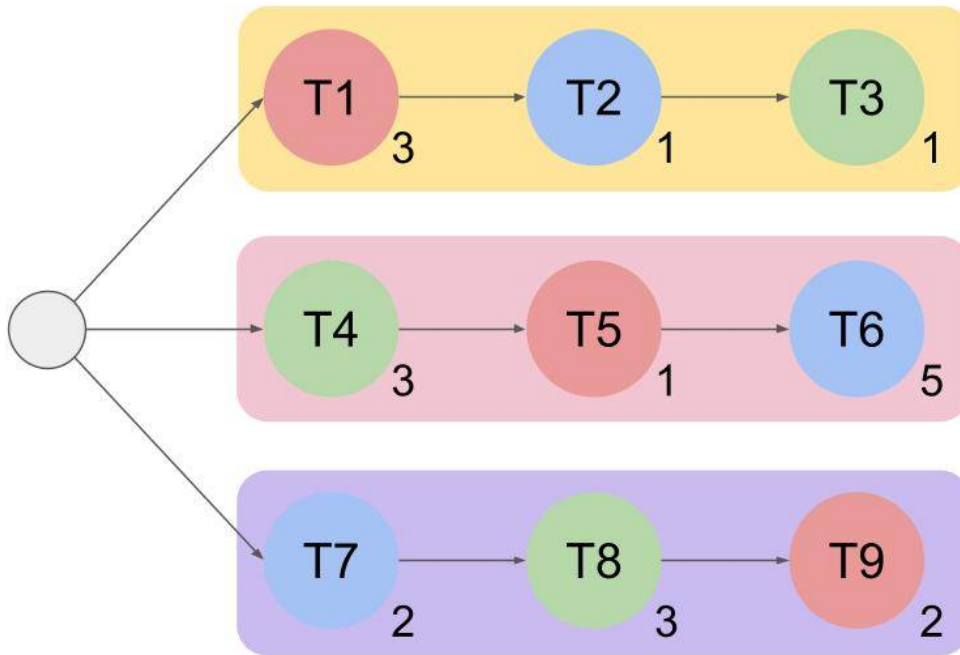
*“Operator  $k$  needs at least  $\delta_{uv}^k$  rest shifts within interval  $[u, v]$ ”*

*= at most  $(v - u - \delta_{uv}^k)$  working shifts*



# Illustrative example

.. of the *MaxW-pJSP*



## MaxW constraints

Op1 (4, Op1, [0, 10])  
(2, Op1, [5, 12])

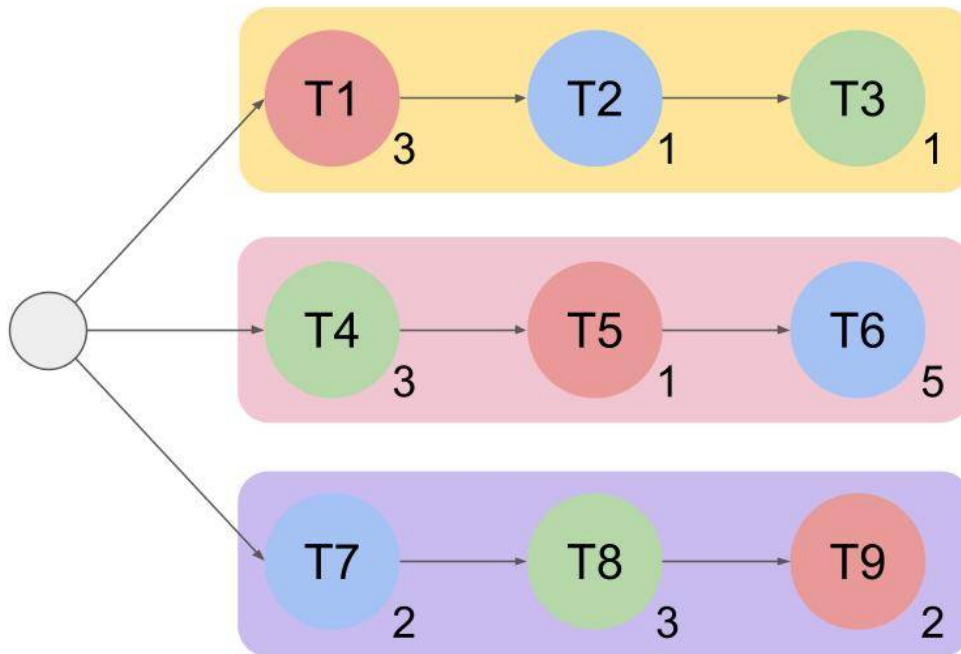
Op2 (2, Op2, [1, 8])  
(3, Op2, [2, 10])

Op3 (2, Op3, [0, 5])



# Illustrative example

.. of the *MaxW-pJSP*

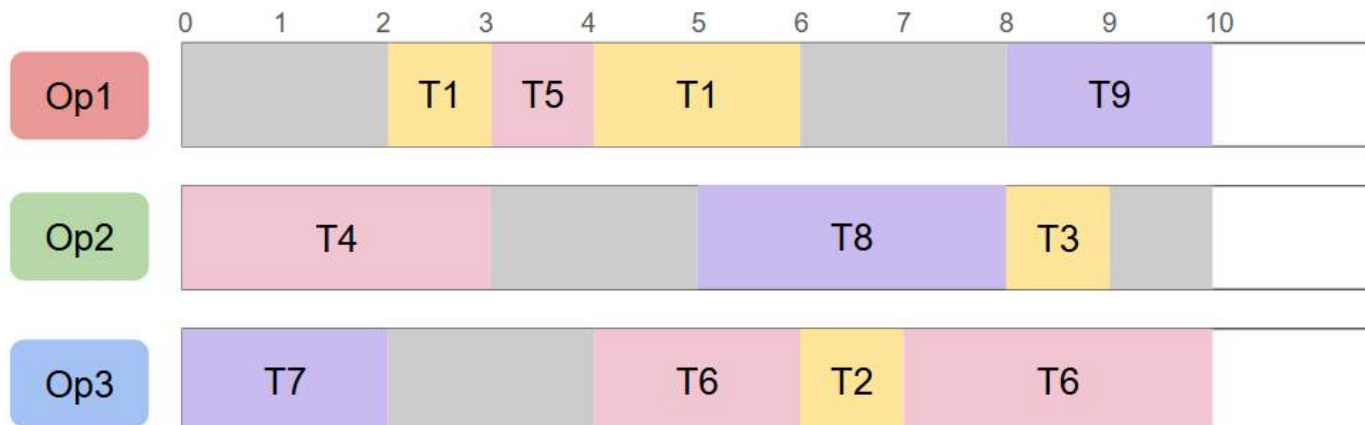


## MaxW constraints

Op1 (4, Op1, [0, 10])  
(2, Op1, [5, 12])

Op2 (2, Op2, [1, 8])  
(3, Op2, [2, 10])

Op3 (2, Op3, [0, 5])



An optimal solution



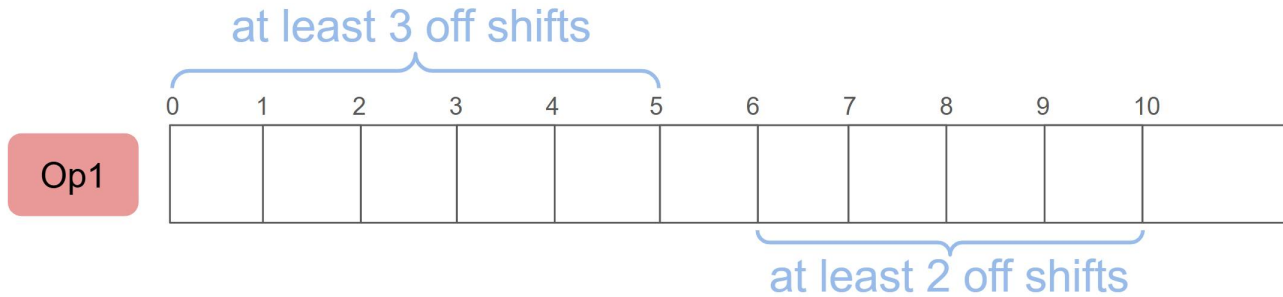
# Summary

---

- ❑ Considered problem
- ❑ Workload limitation
- **CP modelling : CPO vs Mistral**
- ❑ Experimental results
- ❑ Perspectives



