# Fixed Parameter Tractability of Scheduling Dependent Typed Tasks with time windows

Claire Hanen [1,2] Alix Munier Kordon [1]

[1]Sorbonne université, CNRS, LIP6, F-75005, Paris, France
[2]UPL, Université Paris Nanterre, 92000, Nanterre, France

November 15, 2023

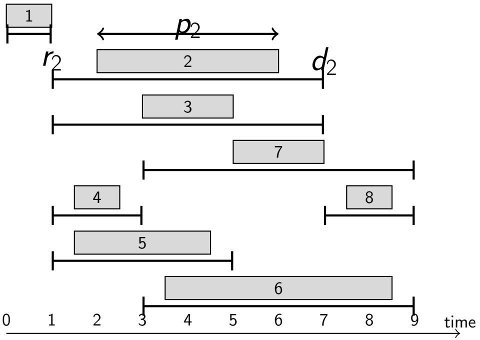
Scheduling Seminar

# Contents

# Problem definition

**Input:**

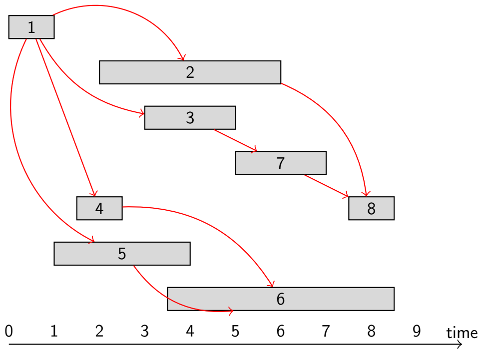
- A set $T = \{1, \ldots, n\}$ of $n$ non-preemptive jobs; each job $i \in T$ has integer processing time $p_i$, an integer release time $r_i$ and an integer deadline $d_i$;



# Problem definition

**Input:**

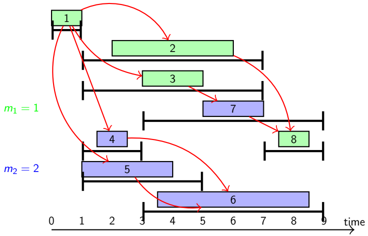
- A set $T = \{1, \ldots, n\}$ of $n$ non-preemptive jobs; each job $i \in T$ has integer processing time $p_i$, an integer release time $r_i$ and an integer deadline $d_i$;
- Precedence graph $G = (T, E)$;



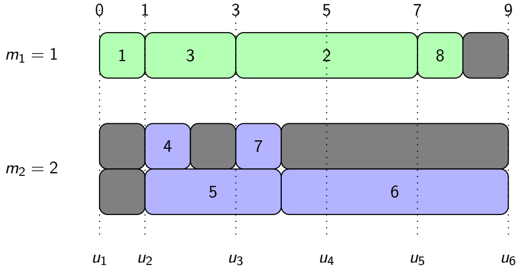
# Problem definition

**Input:**

- A set $T = \{1, \ldots, n\}$ of $n$ non-preemptive jobs; each job $i \in T$ has integer processing time $p_i$, an integer release time $r_i$ and an integer deadline $d_i$;
- Precedence graph $G = (T, E)$;
- Typed tasks: $K$ types machines, $m_k, k \in \{1, \ldots, K\}$ identical machines of type $k$;
- Each job $i \in T$ is processed by a given machine type $\pi_i$.



# Problem definition



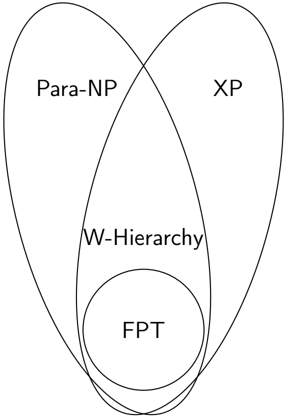

**Objective:** Find, if possible, a feasible schedule

This decision problem is denoted by $P|\mathcal{M}_j(type), prec, r_j, d_j|\star$ using the Graham notation.

