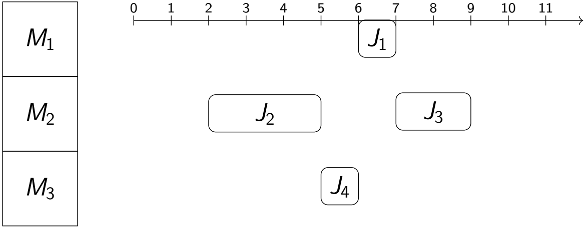
Parameterized complexity of parallel machine scheduling with release dates, deadlines and precedence relations

Maher Mallem, Claire Hanen, Alix Munier-Kordon

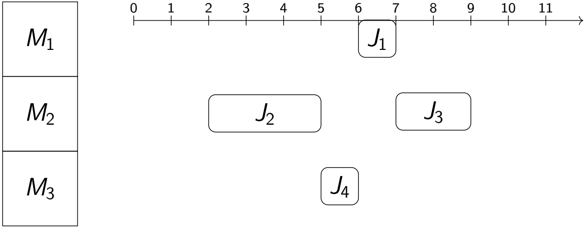
October 31st 2023

SORBONNE UNIVERSITÉ
cnrs
LIP6
Université Paris Nanterre

Parallel machine scheduling

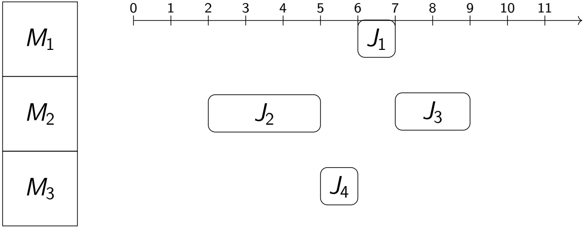


Parallel machine scheduling



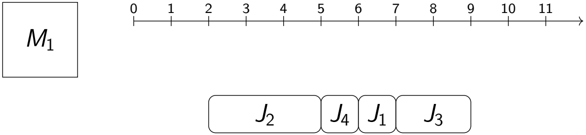
machines | job properties | optimization criterion