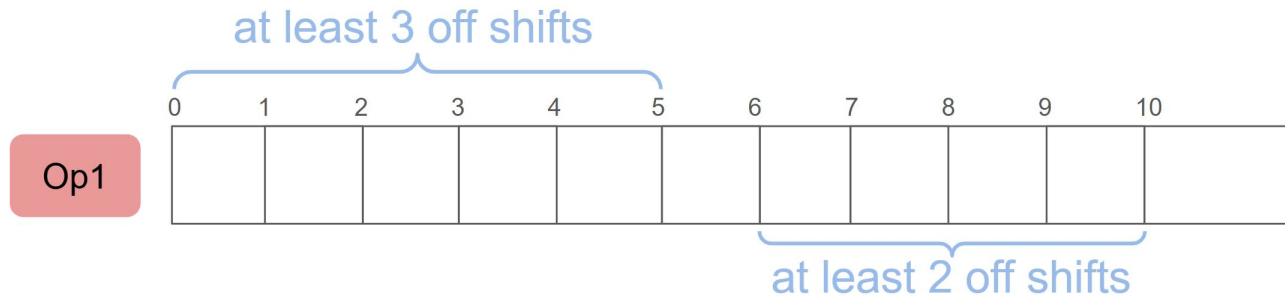
# Modelling



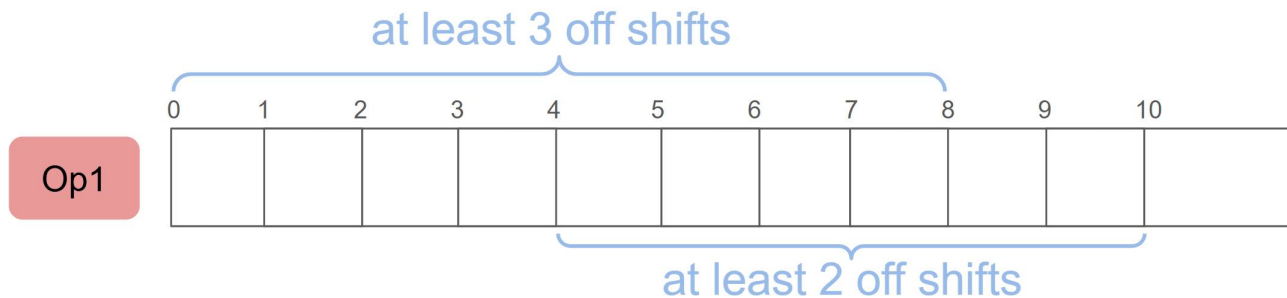
Easy ! :)  
→ Fake tasks



# Modelling



Easy ! :)  
→ Fake tasks



Hard :(



# Modelling

*with CPO*

---

CPO : Preemption 🚫



# Modelling

*with CPO*

---

CPO : Preemption 🚫

$W_{i,j,l}$  = unitary interval var for shift  $l$  of task  $j$  of job  $i$



# Modelling

*with CPO*

---

CPO : Preemption 

$W_{i,j,l}$  = *unitary interval var for shift l of task j of job i*

$O_{k,c,b}$  = *optional unitary interval var for shift b  
of MaxW constraint c of operator k*



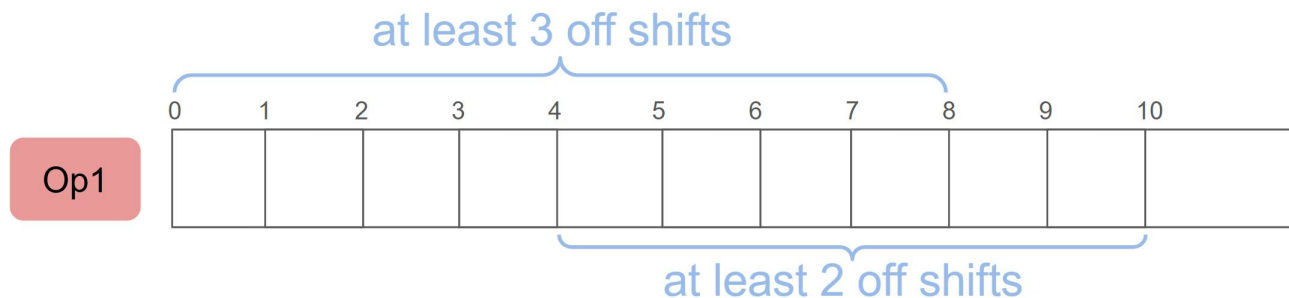
# Modelling

with CPO

CPO : Preemption 

$W_{i,j,l}$  = unitary interval var for shift  $l$  of task  $j$  of job  $i$

$O_{k,c,b}$  = optional unitary interval var for shift  $b$  of MaxW constraint  $c$  of operator  $k$



$O_1 : [0, 8]$   
 $O_2 : [0, 8]$   
 $O_3 : [0, 8]$   
 $O_4 : [4, 10]$   
 $O_5 : [4, 10]$

Presence( $O_1$ ) +  
Presence( $O_2$ ) +  
Presence( $O_3$ ) +  
 $O_4$  is present in  $[0, 8]$  +  
 $O_5$  is present in  $[0, 8] \geq 3$

$O_1$  is present in  $[4, 10]$  +  
 $O_2$  is present in  $[4, 10]$  +  
 $O_3$  is present in  $[4, 10]$  +  
Presence( $O_4$ ) +  
Presence( $O_5$ )  $\geq 2$



# Modelling

with CPO

CPO : Preemption 

$W_{i,j,l}$  = unitary interval var for shift  $l$  of task  $j$  of job  $i$

$O_{k,c,b}$  = optional unitary interval var for shift  $b$  of MaxW constraint  $c$  of operator  $k$

$$\min \max_{i,j} (\text{endOf}(W_{i,j,p_{ij}}))$$

interval  $W_{ij\ell}$ , size = 1

$$\forall i, \forall j, \forall \ell \in \{1, \dots, p_{ij}\}$$

interval  $O_{kc_{kz}\lambda}$ , optional, size = 1

$$\forall k, \forall c_{kz}, \forall \lambda \in \{1, \dots, \delta_z\}$$

$$O_{kc_{kz}\lambda} \subseteq [u_{kz}, v_{kz}]$$

$$\forall k, \forall c_{kz}, \forall \lambda \in \{1, \dots, \delta_z\}$$

ENDBEFORESTART( $W_{ij,\ell-1}, W_{ij,\ell}$ )

$$\forall i, \forall j, \forall \ell \in \{2, \dots, p_{ij}\}$$

ENDBEFORESTART( $W_{i,j,p_{ij}}, W_{i,j+1,0}$ )

$$\forall i, \forall j \in \{0, \dots, n_i - 1\}$$

NOOVERLAP( $[W_{ij\ell}]_{\forall i,j,\ell} \cup [O_{kc\lambda}]_{\forall c,\lambda}$ )

$$\forall k, j \in \text{tasks}_k$$

$$\sum_{\substack{y=1 \dots |C_k| \\ \lambda=1 \dots |\delta_y|}} \text{PRESENCE}(O_{kc_{ky}\lambda}) \cdot (O_{kc_{ky}\lambda} \in [u_{kz}, v_{kz}]) \geq \delta_z$$

$$\forall k, \forall c_{kz}$$

Minimize makespan

Workshifts variables

Offshifts variables

Ordering

Precedences

No ubiquity

MaxW constraints



# Modelling

with Mistral



Mistral : Preemption 👍 (*PreemptiveNoOverlap*)  
[Juvin et al., CP2023]



# Modelling

with Mistral

## Mistral : Preemption (*PreemptiveNoOverlap*)

[Juvin et al., CP2023]

$(s_j, e_j)$  = earliest start and  
latest end of task  $j$



# Modelling

with Mistral

## Mistral : Preemption (*PreemptiveNoOverlap*)

[Juvin et al., CP2023]

$(s_j, e_j)$  = earliest start and latest end of task  $j$

*No Overload* test

► Total duration of any set of tasks is no greater than its span

$$\sum_{t_j \in \Omega} p_j \leq \max(e_j \mid t_j \in \Omega) - \min(s_j \mid t_j \in \Omega)$$

