Parameterized Complexity in Scheduling

Maher Mallem[1]

[1]Université Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France

June 8, 2026

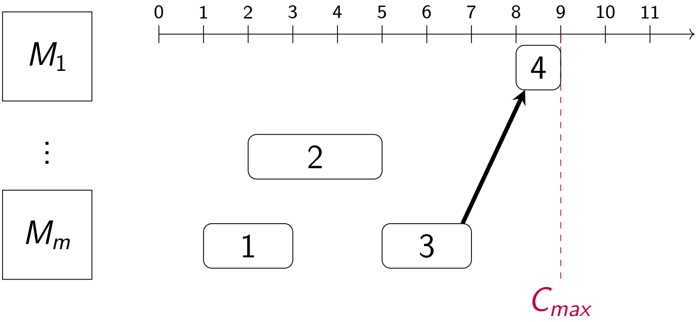
Plan

1. Base concept
   - Motivation
   - Fixed-parameter tractability (FPT)
   - Parameterized hardness
2. Recent developments
   - The XNLP class
   - Kernelization
   - N-fold integer programs
3. Conclusion
4. References

Outline

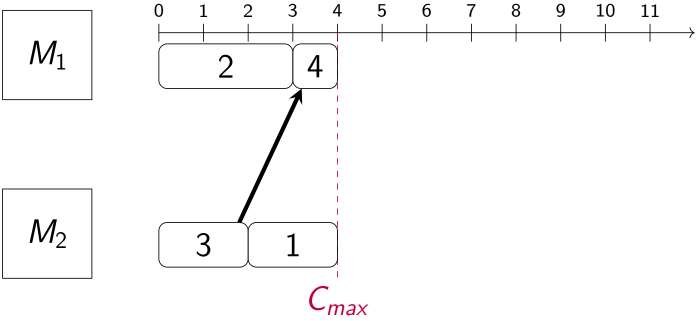
1. Base concept
   - Motivation
   - Fixed-parameter tractability (FPT)
   - Parameterized hardness
2. Recent developments
   - The XNLP class
   - Kernelization
   - N-fold integer programs
3. Conclusion
4. References

How hard is scheduling?



Set of tasks $\mathcal{T} = \{1, \cdots, n\}$.
Number of machines $m \in \mathbb{N}$.
Precedence graph $G = (\mathcal{T}, A)$.

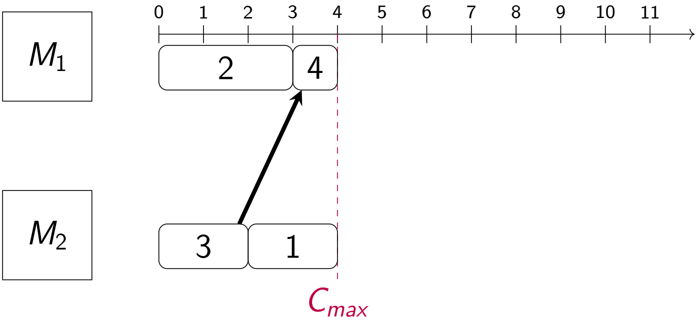
Goal: schedule $\tau : \mathcal{T} \mapsto \mathbb{Z}^+$ minimizing makespan C_{max}.



Set of tasks $\mathcal{T} = \{1, \cdots, n\}$.
Number of machines $m \in \mathbb{N}$.
Precedence graph $G = (\mathcal{T}, A)$.

Goal: schedule $\tau : \mathcal{T} \mapsto \mathbb{Z}^+$ minimizing makespan C_{max}.

How hard is scheduling?



Set of tasks $\mathcal{T} = \{1, \cdots, n\}$.
Number of machines $m \in \mathbb{N}$.
Precedence graph $G = (\mathcal{T}, A)$.

Strongly NP-complete [Garey and Johnson, 1978].

Goal: schedule $\tau : \mathcal{T} \mapsto \mathbb{Z}^+$ minimizing makespan C_{max}.