Parallel machine scheduling



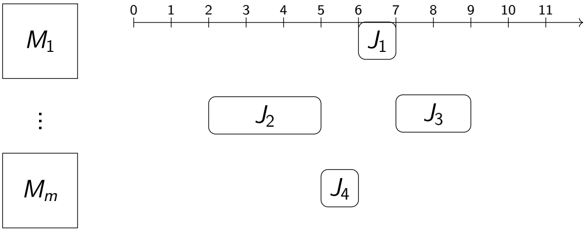
$$P3|.|.$$

Parallel machine scheduling



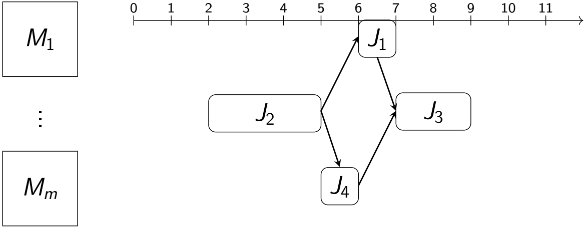
$1|.|.$

Parallel machine scheduling



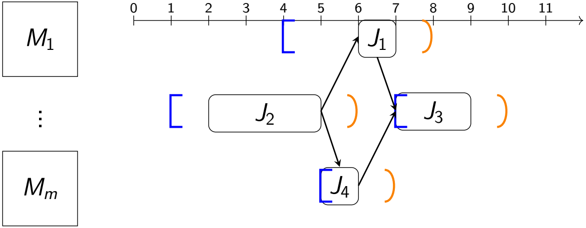
$P|.|.$

Parallel machine scheduling



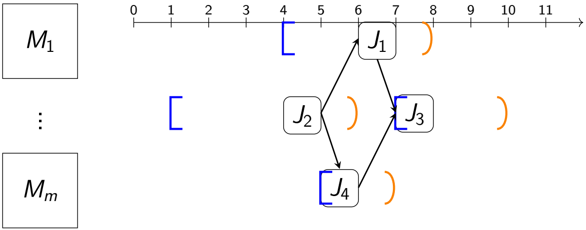
$$P|prec|.$$

Parallel machine scheduling



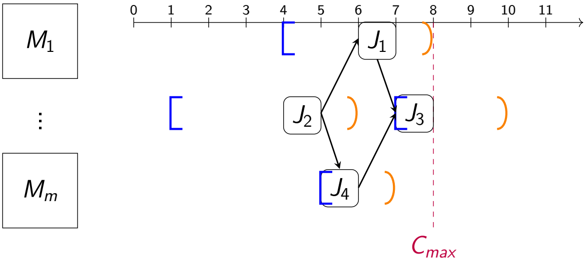
$$P|prec, r_j, d_j|.$$

Parallel machine scheduling

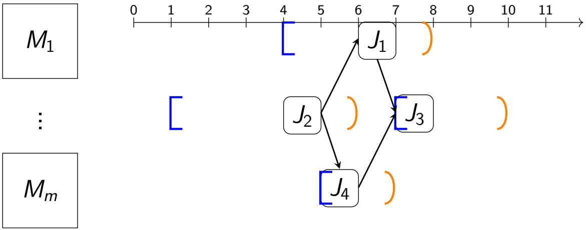


$$P|prec, p_j = p, r_j, d_j|.$$

Parallel machine scheduling



$$P|prec, p_j = p, r_j, d_j|C_{max}$$



$$P|prec, p_j = p, r_j, d_j|\star$$

Typically desired properties

- $1 \longrightarrow P$

Typically desired properties

- $1 \longrightarrow P$
- $p_j = 1 \longrightarrow$ any p_j

Typically desired properties

- $1 \longrightarrow P$
- $p_j = 1 \longrightarrow$ any p_j
- Add precedence relations *prec*

Typically desired properties

- $1 \longrightarrow P$
- $p_j = 1 \longrightarrow$ any p_j
- Add precedence relations *prec*
- Add precedence delays $\ell_{i,j}$

Typically desired properties

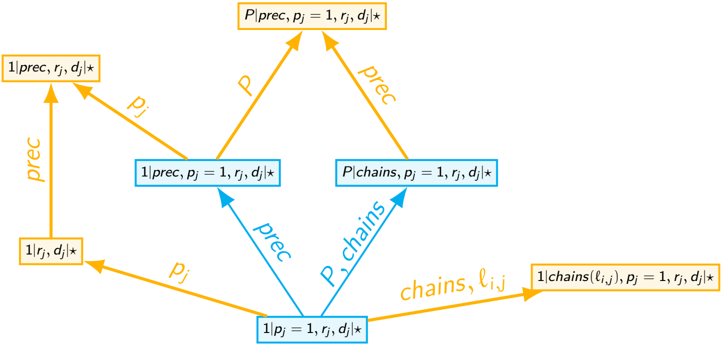
- $1 \longrightarrow P$
- $p_j = 1 \longrightarrow$ any p_j
- Add precedence relations *prec*
- Add precedence delays $\ell_{i,j}$

$1|r_j, d_j|\star$ is strongly NP-hard. [Brucker, Kan, Lenstra 1977]

Classical complexity theory

$\boxed{1|p_j = 1, r_j, d_j|\star}$

Classical complexity theory



Going beyond NP-hardness

- P $\subseteq$ NP
- P: deterministic time $poly(n)$
- NP: nondeterministic time $poly(n)$

Going beyond NP-hardness

- P $\subseteq$ NP
  - FPT $\subseteq$ para-NP, parameter k
- P: deterministic time $poly(n)$
  - FPT: deterministic time $f(k) \times poly(n)$
- NP: nondeterministic time $poly(n)$
  - para-NP: nondeterministic time $f(k) \times poly(n)$

Going beyond NP-hardness

- $\mathrm{P} \subseteq \mathrm{NP}$
  - FPT $\subseteq$ para-NP, parameter k
- P: deterministic time $poly(n)$
  - FPT: deterministic time $f(k) \times poly(n)$
- NP: nondeterministic time $poly(n)$
  - para-NP: nondeterministic time $f(k) \times poly(n)$
- FPT $\subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq$ para-NP.
  - $W[1] - hard$ = "unlikely to be FPT".