# Challenges of Parameterized complexity for scheduling

- Many scheduling problems are NP-complete. But, is it possible to go little bit further in the theoretical study of the complexity of the problem ?
- From a practical point of view, if some instances have parameters bounded by constant values, can we solve the problem in polynomial time ?
- What are the relevant structural parameters for scheduling problems?
- What about the parameterized complexity of basic scheduling problems ?

# Parameterized complexity classes

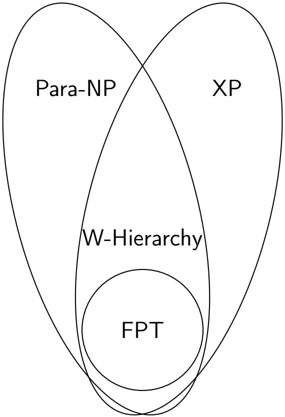



A parameterized problem of size $n$ with parameter $k$ :

**Definition**

FPT is the class of problems solvable by a fixed-parameter tractable algorithm with time complexity $\mathcal{O}(f(k) \times \text{poly}(n))$, where $f$ is a computable function and poly($n$) a polynome of $n$.

# Parameterized complexity classes




A parameterized problem of size $n$ with parameter $k$ :

## Definition

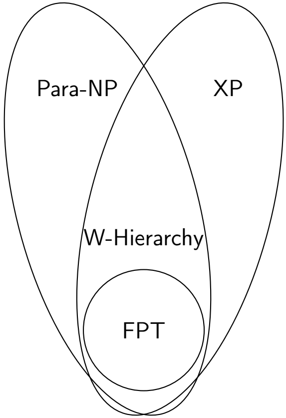
$XP$ is the class of parameterized problems solvable by an algorithm with time complexity $\mathcal{O}(n^{f(k)})$, where $f$ is a computable function.

## Definition

$para - NP$ is the class of parameterized problems solvable by a non-determininstic FPT algorithm

In practice [Flum and Grohe 2006]: A problem with parameter $k$ is para-NP complete if it is NP-complete for one fixed value of $k$

# Parameterized complexity classes



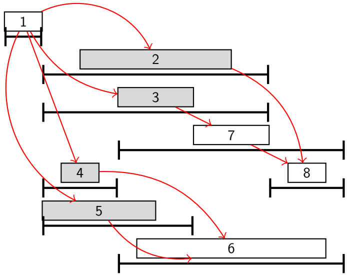

It is conjectured that all the complexity classes are distinct.

# Literature review

**Parameters:**

- $C_{max}$
- $m$
- $p_{max}$
- $\sigma = \max_i(d_i - r_i - p_i)$ or maximal allowed slack w.r.t. earliest schedule
- width of the precedence graph $w(G)$
- nb of different values $d_i$, $p_i$ (Seminar jan 22 by Dvir Shabtay)

$p_{max} = 6,\ m = 3,\ w(G) = 4,\ \sigma = 4$

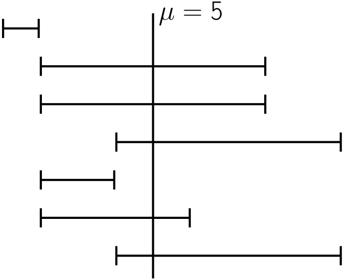



| Parameter(s) | Some results |
|---|---|
| $C_{\max}$ | $P\|prec, p_j = 1\|C_{max}$ is para-NP-complete [Lenstra and Rinnooy Kan 1978] |
| $w(G)$ | $P2\|prec, p_j \in \{1,2\}\|C_{max}$ is $W[2]$-hard [Van Bevern et al. 2016] |
| $w(G)$ | $P\|prec, p_j = 1\|C_{max} \leq D$ is XNLP-complete[Bodlaender et al, 2022] |
| $w(G) + \sigma$ | $PS\|prec\|C_{max}$ if FPT [Van Bevern et al. 2016] |