Strongly NP-complete even with unit-time tasks or empty G.

Outline

Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
  - XP: deterministic time $|\mathcal{I}|^{f(k)}$.

Fixed-parameter algorithms in scheduling

- MIP with a bounded number of integer variables.
  - Initiated by [Mnich and Wiese, 2015].
  - Meta-theorems: MILP [Lenstra, 1983], MICP [Dadush et al., 2011].
  - Bounded number of different task lengths [Hermelin et al., 2021].
- Dynamic Programming
  - Width of precedence graph [van Bevern et al., 2016].
  - Vertex cover number of coupled-task compatibility graph [Bessy and Giroudeau, 2019].
  - Pathwidth of time window intersection graph [Munier Kordon, 2021, Baart et al., 2021].

Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
  - XP: deterministic time $|\mathcal{I}|^{f(k)}$.

Fixed-parameter algorithms in scheduling

- MIP with a bounded number of integer variables.
  - Initiated by [Mnich and Wiese, 2015].
  - Meta-theorems: MILP [Lenstra, 1983], MICP [Dadush et al., 2011].
  - Bounded number of different task lengths [Hermelin et al., 2021].
- Dynamic Programming
  - Width of precedence graph [van Bevern et al., 2016].
  - Vertex cover number of coupled-task compatibility graph [Bessy and Giroudeau, 2019].
  - Pathwidth of time window intersection graph [Munier Kordon, 2021, Baart et al., 2021].

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \rightarrow \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
  - XP: deterministic time $|\mathcal{I}|^{f(k)}$.

Fixed-parameter algorithms in scheduling

- MIP with a bounded number of integer variables.
  - Initiated by [Mnich and Wiese, 2015].
  - Meta-theorems: MILP [Lenstra, 1983], MICP [Dadush et al., 2011].
  - Bounded number of different task lengths [Hermelin et al., 2021].
- Dynamic Programming
  - Width of precedence graph [van Bevern et al., 2016].
  - Vertex cover number of coupled-task compatibility graph [Bessy and Giroudeau, 2019].
  - Pathwidth of time window intersection graph [Munier Kordon, 2021, Baart et al., 2021].

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
  - XP: deterministic time $|\mathcal{I}|^{f(k)}$.

Fixed-parameter algorithms in scheduling

- MIP with a bounded number of integer variables.
  - Initiated by [Mnich and Wiese, 2015].
  - Meta-theorems: MILP [Lenstra, 1983], MICP [Dadush et al., 2011].
  - Bounded number of different task lengths [Hermelin et al., 2021].
- Dynamic Programming
  - Width of precedence graph [van Bevern et al., 2016].
  - Vertex cover number of coupled-task compatibility graph [Bessy and Giroudeau, 2019].
  - Pathwidth of time window intersection graph [Munier Kordon, 2021, Baart et al., 2021].

Outline

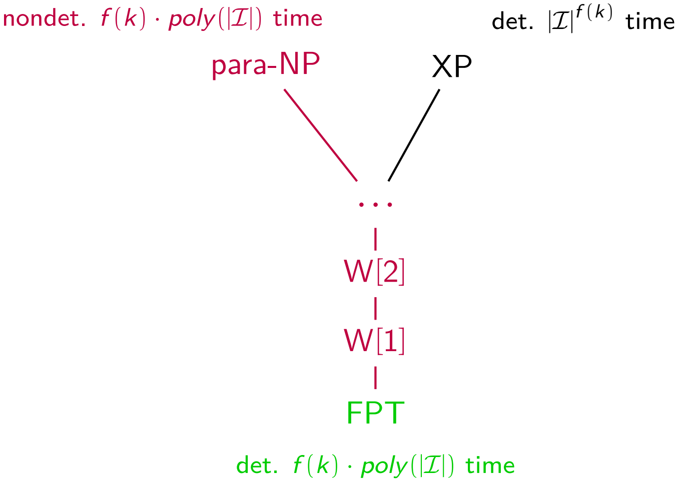
1. Base concept
   - Motivation
   - Fixed-parameter tractability (FPT)
   - Parameterized hardness
2. Recent developments
   - The XNLP class
   - Kernelization
   - N-fold integer programs
3. Conclusion
4. References

Parameterized hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic $f(k) \cdot poly(|\mathcal{I}|)$ time.
- NP: nondeterministic time $poly(|\mathcal{I}|)$.
  - para-NP: nondeterministic $f(k) \cdot poly(|\mathcal{I}|)$ time.
  - para-NP-complete: NP-complete for some fixed k.