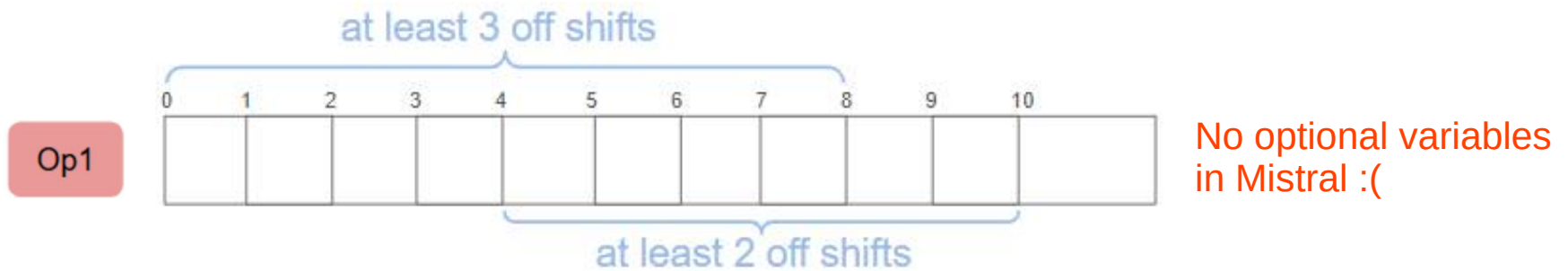
+ JacksonPreemptiveScheduler



# Modelling

with Mistral

Handle **MaxW** constraints:

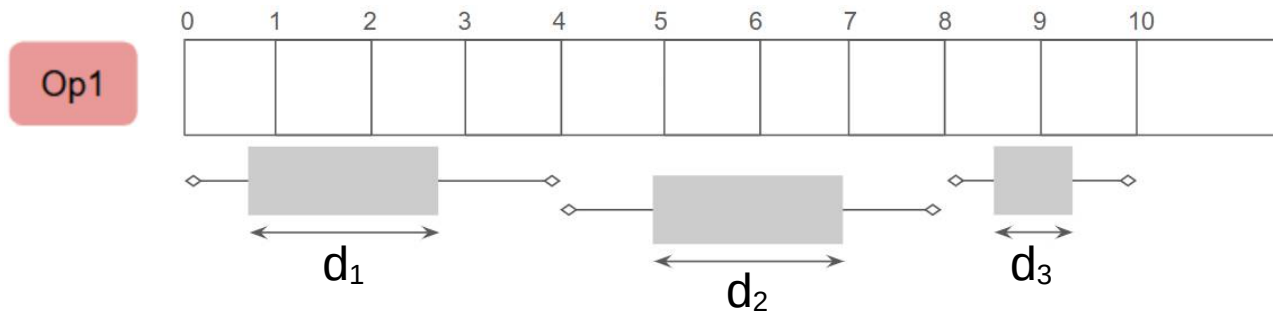
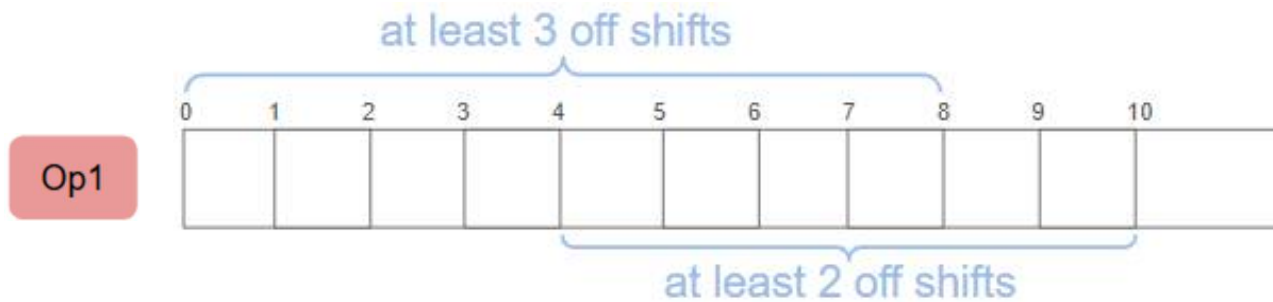




# Modelling

with Mistral

Handle **MaxW** constraints:

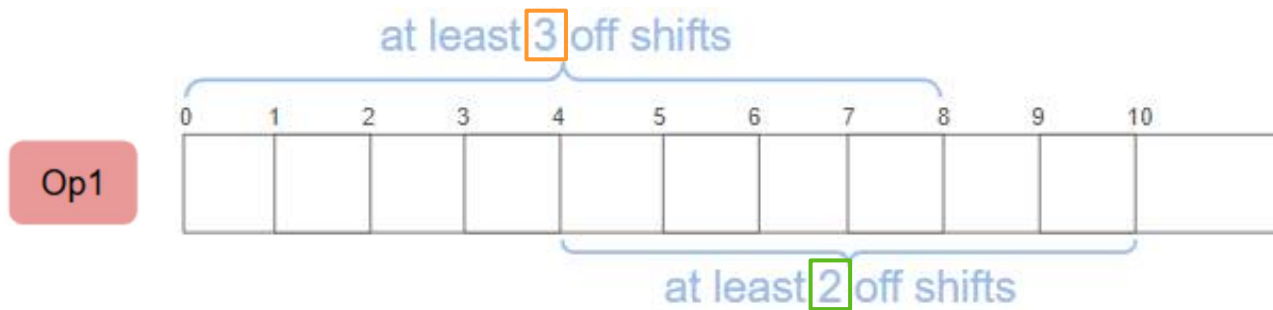




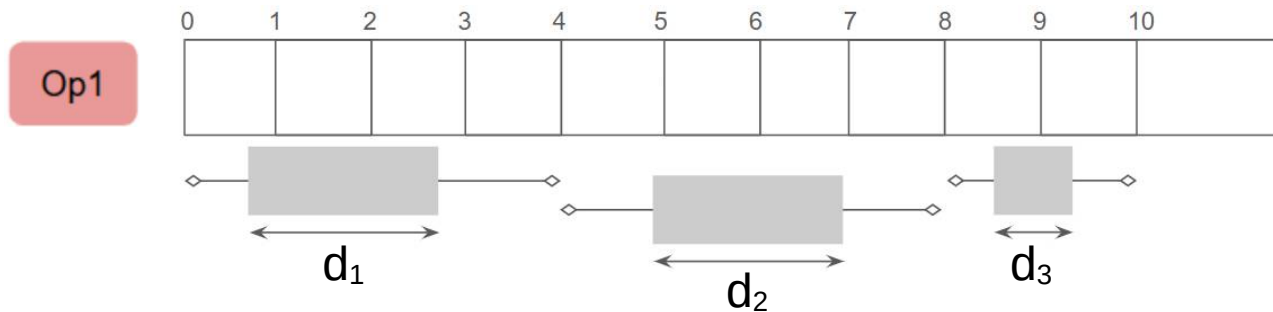
# Modelling

with Mistral

Handle **MaxW** constraints:



$$\begin{aligned} d_1 + d_2 &\geq 3 \\ d_2 + d_3 &\geq 2 \end{aligned}$$





# Modelling

pJSP with Mistral

$$\min c_{max}$$

$$s_{i,j} \in [0, UB - P_{i,j}], \quad e_{i,j} \in [P_{i,j}, UB] \quad \forall i, j \quad (V1)$$

$$c_{max} \in [0, UB] \quad (V2)$$

$$e_{i,j} \geq s_{i,j} + P_{i,j} \quad \forall i, \forall j \quad (C1)$$

$$s_{i,j+1} \geq e_{i,j} \quad \forall i, \forall j \quad (C2)$$

$$s_{i,0} = 0 \quad \text{and} \quad e_{i,n_i} \leq c_{max} \quad \forall i \quad (C3)$$

$$\text{PRE.NOOVERLAP} \left( \left\{ (s_{i,j}, e_{i,j}, P_{i,j}) \right\}_{\forall i, \forall j \in T_k} \right) \quad \forall k \quad (C4)$$



# Modelling

MaxW-pJSP with Mistral

$$\min c_{max}$$

$$s_{i,j} \in [0, UB - P_{i,j}], \quad e_{i,j} \in [P_{i,j}, UB] \quad \forall i, j \quad (V1)$$

$$c_{max} \in [0, UB] \quad (V2)$$

$$d_q^k \in [0, |q|] \quad \forall k, \forall q \quad (V3)$$

$$e_{i,j} \geq s_{i,j} + P_{i,j} \quad \forall i, \forall j \quad (C1)$$

$$s_{i,j+1} \geq e_{i,j} \quad \forall i, \forall j \quad (C2)$$

$$s_{i,0} = 0 \quad \text{and} \quad e_{i,n_i} \leq c_{max} \quad \forall i \quad (C3)$$

$$\text{PRE.NOOVERLAP} \left( \left\{ (s_{i,j}, e_{i,j}, P_{i,j}) \right\}_{\forall i, \forall j \in T_k} \cup \left\{ (S_q, E_q, d_q^k) \right\}_{\forall q} \right) \quad \forall k \quad (C4)$$

$$\left( V_c^k - U_c^k \right) - \delta_c^k \leq \sum_{\forall q \in [U_c^k, V_c^k]} d_q^k \quad \forall k, \forall c \in \text{MaxW}^k \quad (C5)$$

$$d_q^k \leq \max_{\substack{\forall c \in \text{MaxW}^k \\ \text{st. } q \in [U_c^k, V_c^k]}} \left( V_c^k - U_c^k \right) - \delta_c^k \quad \forall k, \forall q \quad (C6)$$