# Literature review on the pathwidth

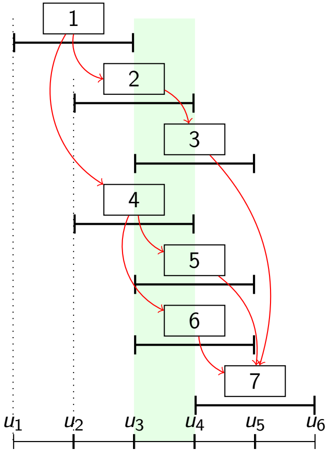
**Parameters:**

- $\mu$ =maximum number of overlaping time windows = pathwidth of the interval graph + 1
- Called pathwidth




| Parameter(s) | Some results |
|---|---|
| $\mu$ | $P\|prec, p_i = 1, r_i, d_i\|\star$ FPT [Munier Kordon 2021] |
| $\mu$ | $P2\|prec, r_i, d_i\|\star$ para NP Complete [Hanen and Munier Kordon 2023] |
| $\mu'$ | $P\|chains(\ell_{i,j}), p_j = 1, r_C, d_C\|\star$ is W[2]-hard [Bodlaender et al 2020] |
| $\mu$ | $P\|chains(\ell_{i,j}), p_j = 1, r_j, d_j\|\star$ is para-NP-Complete [Mallem et al, 2022] |
| $\mu + \ell_{max}$ | $P\|chains(\ell_{i,j}), p_j = 1, r_j, d_j\|\star$ is FPT [Mallem et al, 2022] |
| $\mu + \min(\sigma, p_{max})$ | $P\|\mathcal{M}_j(type), prec, r_j, d_j\|\star$ is FPT [Hanen and Munier Kordon 2023] |

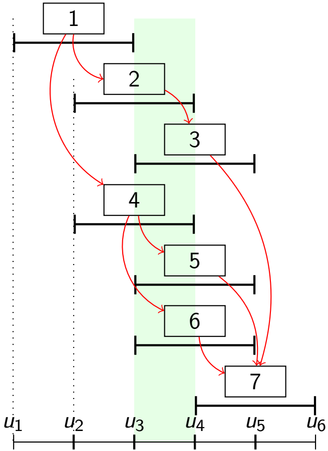
# The unit processing time case $p_i = 1$




| $i \in \mathcal{T}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $r_i$ | 0 | 1 | 2 | 1 | 2 | 2 | 3 |
| $d_i$ | 2 | 3 | 4 | 3 | 4 | 4 | 5 |

1. $u_\alpha$, $\alpha \in \mathbb{N}^\star$ sorted endpoints of $\{[r_i, d_i), i \in \mathcal{T}\}$;
2. $\kappa \leq 2n$ is the number of terms of the sequence $u_\alpha$
3. Here $\kappa = 6$

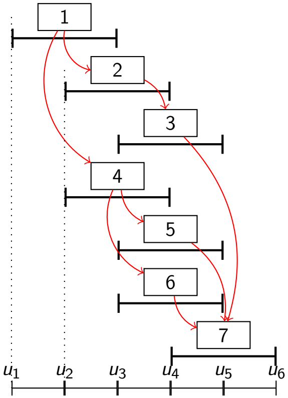
# The unit processing time case $p_i = 1$

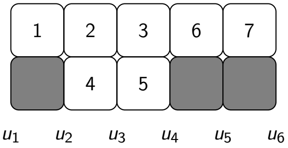


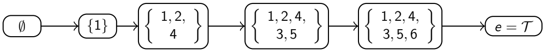

| $i \in \mathcal{T}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $r_i$ | 0 | 1 | 2 | 1 | 2 | 2 | 3 |
| $d_i$ | 2 | 3 | 4 | 3 | 4 | 4 | 5 |

1. $X_\alpha = \{i \in \mathcal{T}, r_i \leq u_\alpha$ and $u_{\alpha+1} \leq d_i\}$ for $\alpha \in \{1, \ldots, \kappa - 1\}$;
2. $X_3 = \{2, 3, 4, 5, 6\}$;
3. parameter $\mu = |X_3| = 5$;
4. $|X_\alpha| \leq \mu$ for all $\alpha$.