Parameterized hardness

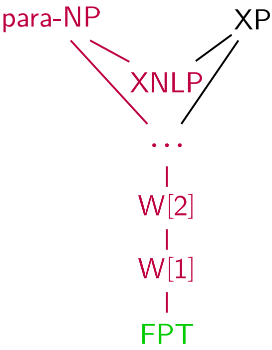
- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic $f(k) \cdot poly(|\mathcal{I}|)$ time.
- NP: nondeterministic time $poly(|\mathcal{I}|)$.
  - para-NP: nondeterministic $f(k) \cdot poly(|\mathcal{I}|)$ time.
  - para-NP-complete: NP-complete for some fixed k.


nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time
det. $|\mathcal{I}|^{f(k)}$ time
para-NP
XP
...
W[2]
W[1]
FPT
det. $f(k) \cdot poly(|\mathcal{I}|)$ time

Outline

1. Base concept
   - Motivation
   - Fixed-parameter tractability (FPT)
   - Parameterized hardness
2. Recent developments
   - The XNLP class
   - Kernelization
   - N-fold integer programs
3. Conclusion
4. References

The XNLP class



The XNLP class

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time

para-NP

det. $|\mathcal{I}|^{f(k)}$ time

XP

XNLP

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time
& $g(k) \cdot \log(|\mathcal{I}|)$ space
[Elberfeld et al., 2015]

$\cdots$

W[2]

W[1]

FPT

det. $f(k) \cdot poly(|\mathcal{I}|)$ time

The XNLP class

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time

$$\text{para-NP} \to \text{XNLP} \to \cdots \to \text{W}[2] \to \text{W}[1]$$

XNLP: nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time & $g(k) \cdot \log(|\mathcal{I}|)$ space [Elberfeld et al., 2015]

The XNLP class

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time

para-NP

$\searrow$

XNLP — nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time & $g(k) \cdot \log(|\mathcal{I}|)$ space [Elberfeld et al., 2015]

$\downarrow$

$\cdots$

$\downarrow$

W[2]

$\downarrow$

W[1]

Recent use of XNLP in scheduling

- Parallel identical machines.
  - $P|prec, p_j = 1|C_{max}$ is XNLP-hard parameterized by the number of machines [Bodlaender et al., 2022].
- Single machine with precedence delays.
  - [Mallem et al., 2026, Bodlaender and Mallem, 2026].

Outline

Kernelization

- Decision problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \rightarrow \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- Kernelization algorithm: outputs $\mathcal{I}'$ in time $poly(|\mathcal{I}|)$ such that:
  - $\mathcal{I}'$ is equivalent to $\mathcal{I}$.
  - $|\mathcal{I}'| \leq f(k)$ for some computable function f.
  - $\bar{k}(\mathcal{I}') \leq g(k)$ for some computable function g.

Reduction rules in scheduling

- Numerical values (time, weight, etc.).
  - Meta-theorem [Frank and Tardos, 1987].
  - Knapsack [Etscheid et al., 2017, Gurski et al., 2019].
- Number of tasks.
  - Single machine with task time windows [Mallem et al., 2024].

Kernelization

- Decision problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k} : \mathcal{P} \to \mathbb{N}$.
  - Instance $\mathcal{I}$ with parameter value $\bar{k}(\mathcal{I}) \leq k$.
- Kernelization algorithm: outputs $\mathcal{I}'$ in time $poly(|\mathcal{I}|)$ such that:
  - $\mathcal{I}'$ is equivalent to $\mathcal{I}$.
  - $|\mathcal{I}'| \leq f(k)$ for some computable function f.
  - $\bar{k}(\mathcal{I}') \leq g(k)$ for some computable function g.