# Preliminary results

*Mistral vs CPO*

- 78 pJSP instances of OR-Library
- 9 MaxW instances per pJSP instance  
→ **78\*9 = 702 instances**



**45 min**



# Preliminary results

*Mistral vs CPO*

- 78 pJSP instances of OR-Library
- 9 MaxW instances per pJSP instance  
→ **78\*9 = 702 instances**



**45 min**

| Solver  | # Optimality proofs |
|---------|---------------------|
| CPO     | 35 (4.98%)          |
| Mistral | <b>171 (24.3%)</b>  |



# Preliminary results

*Mistral vs CPO*

- 78 pJSP instances of OR-Library
  - 9 MaxW instances per pJSP instance
- **78\*9 = 702 instances**



**45 min**

| Solver  | # Optimality proofs |
|---------|---------------------|
| CPO     | 35 (4.98%)          |
| Mistral | <b>171 (24.3%)</b>  |

*How to deal with the exponential number of subintervals generated by the MaxW constraints decomposition ?*



# Classical approach

*For a Constraint Optimization Problem*

---

```
graph LR; A[Start with all MaxW] --> B(Solve with all MaxW)
```

Start with  
all MaxW

Solve with  
all MaxW



# Classical approach

*For a Constraint Optimization Problem*

Start with  
all MaxW

Solve with  
all MaxW

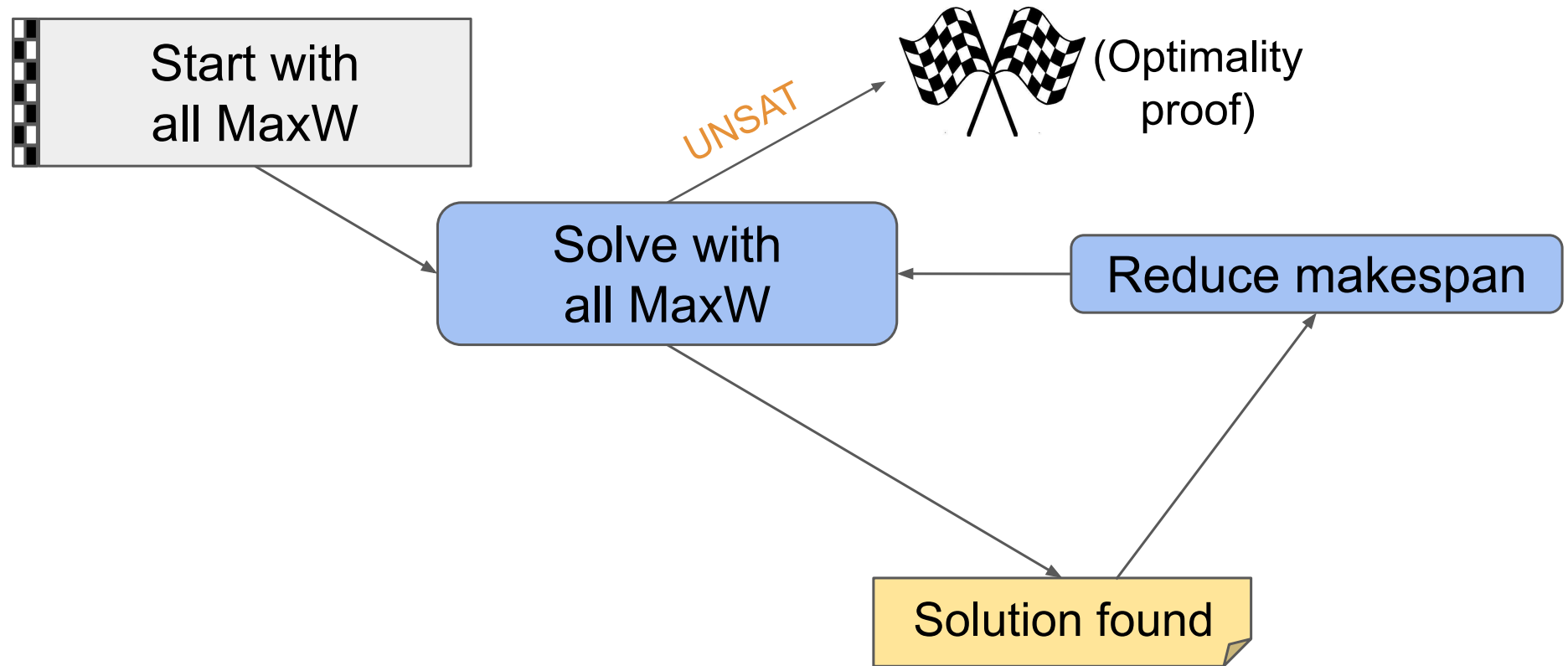
Reduce makespan

Solution found



# Classical approach

*For a Constraint Optimization Problem*





# Iterative approach

*Lazy Constraint Generation*

---



# Iterative approach

*Lazy Constraint Generation*

---

Start with empty  
MaxW set

Solve with  
current MaxW set

Are all MaxW verified ?



# Iterative approach

*Lazy Constraint Generation*

Start with empty  
MaxW set

Solve with  
current MaxW set

Activate some  
MaxW (heuristic)

Are all MaxW verified ?

NO



- check violation of each MaxW
- activate most violated  $C$
- forget about MaxWs overlapping  $C$
- activate most violated
- ...



# Iterative approach

*Lazy Constraint Generation*

Start with empty  
MaxW set

Solve with  
current MaxW set

Reduce makespan

Activate some  
MaxW (heuristic)

Are all MaxW verified ?