# Schedule structure



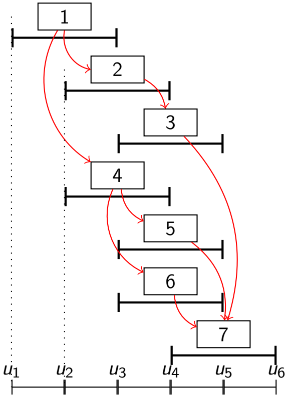



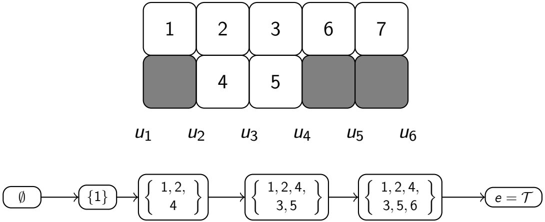




- Path on a state graph.
- A state $v$ of level $\alpha \implies V(v)$ set of jobs completed not later than $u_{\alpha+1}$;
- $V(v)$ comprises :

$$\begin{cases} \text{all jobs } i \text{ with deadline } d_i \leq u_{\alpha+1} = Z_\alpha \\ \text{A set } P(v) \text{ of other jobs} \end{cases}$$

# Schedule structure

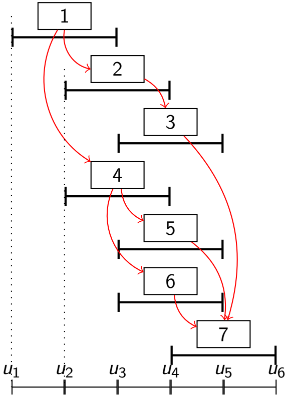


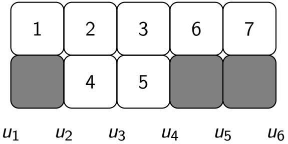



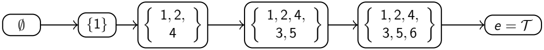

**Theorem ([Munier Kordon 2021])**

*For every state $v \in \mathcal{V}_\alpha$ associated with a feasible partial schedule in $[u_1, u_{\alpha+1})$, $V(v) = Z_\alpha \cup P(v)$ with $P(v) \subseteq X_\alpha$.*

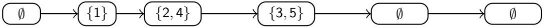
# Schedule structure




A compact representation of the states




$P(v) \subseteq X_\alpha \implies$ : $\mathcal{O}(2^\mu)$ different states of level $\alpha$

# Size of the state graph

**Corollary (Munier-Kordon 2021)**

*For every $\alpha \in \{1, \dots, \kappa\}$, $|\mathcal{V}_\alpha| \leq 2^\mu$. So, the total number of nodes of the state graph $|\mathcal{V}| \leq n \times 2^\mu$. Moreover, the total number of arcs $|\mathcal{A}| \leq n \times 2^{2\mu}$.*

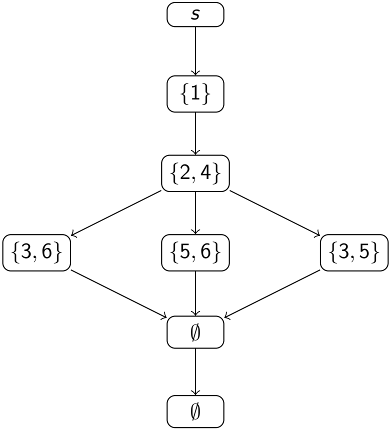
An arc $(u, v)$ of the state graph links a state $u$ of level $\alpha$ to a state $v$ of level $\alpha + 1$.

- consistency of job subsets $P(u) \subseteq P(v) \cup Z_{\alpha+1}$
- consistency of prec constraints
- existence of a feasible schedule of $P(v) \cup Z_{\alpha+1} \backslash P(u) \cup Z_\alpha$ in the interval $[u_\alpha, u_{\alpha+1})$

**Theorem ([Munier Kordon 2021])**

*Checking the existence of an arc $(u, v)$ in the state graph can be done in time complexity $\mathcal{O}(\mu^3 \times 2^{2\mu})$.*

# Full state graph associated to our example




A state graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$

1. Nodes are associated to partial schedules and are represented by less than $\mu$ tasks;
2. Paths from source to sink represent all the feasible schedules.