Reduction rules in scheduling

- Numerical values (time, weight, etc.).
  - Meta-theorem [Frank and Tardos, 1987].
  - Knapsack [Etscheid et al., 2017, Gurski et al., 2019].
- Number of tasks.
  - Single machine with task time windows [Mallem et al., 2024].

Outline

N-fold integer programming

$A\mathbf{x} \leq \mathbf{b}$ with A taking the following form:

$$\begin{pmatrix} A_1^1 & A_1^2 & \cdots & A_1^N \\ A_2^1 & 0 & \cdots & 0 \\ 0 & A_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_2^N \end{pmatrix}$$

Highlights

- MoMA 2025 workshop (GT CoA, Graphes, AGaPe, GOThA).
  - Invited lecture: *Integer Programming meets Fixed-Parameter Tractability* by Alexandra Lassota (TU Eindhoven).
- High-multiplicity packing/scheduling [Knop et al., 2019].
  - General framework with a wide range of suitable objective functions.

Outline

Conclusion

Takeaway

- A toolkit to better understand scheduling problems.
- More efficient exact (and approximation!) algorithms.

Going further

- Implementation and experiments.
- Fine-grained complexity.
  - See [Swennenhuis, 2022] and Klaus Jansen's work (Kiel University).
- Scheduling Zoo.
  - https://schedulingzoo.lip6.fr ("advanced" interface).

References I

Baart, R., de Weerdt, M., and He, L. (2021).
Single-machine scheduling with release times, deadlines, setup times, and rejection.
European Journal of Operational Research, 291(2):629–639.
Number: 2 Publisher: Elsevier.

Bessy, S. and Giroudeau, R. (2019).
Parameterized complexity of a coupled-task scheduling problem.
Journal of Scheduling, 22(3):305–313.

Bodlaender, H. L., Groenland, C., Nederlof, J., and Swennenhuis, C. M. (2022).
Parameterized problems complete for nondeterministic FPT time and logarithmic space.
In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 193–204. IEEE.

Bodlaender, H. L. and Mallem, M. (2026).
The parameterized complexity of scheduling with precedence delays: Shuffle product and directed bandwidth.

Dadush, D., Peikert, C., and Vempala, S. (2011).
Enumerative lattice algorithms in any norm via m-ellipsoid coverings.
In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 580–589. IEEE.

References II

Elberfeld, M., Stockhusen, C., and Tantau, T. (2015).
On the space and circuit complexity of parameterized problems: classes and completeness.
Algorithmica, 71:661–701.

Etscheid, M., Kratsch, S., Mnich, M., and Röglin, H. (2017).
Polynomial kernels for weighted problems.
Journal of Computer and System Sciences, 84:1–10.

Frank, A. and Tardos, É. (1987).
An application of simultaneous diophantine approximation in combinatorial optimization.
Combinatorica, 7:49–65.

Garey, M. and Johnson, D. (1978).
Strong NP-completeness results: motivation, examples, and implications.
J. Assoc. Comput. Mach., 25(3):499–508.

Gurski, F., Rehs, C., and Rethmann, J. (2019).
Knapsack problems: a parameterized point of view.
Theoretical Computer Science, 775:93–108.

References III

Hermelin, D., Karhi, S., Pinedo, M., and Shabtay, D. (2021).
New algorithms for minimizing the weighted number of tardy jobs on a single machine.
Annals of Operations Research, 298(1):271–287.
Number: 1 Publisher: Springer.

Knop, D., Koutecký, M., Levin, A., Mnich, M., and Onn, S. (2019).
Multitype integer monoid optimization and applications.
arXiv:1909.07326 [cs, math].
arXiv: 1909.07326.

Lenstra, H. W. (1983).
Integer programming with a fixed number of variables.
Mathematics of Operations Research, 8(4):538–548.