YES

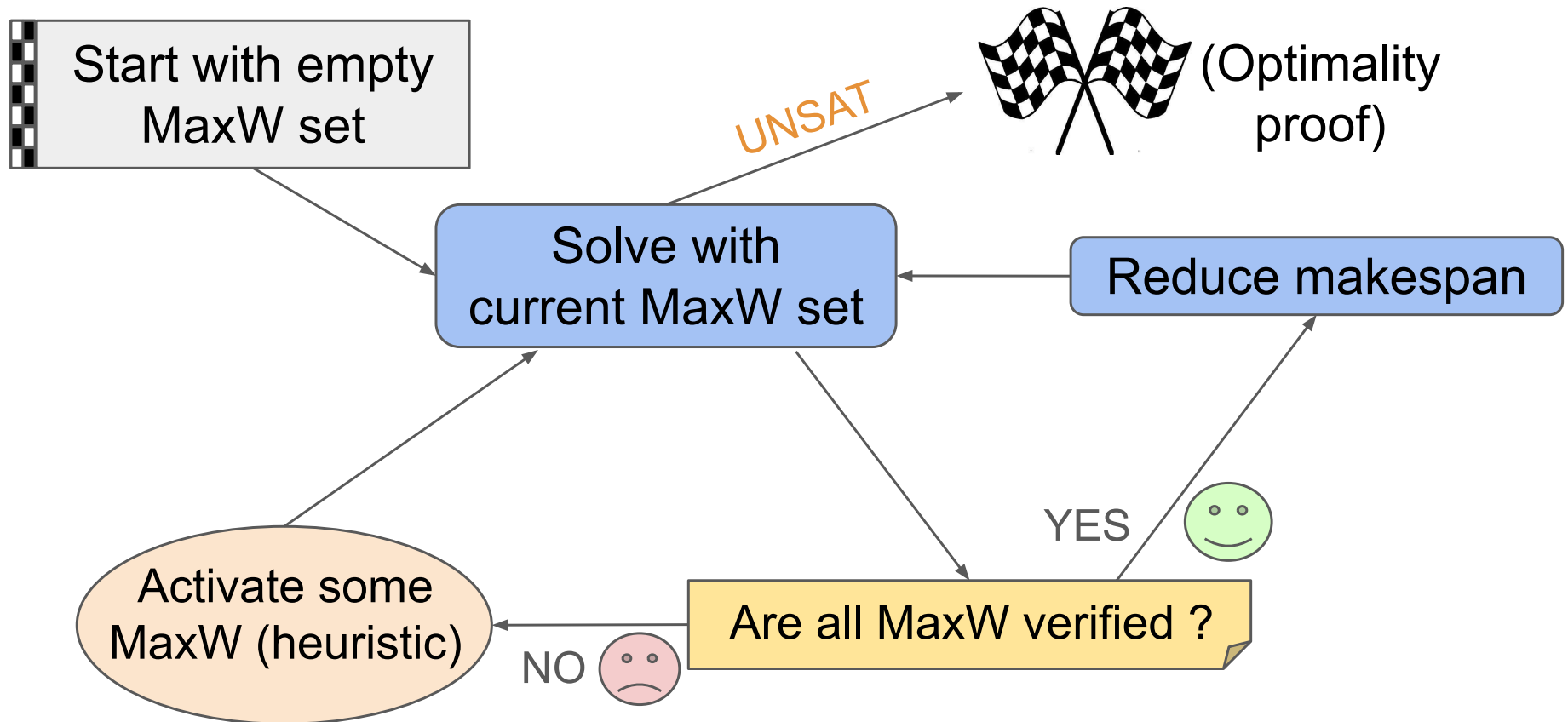
NO





# Iterative approach

*Lazy Constraint Generation*





# Summary

---

- ☐ Considered problem
- ☐ Workload limitation
- ☐ Modelling

## ☒ Experimental results

- ☐ Perspectives



# Number of optimal solutions

CPO vs Mistral vs Iterative

**78\*9 = 702 instances**



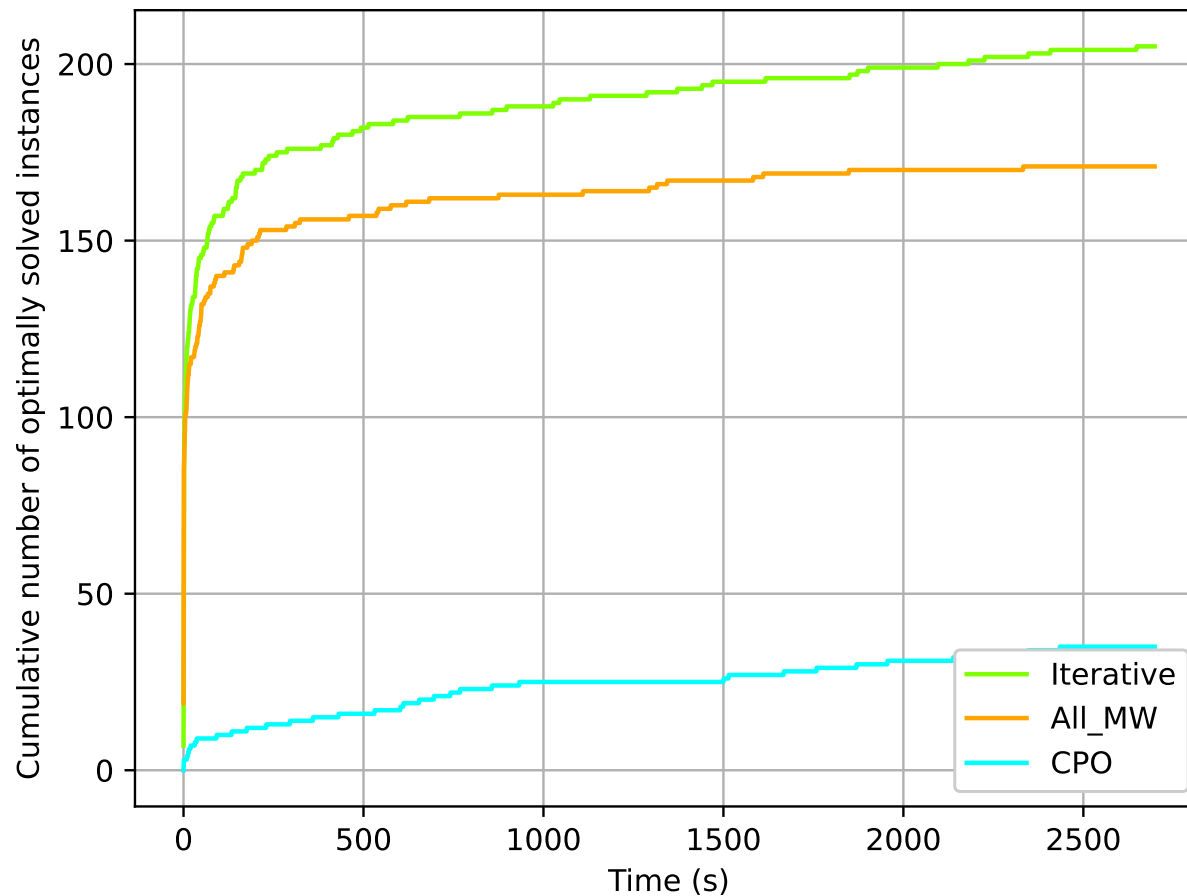
**45 min**

| Solver                    | # Optimality proofs |
|---------------------------|---------------------|
| CPO                       | 35 (4.98%)          |
| Mistral                   | 171 (24.3%)         |
| Mistral- <i>Iterative</i> | <b>205 (29.2%)</b>  |



# Time

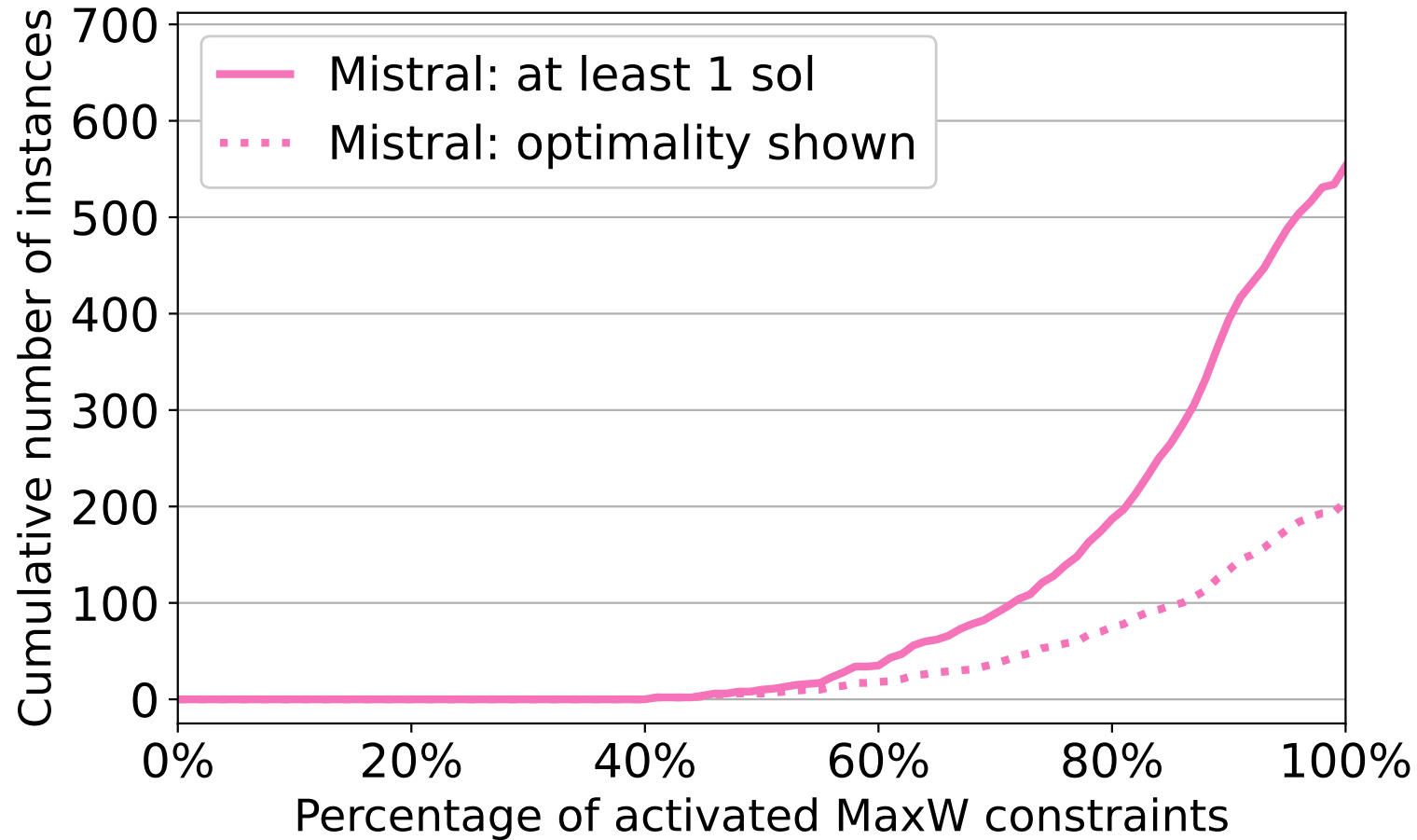
*CPO vs Mistral vs Iterative*





# Added proportion of MaxW

*Iterative*





# New solver ! “*TEMPO*”

---

- new constraint programming lazy clause and literal generation solver
- developed by Emmanuel Hébrard in LAAS-CNRS (Toulouse)
- **PreemptiveNoOverlap** implemented :)
- same model as Mistral