Parameterized complexity in machine scheduling

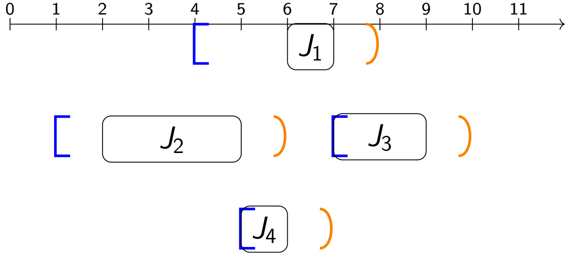
- [Bodlaender, Fellows 1995]
  - #machines m: $P|prec, p_j = 1|C_{max}$ is W[2]-hard.

Parameterized complexity in machine scheduling

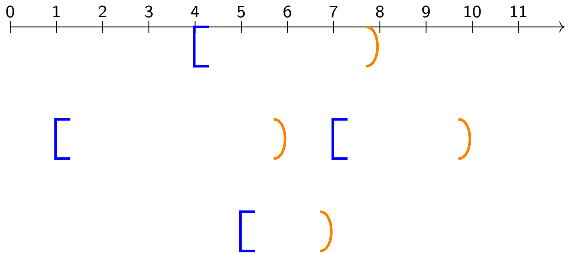
- [Bodlaender, Fellows 1995]
  - #machines m: $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- [Mnich, Wiese 2015]
  - Maximum processing time p_{max}: $P||C_{max} \in$ FPT.

- [Bodlaender, Fellows 1995]
  - #machines m: $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- [Mnich, Wiese 2015]
  - Maximum processing time p_{max}: $P||C_{max} \in$ FPT.
- [Hermelin, Karhi, Pinedo, Shabtay 2018]
  - #distinct d_j, #distinct p_j, #distinct w_j: $1||\sum w_j U_j \in$ FPT with two of the three.