Mallem, M., Hanen, C., and Munier-Kordon, A. (2024).
Kernelization algorithms on single machine scheduling with time windows.
INFORMS Journal on Computing.
Under minor revision.

Mallem, M., Hanen, C., and Munier-Kordon, A. (2026).
Single machine scheduling with precedence constraints and bounded maximum delay value.
Journal of Combinatorial Optimization, 51(3):33.

References IV

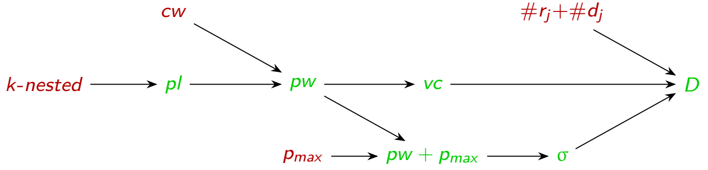
Mnich, M. and Wiese, A. (2015).
Scheduling and fixed-parameter tractability.
Mathematical Programming, 154(1-2):533–562.

Munier Kordon, A. (2021).
A fixed-parameter algorithm for scheduling unit dependent tasks on parallel machines with time windows.
Discrete Applied Mathematics, 290:1–6.

Swennenhuis, C. M. F. (2022).
Fine-grained parameterized complexity of scheduling and sequencing problems.

van Bevern, R., Bredereck, R., Bulteau, L., Komusiewicz, C., Talmon, N., and Woeginger, G. J. (2016).
Precedence-constrained scheduling problems parameterized by partial order width.
In Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., and Pardalos, P., editors, *Discrete Optimization and Operations Research*, pages 105–120, Cham. Springer International Publishing.

Parameter hierarchy on $1|r_j, d_j|C_{max}$



k-nested: depth of the forest inferred by the "strictly included on both ends" relation.

cw: clique-width of the time window interval graph.

p_{max}: maximum processing time.

$\#r_j + \#d_j$: number of distinct release date and deadline values.

pl: proper level.

pw: pathwidth of the time window intersection graph.

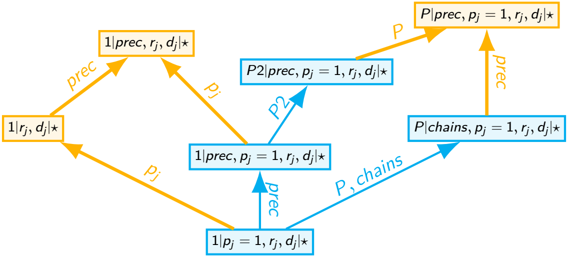
vc: vertex cover number of the time window intersection graph.

σ slack: $\max_j(d_j - r_j - p_j)$.

D: maximum deadline.

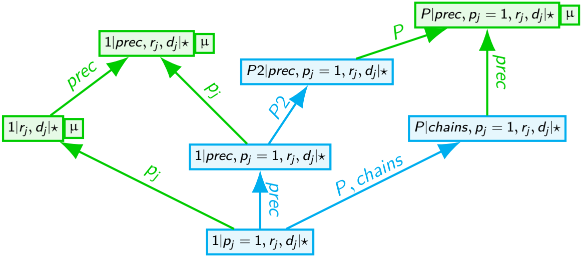
- $a \longrightarrow b$ if and only if, for some computable function f, $a(\mathcal{I}) \leq f(b(\mathcal{I}))$ for every $\mathcal{I} \in \mathcal{P}$.