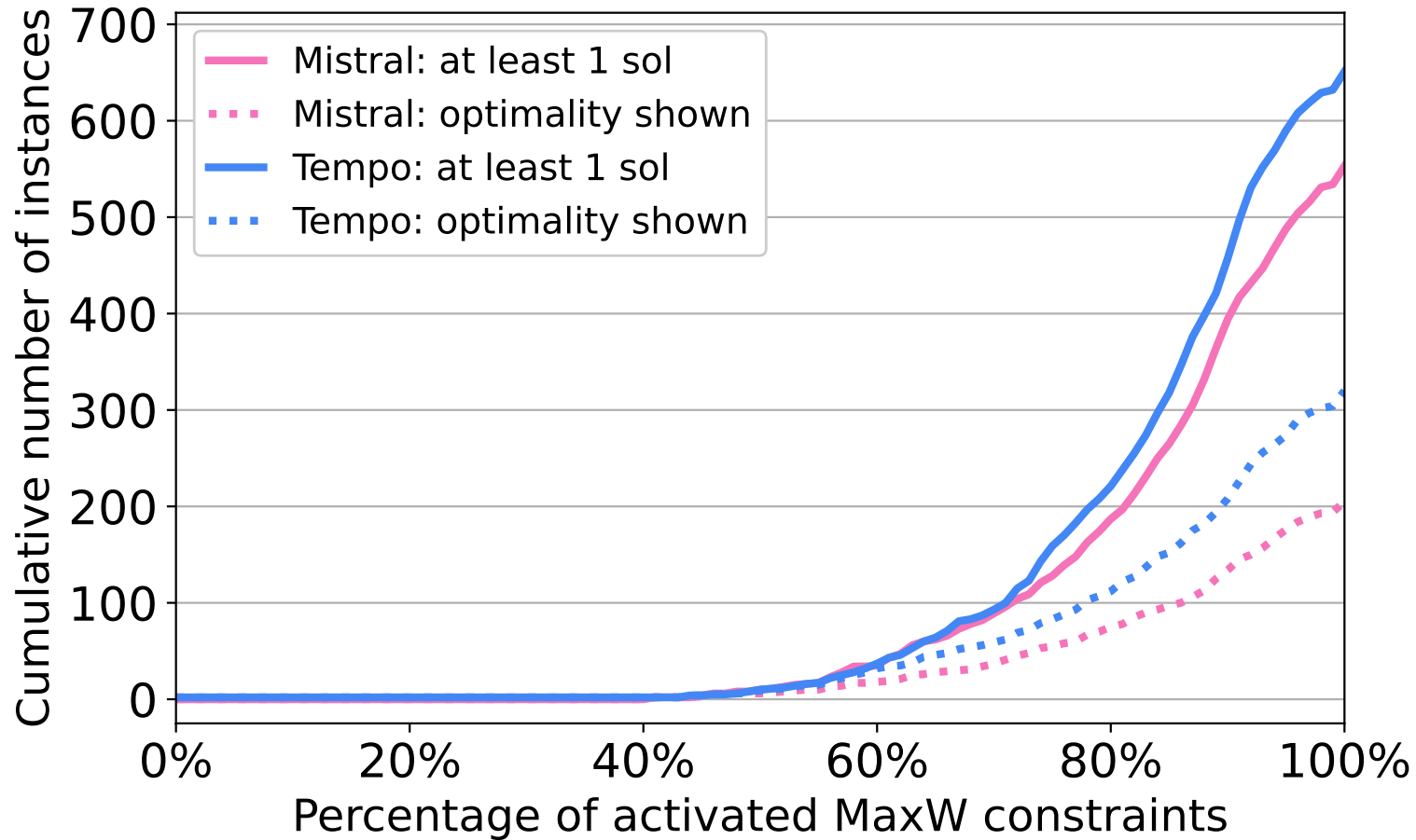
# New solver ! “*TEMPO*”

- new constraint programming lazy clause and literal generation solver
- developed by Emmanuel Hébrard in LAAS-CNRS (Toulouse)
- **PreemptiveNoOverlap** implemented :)
- same model as Mistral

| Solver                    | # Optimality proofs |
|---------------------------|---------------------|
| CPO                       | 35 (4.98%)          |
| Mistral                   | 171 (24.3%)         |
| Mistral- <i>Iterative</i> | 205 (29.2%)         |
| TEMPO                     | 303 (43.2%)         |
| TEMPO- <i>Iterative</i>   | <b>320 (45.6%)</b>  |



# Iterative : *Mistral* VS *TEMPO*





# Takeaways

---





# Takeaways

---

2

1

3

**PreemptiveNoOverlap**  
is great for preemptive  
scheduling



# Takeaways

---

**TEMPO**  
is great

2

1

**PreemptiveNoOverlap**  
is great for preemptive  
scheduling

3



# Takeaways

---

**Lazy Constraint Generation**  
can be great if some  
constraints can be verified  
without being enforced in the  
model

**TEMPO**  
is great

**PreemptiveNoOverlap**  
is great for preemptive  
scheduling

2

1

3



# Summary

---

- ☐ Considered problem
- ☐ Workload limitation
- ☐ Modelling
- ☐ Experimental results

## ☒ Perspectives



# Perspectives

---

## Operations research

- Try to make use of Tempo's clause learning
- Study different MaxW constraint addition heuristic
- Integrate MinW constraints (*not symmetric*)



# Questions ?

---

Tanguy.Terrien@laas.fr

My paper



Disjunctive scheduling  
on **TEMPO** (CP25)





# Appendix

---



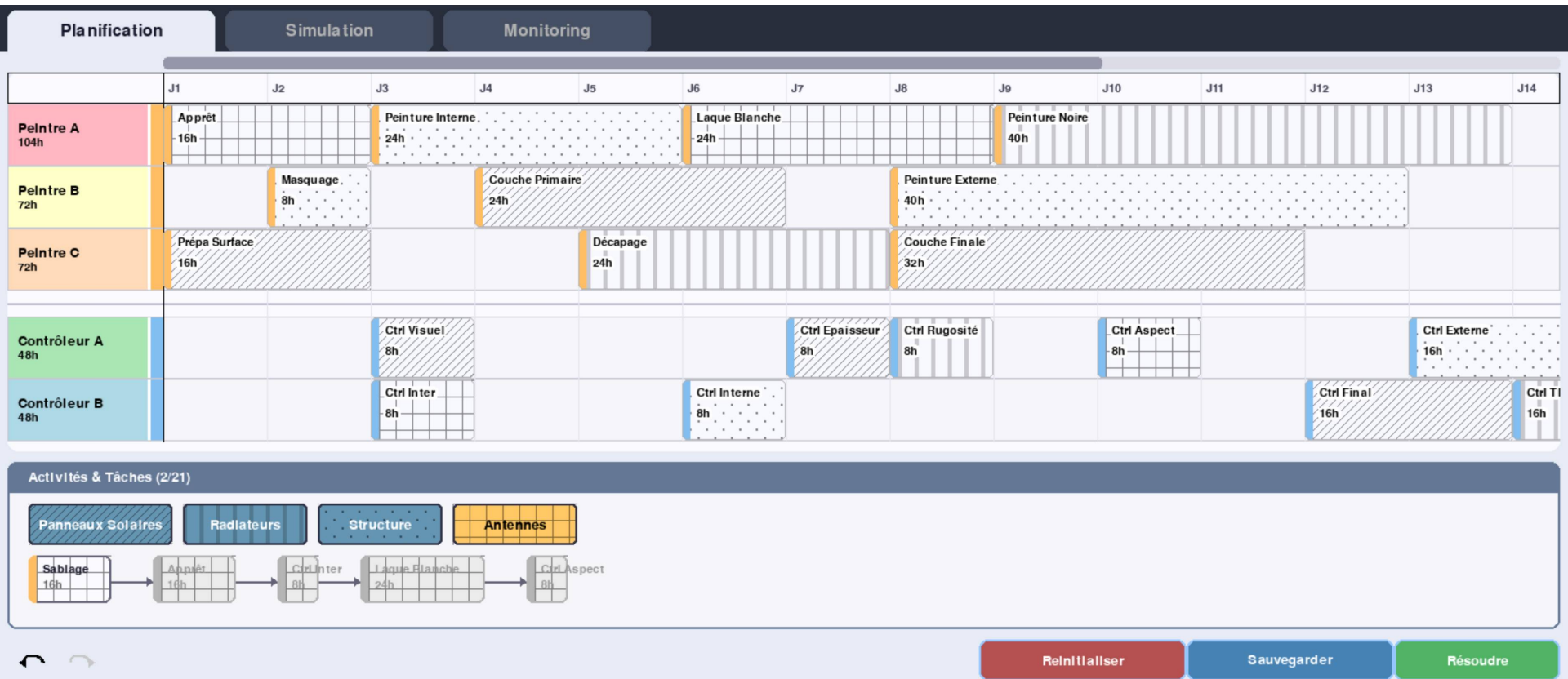


# Future work

## HMI

### Human-Machine Interaction

- Develop a user interface that is in line with both optimization functionalities and cognitive-wise needs (*ongoing*)