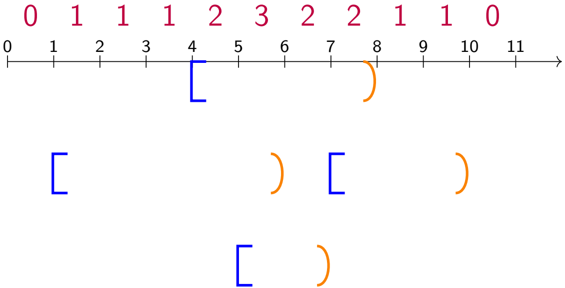
Pathwidth pw



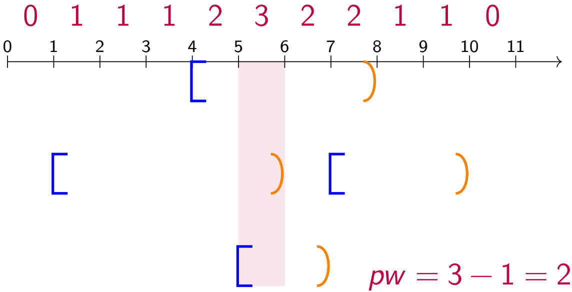
Pathwidth *pw*



Pathwidth pw

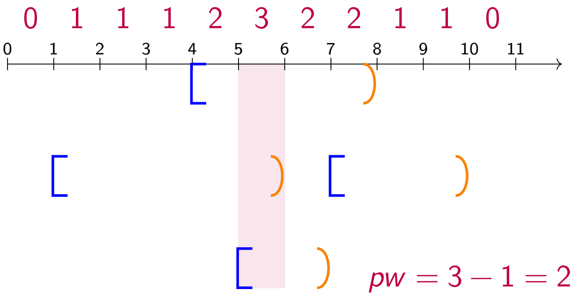


Pathwidth pw



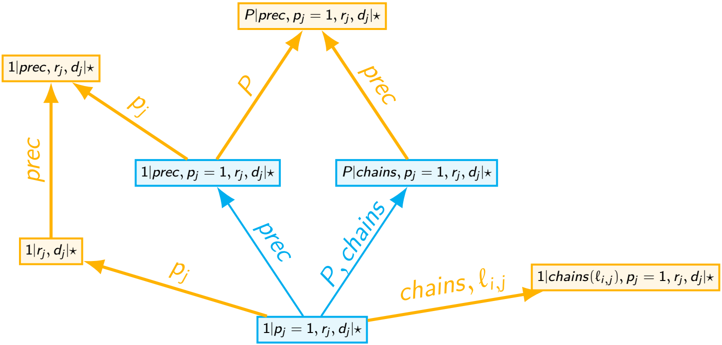
$$pw = 3 - 1 = 2$$

Pathwidth pw

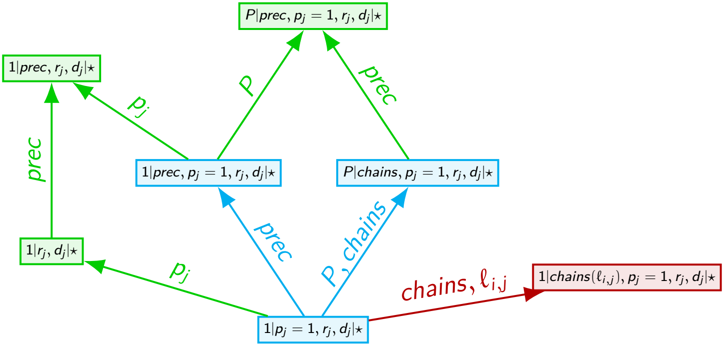


- [Munier 2021] $P|prec, p_j = 1, r_j|L_{max}$ is FPT parameterized by pw.

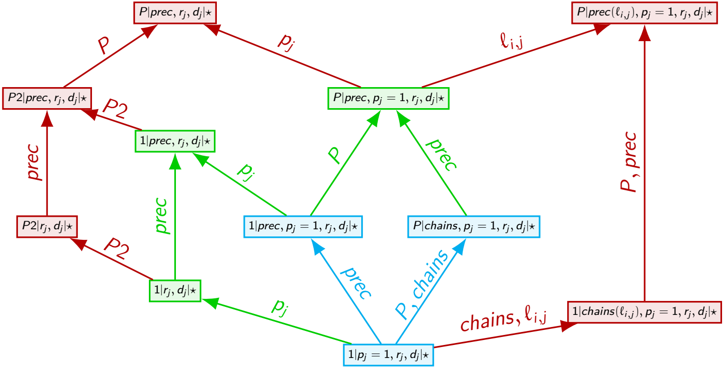
Parameter *pw*



Parameter *pw*

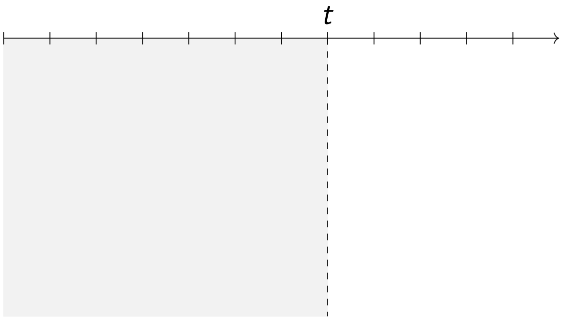


Parameter *pw*



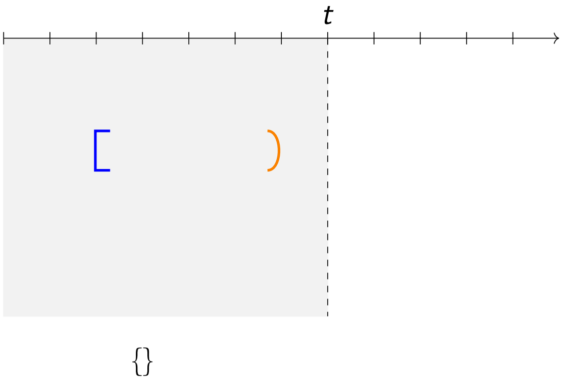
$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules



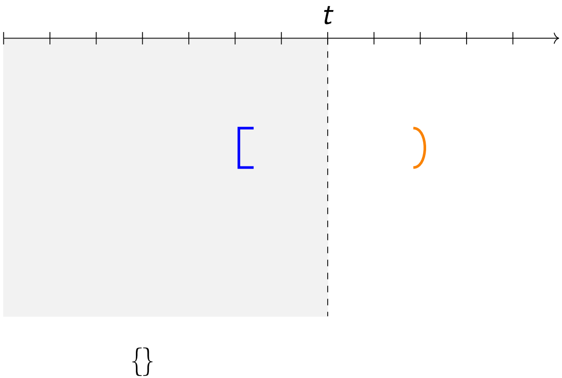
$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules



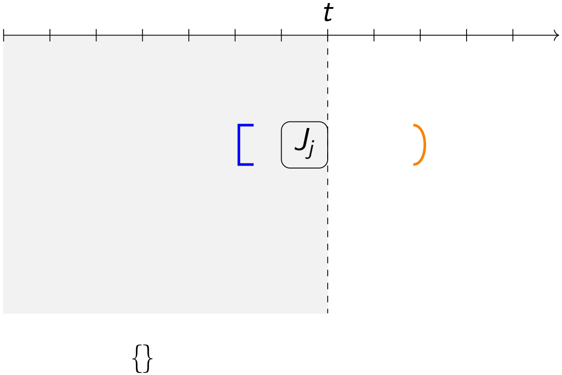
$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules



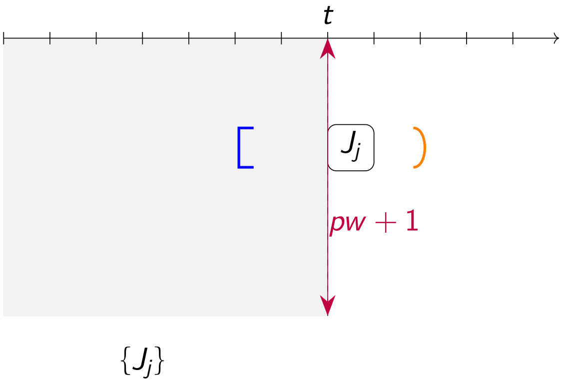
$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules

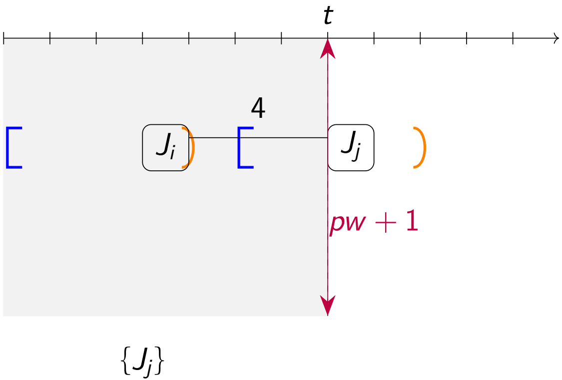


$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules

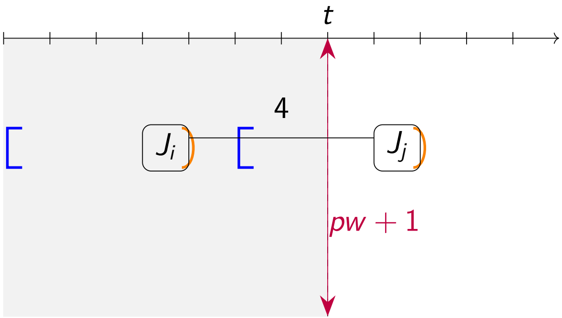


- Dynamic programming algorithm on active schedules



$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

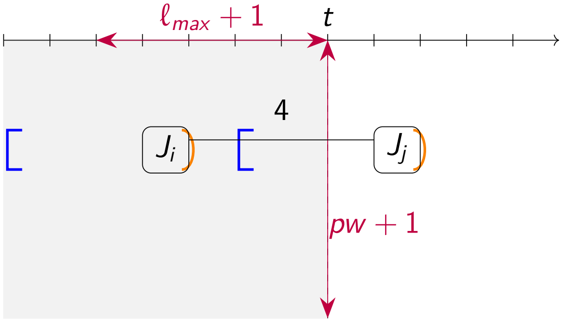
- Dynamic programming algorithm on active schedules