# Algorithm for building the state graph $\mathcal{G}$

For $\alpha \in \{1, \ldots, \kappa\}$, $\mathcal{V}_\alpha$ is the set of states associated to the feasible schedule in $[u_1, u_\alpha)$;

**Require:** An instance $\mathcal{I}$ of $P|, prec, p_i = 1, r_i, d_i|\star$
**Ensure:** True iff $\mathcal{I}$ is feasible

```
V_1 ← {∅}, G ← (V, A) with A ← ∅ and V ← V_1;
for α ∈ (2, ..., κ) do
    Build the nodes of V_α, V ← V ∪ V_α;
    for all (v, v') ∈ V_{α−1} × V_α do
        if Existence_arc(v, v') then
            A ← A ∪ {(v, v')}
        end if
    end for
end for
return ∃ a path in G from s ∈ V_1 to a node v associated to T.
```

# A FPT Algorithm for $P|prec, p_i = 1, r_i, d_i|\star$

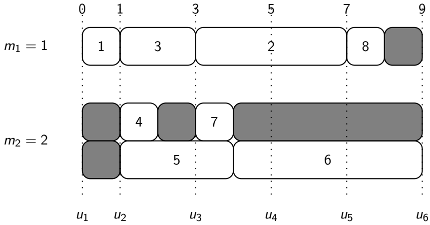
**Theorem ([Munier Kordon 2021])**

*$P|prec, p_i = 1, r_i, d_i|\star$ is fixed-parameter tractable by the pathwidth $\mu$. The time complexity of the FPT-Algorithm is in $\mathcal{O}(n^4 2^{4\mu})$.*

# Extension for general processing times - Intuition

Information to be recorded in a state at level $\alpha$ (i.e. for jobs started before $u_{\alpha+1}$ )is:

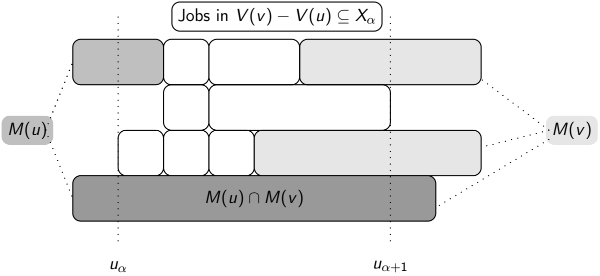
- The set $P(v)$ of scheduled jobs not in $Z_\alpha$
- The exact schedule $M(v)$ of jobs crossing $u_{\alpha+1}$:
- $((2,7), \bullet, (6,9))$ indicates that job 2 completes at 7 on machine 1 and job 6 completes at 9 on machine 3.




| $\alpha$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $P(v)$ | $\emptyset$ | $\{3,5\}$ | $\{2,6,7\}$ | $\{6,7\}$ | $\emptyset$ |
| $M(v)$ | $(\bullet,\bullet,\bullet)$ | $(\bullet,\bullet,(5,4))$ | $((2,7),\bullet,(6,9))$ | $(\bullet,\bullet,(6,9))$ | $(\bullet,\bullet,\bullet)$ |

# Checking an arc $(u, v)$ of the state graph

- Consistency of sets $V(u)$ and $V(v)$;
- Consistency of the schedules $M(u)$ and $M(v)$;
- Existence of a feasible schedule of the other jobs $\subset X_\alpha$.



# Complexity analysis arguments

Information to be recorded in a state $v$ at level $\alpha$ (i.e. for jobs started before $u_{\alpha+1}$ )is:

- The set $P(v)$ of scheduled jobs not in $Z_\alpha$
- The exact schedule $M(v)$ of jobs crossing $u_{\alpha+1}$

- The set $P(v) \subseteq X_\alpha$, so there are $2^\mu$ such subsets.
- The set of jobs crossing $u_{\alpha+1}$ is in $X_\alpha \cap X_{\alpha+1}$.There are at most $2^\mu$ such subsets.
- For each crossing subset there are at most $\min(\sigma, p_{\max})^\mu \times (\mu+1)^\mu$ different schedules (considering that the nb of machines is less than $\mu$)

# FPT for $p_i \in \mathbf{N}$ parameterized by $(\mu, \min(p_{\max}, \sigma))$

**Theorem (Hanen and Munier Kordon 2023)**

*$P|\mathcal{M}_j(type), prec, r_i, d_i|\star$ is FPT for parameters $(\mu, \min(p_{\max}, \sigma))$.*

# Are both parameters necessary to get a FPT algorithm?

- $P|prec, p_j = 1|C_{max} \leq 3$ is NP-complete [Lenstra and Rinnooy Kan 1978] ;
- Here $\sigma = 2, p_{max} = 1$.