# Future work

*Cognitive science (+ HMI)*

---

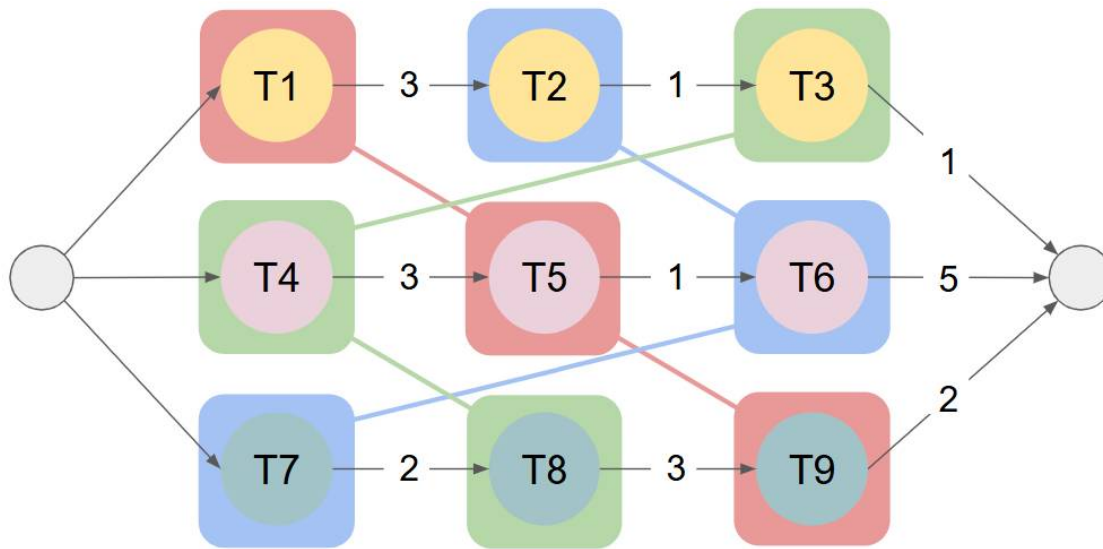
Cognitive science (+ HMI)

- Evaluate the effect of a decision-support system on users' performances and cognitive and emotional state (*ongoing*)



# Illustrative example

..of the *Preemptive Jobshop Scheduling problem (pJSP)*

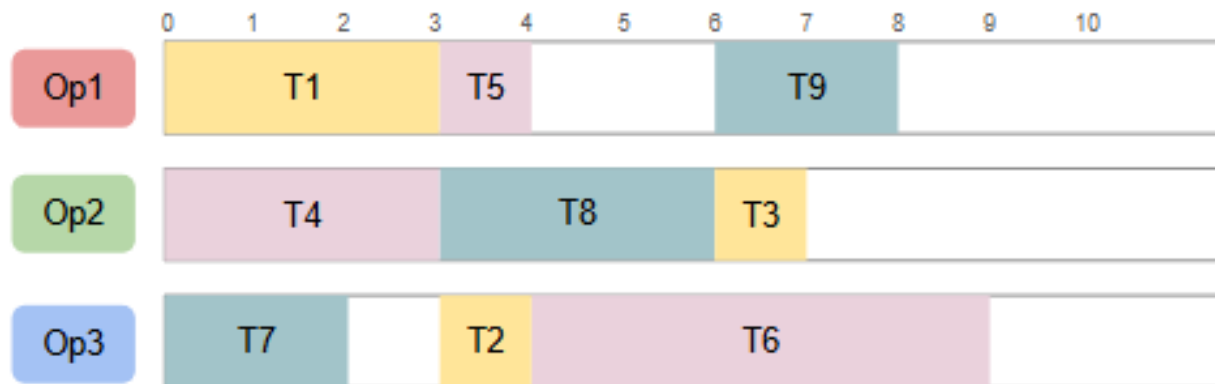


Op1  $3+1+2 = 6$

Op2  $3+3+1 = 7$

Op3  $2+1+5 = 8$

## Optimal solution



Operators LB:  $\max(6, 7, 8)$

Jobs LB:

$$\max(3 + 1 + 1, 3 + 1 + 5, 2 + 3 + 2)$$

Global LB: 9



# Summary

---

- ❑ Considered problem
- ❑ Workload limitations
- ❑ Modelling : CPO vs Mistral
- ❑ Experimentations

## ■ Specific case



# Specific case

*Another new constraint*

**RMaxW** constraint :

$$\left( \delta_{uv,N}^k, k, N, [u, v] \right)$$

*“On  $[u, v]$ , operator  $k$  should have at least  $\delta_{uv,N}^k$  off shifts in any  $N$  consecutive shifts”*



# Specific case

*Another new constraint*

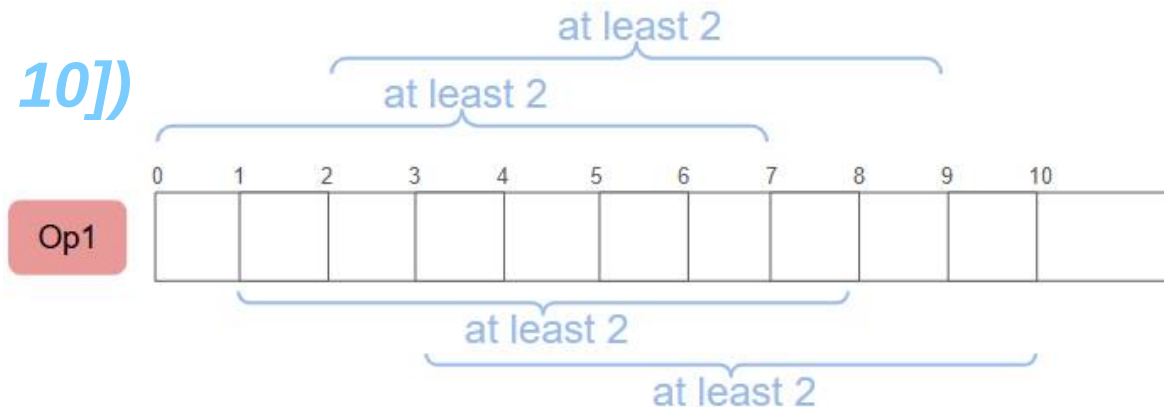
**RMaxW** constraint :

$$\left( \delta_{uv,N}^k, k, N, [u, v] \right)$$

*“On  $[u, v]$ , operator  $k$  should have at least  $\delta_{uv,N}^k$  off shifts in any  $N$  consecutive shifts”*

$$\Leftrightarrow (\delta_{uv,N}^k, k, [u, u + N]) \quad \forall u \in \{1, \dots, v - N\}$$

**(2, 7, [0, 10])**





# Modelling of RMaxW

*A better method*

---

**Maximum time lag :**

Maximum time between two tasks or two occurrences of a repeated task



# Modelling of RMaxW

*A better method*

**Maximum time lag :**

Maximum time between two tasks or two occurrences of a repeated task

→ We add  $Z$  unitary fictitious tasks to the whole horizon of **RMaxW**



# Modelling of RMaxW

*A better method*

## Maximum time lag :

Maximum time between two tasks or two occurrences of a repeated task

→ We add  $\mathcal{Z}$  unitary fictitious tasks to the whole horizon of **RMaxW**