Parameter μ

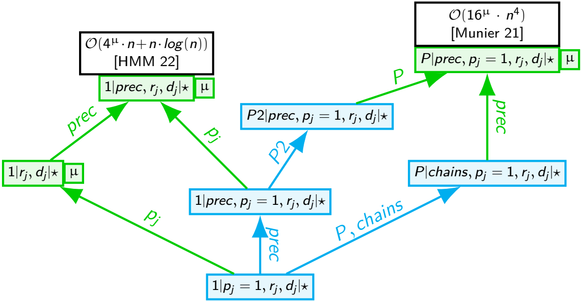


$P|prec, p_j = 1, r_j, d_j|\star$
$1|prec, r_j, d_j|\star$
$P2|prec, p_j = 1, r_j, d_j|\star$
$1|r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
$P|chains, p_j = 1, r_j, d_j|\star$
$1|p_j = 1, r_j, d_j|\star$
prec
p_j
P
P2
prec
p_j
prec
P, chains

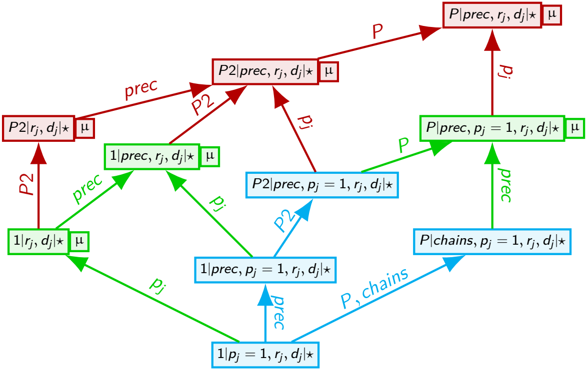
Parameter μ



Parameter μ

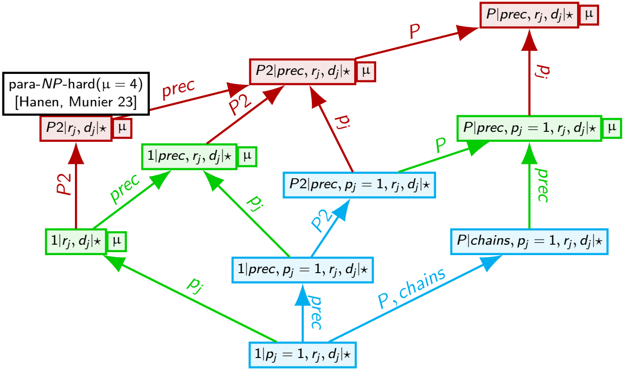

$\mathcal{O}(4^{\mu} \cdot n + n \cdot log(n))$
[HMM 22]
$1|prec, r_j, d_j|\star$
μ
$\mathcal{O}(16^{\mu} \cdot n^4)$
[Munier 21]
$P|prec, p_j = 1, r_j, d_j|\star$
μ
P
$P2|prec, p_j = 1, r_j, d_j|\star$
prec
prec
pj
P2
$1|r_j, d_j|\star$
μ
$P|chains, p_j = 1, r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
pj
prec
P, chains
$1|p_j = 1, r_j, d_j|\star$

Parameter μ



$P|prec, r_j, d_j|\star$ μ
P
$P2|prec, r_j, d_j|\star$ μ
pj
prec
P2
$P2|r_j, d_j|\star$ μ
pj
$P|prec, p_j = 1, r_j, d_j|\star$ μ
P
$1|prec, r_j, d_j|\star$ μ
P2
prec
$P2|prec, p_j = 1, r_j, d_j|\star$
prec
pj
P2
$1|r_j, d_j|\star$ μ
$P|chains, p_j = 1, r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
pj
P, chains
prec
$1|p_j = 1, r_j, d_j|\star$

Parameter μ


$P|prec, r_j, d_j|\star$ μ
P
$P2|prec, r_j, d_j|\star$ μ
pj
para-NP-hard(μ = 4)
[Hanen, Munier 23]
prec
P2
$P2|r_j, d_j|\star$ μ
pj
$P|prec, p_j = 1, r_j, d_j|\star$ μ
$1|prec, r_j, d_j|\star$ μ
P
$P2|prec, p_j = 1, r_j, d_j|\star$
P2
prec
pj
prec
P2
$1|r_j, d_j|\star$ μ
$P|chains, p_j = 1, r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
pj
P , chains
prec
$1|p_j = 1, r_j, d_j|\star$