Following the definition of the para-NP-completeness and [Flum and Grohe 2006]:

**Corollary**

*The problem scheduling $P|prec, r_i, d_i|\star$ parameterized by $\min(p_{\max}, \sigma)$ is para-NP-complete.*

# Complexity of $P|prec, r_j, d_j|\star$ parameterized by the pathwidth

A reduction from Partition-SC allows us to get the following theorem:

**Theorem**

*The decision problem $P2|r_i, d_i|\star$ with $\mu = 4$ is* NP*-complete.*

**Corollary**

*The scheduling problem $P2|r_i, d_i|\star$ parameterized by the pathwidth is para-*NP*-complete.*

**Corollary**

*The scheduling problem $P|r_i, d_i|\star$ parameterized by the pathwidth and the number of machines is para-*NP*-complete.*

# Complexity of $P|prec, r_j, d_j|\star$ parameterized by the pathwidth

Partition-SC

**Input:** $n = 2p$ positive integer values $a_1, a_2, \ldots, a_n$ such that, for any value $j \in \{1, \ldots, p\}$, $a_{2j-1} < a_{2j}$.

**Question:** is there a subset $A \subset \{1, \ldots, n\}$ such that, for any value $j \in \{1, \ldots, p\}$, exactly one value from $\{2j-1, 2j\}$ is in $A$ and $\sum_{u \in A} a_u = \sum_{u \in \{1, \ldots n\} - A} a_u$?

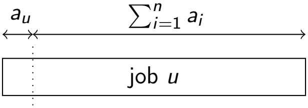
**Lemma**

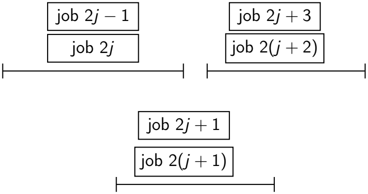
*The decision problem* Partition-SC *is* NP*-complete.*

Claimed in [Garey and Johnson 1979]. Proved using a reduction from Partition.

# From Partition-SC to a scheduling problem

- $n = 2p$ positive integer values $a_1, a_2, \ldots, a_n$ such that, for any value $j \in \{1, \ldots, p\}$, $a_{2j-1} < a_{2j}$.
- Each value $a_u \implies$ job $u$
- job $2j - 1$ and job $2j$ (same interval) cannot be processed on the same machine.
- Intervals of jobs $2j - 1, 2j$ do not intersect with intervals of jobs $2j + 3, 2(j + 2) \implies \mu = 4$



# Two main references for this talk

Main references for this talk:

Munier Kordon 2021 **A fixed-parameter algorithm for scheduling unit dependent tasks on parallel machines with time windows.** Discret. Appl. Math. 290: 1-6 (2021)

Hanen and Munier Kordon 2023 **Fixed-parameter tractability of scheduling dependent typed tasks subject to release times and deadlines.** J Sched (2023).

# Conclusion

- The tuple $(\mu, p_{\max})$ seems to be a good parameter to capture the parallelism of scheduling problems;
- Are there other (more) interesting parameters for these basic scheduling problems ?
- Are there relations between parameters?
- New exact efficient methods?
- Parameterized complexity of scheduling problems is a wide open field.

# Our recent work on parameterized algorithms for scheduling problems


1. Alix Munier Kordon and Ning Tang, **A fixed-parameter algorithm for a unit-execution-time unit-communication-time tasks scheduling problem with a limited number of identical processors.** RAIRO Oper. Res. 56(5): 3777-3788 (2022)
2. Maher Mallem, Claire Hanen, Alix Munier Kordon, **Parameterized Complexity of a Parallel Machine Scheduling Problem.** IPEC 2022: 21:1-21:21
3. Istenc Tarhan, Jacques Carlier, Claire Hanen, Antoine Jouglet, Alix Munier Kordon, **Parameterized Analysis of a Dynamic Programming Algorithm for a Parallel Machine Scheduling Problem.** Euro-Par 2023: 139-153

# Questions?