- ordering of unitary *fictitious* tasks:

$$e_z \leq s_{z+1} \quad \forall z \in \{1, \dots, \mathcal{Z} - 1\}$$



# Modelling of RMaxW

*A better method*

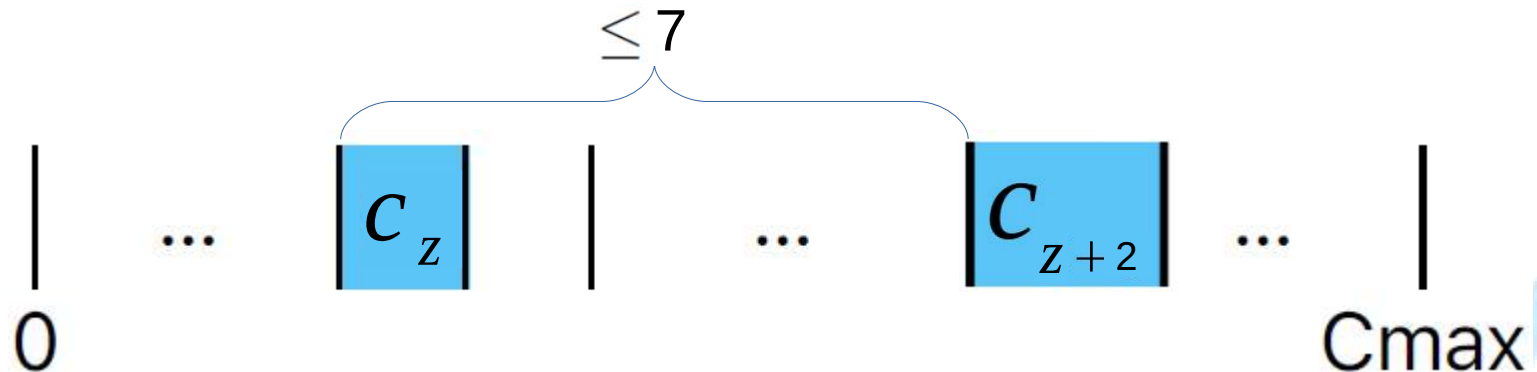
## Maximum time lag :

Maximum time between two tasks or two occurrences of a repeated task

→ We add  $\mathcal{Z}$  unitary fictitious tasks to the whole horizon of **RMaxW**

- ordering of unitary *fictitious* tasks
- ensuring that the **RMaxW** constraint is verified:

$$s_{z+\delta} - s_z \leq N \quad \forall z \in \{1, \dots, \mathcal{Z} - \delta\}$$





# Modelling of RMaxW

*A better method*

## Maximum time lag :

Maximum time between two tasks or two occurrences of a repeated task

→ We add  $Z$  unitary fictitious tasks to the whole horizon of **RMaxW**

- ordering of unitary *fictitious* tasks
- ensuring that the **RMaxW** constraint is verified
- enforce off shifts to be well divided along the horizon:

$$e_{\delta} \leq N$$

$$s_{Z-\delta} \geq C_{\max} - N$$





# CPO model

$$\min \max_{i,j} (\text{endOf}(W_{i,j,p_{ij}}))$$

$$\text{interval } W_{ij\ell}, \text{ size} = 1 \quad \forall i, \forall j, \forall \ell \in \{1, \dots, p_{ij}\} \quad (\text{V1})$$

$$\text{interval } O_{kc_{kz}\lambda}, \text{ optional}, \text{ size} = 1 \quad \forall k, \forall c_{kz}, \forall \lambda \in \{1, \dots, \delta_z\} \quad (\text{V2})$$

$$O_{kc_{kz}\lambda} \subseteq [u_{kz}, v_{kz}] \quad \forall k, \forall c_{kz}, \forall \lambda \in \{1, \dots, \delta_z\} \quad (\text{C1})$$

$$\text{ENDBEFORESTART}(W_{ij,\ell-1}, W_{ij,\ell}) \quad \forall i, \forall j, \forall \ell \in \{2, \dots, p_{ij}\} \quad (\text{C2})$$

$$\text{ENDBEFORESTART}(W_{i,j,p_{ij}}, W_{i,j+1,0}) \quad \forall i, \forall j \in \{0, \dots, n_i - 1\} \quad (\text{C3})$$

$$\text{NOOVERLAP}([W_{ij\ell}]_{\forall i,j,\ell} \cup [O_{kc\lambda}]_{\forall c,\lambda}) \quad \forall k, j \in \text{tasks}_k \quad (\text{C4})$$

$$\sum_{\substack{y=1 \dots |C_k| \\ \lambda=1 \dots |\delta_y|}} \text{PRESENCE}(O_{kc_{ky}\lambda}) \cdot (O_{kc_{ky}\lambda} \in [u_{kz}, v_{kz}]) \geq \delta_z \quad \forall k, \forall c_{kz} \quad (\text{C5})$$



# Modèle complet Mistral

$$\min C_{\max}$$

$$s_{i,j} \in [0, \text{ub} - p_{i,j}], \quad e_{i,j} \in [p_{i,j}, \text{ub}] \quad \forall i, j \quad (\text{V1})$$

$$C_{\max} \in [0, \text{ub}] \quad (\text{V2})$$

$$d_{kq} \in [0, |q|] \quad \forall k, \forall q \quad (\text{V3})$$

$$e_{i,j} \geq s_{i,j} + p_{i,j} \quad \forall i, \forall j \quad (\text{C1})$$

$$s_{i,j+1} \geq e_{i,j} \quad \forall i, \forall j \quad (\text{C2})$$

$$s_{i,0} = 0 \quad \text{and} \quad e_{i,n_i} = C_{\max} \quad \forall i \quad (\text{C3})$$

$$\text{PRE.NOOVERLAP} \left( \{(s_{i,j}, e_{i,j}, p_{i,j})\} \cup \{(start_q, end_q, d_{kq})\} \right) \quad \forall i, \forall j \in \text{tasks}_k, \forall k, \forall q \quad (\text{C4})$$

$$\sum_{q \in [u_z, v_z]} d_{kq} \geq \delta_z \quad \forall k, \forall q, \forall (\delta_z, u_z, v_z) \in \mathcal{C}_k \quad (\text{C5})$$

$$d_{kq} \leq \max(\delta) \quad \forall k, \forall (\delta_z, u_z, v_z) \in \mathcal{C}_k, \forall q \in [u, v] \quad (\text{C6})$$



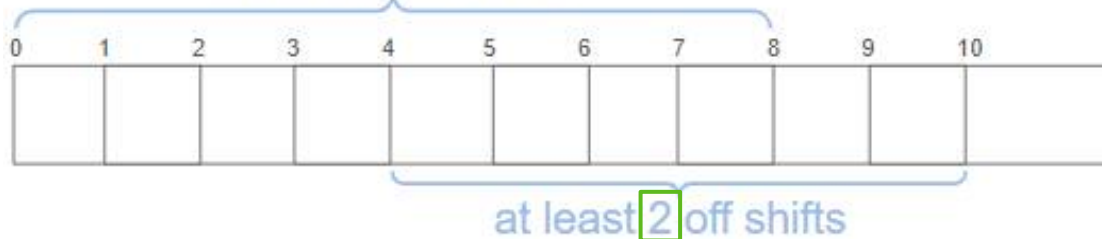
# Modelling

with Mistral

## Handling MaxW constraints :

at least 3 off shifts

Op1



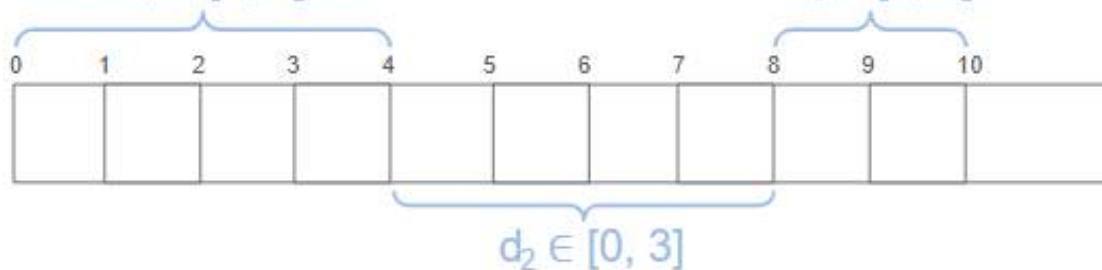
Original task

Fixed duration,  
variable start/end

$d_1 \in [0, 3]$

$d_3 \in [0, 2]$

Op1



Added rest task

Fixed start/end,  
variable duration

$q_1$ : start at 0, end  
at 4, duration  $d_1$

$q_2$ : start at 4, end  
at 8, duration  $d_2$

$q_3$ : start at 8, end  
at 10, duration  $d_3$

$$d_1 + d_2 \geq \text{3}$$

$$d_1 + d_3 \geq \text{2}$$