$\{J_j, J_i\}$

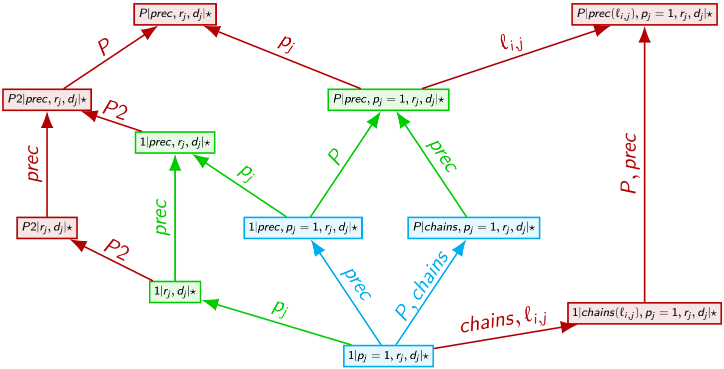
$P|prec(\ell_{i,j}), p_j = 1, r_j, d_j|\star$ is FPT parameterized by $pw + \ell_{max}$

- Dynamic programming algorithm on active schedules

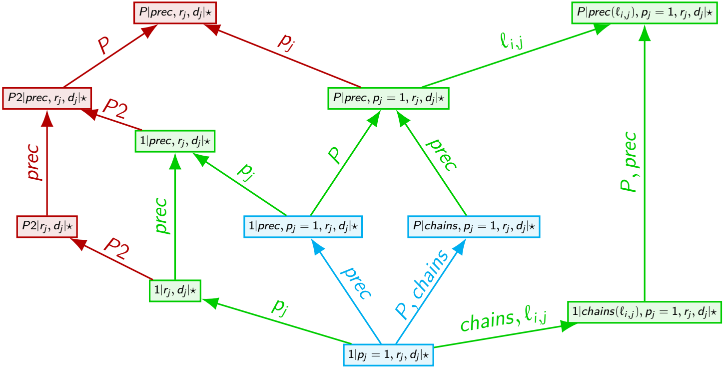


$$|\{J_j, J_i\}| = \mathcal{O}(pw \times \ell_{max})$$

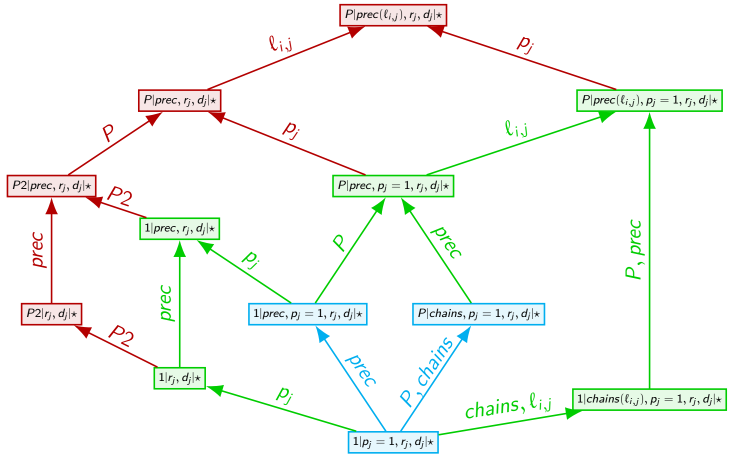
Parameter $pw + \ell_{max}$



Parameter $pw + \ell_{max}$



Parameter $pw + \ell_{max}$



Conclusion

- In-depth classification of scheduling problems (beyond P vs NP).
- More instance families with efficient exact methods.
- Scheduling Zoo [Dürr et al. 2006-Present]