Introduction to Parameterized Complexity in Scheduling

Maher Mallem

May 4th 2022

FPT vs para-NP

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

FPT vs para-NP

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

Example of an FPT problem

- SATISFIABILITY
  - INPUT: a set S of variables, a boolean formula $\mathcal{F}$ of size n based on S

    QUESTION: $\exists\ \mathcal{I} : S \rightarrow \{TRUE, FALSE\}$ s.t. $\mathcal{F}_{|\mathcal{I}} = TRUE$
  - NP-hard

Example of an FPT problem

- SATISFIABILITY
  - INPUT: a set S of variables, a boolean formula $\mathcal{F}$ of size n based on S

    QUESTION: $\exists\, \mathcal{I} : S \rightarrow \{TRUE, FALSE\}$ s.t. $\mathcal{F}_{|\mathcal{I}} = TRUE$
  - NP-hard
- Parameterized SATISFIABILITY
  - INPUT: a set S of variables, a boolean formula $\mathcal{F}$ of size n based on S
    PARAMETER: the number k of distinct variables used in $\mathcal{F}$
    QUESTION: $\exists\, \mathcal{I} : S \rightarrow \{TRUE, FALSE\}$ s.t. $\mathcal{F}_{|\mathcal{I}} = TRUE$

Example of an FPT problem

- SATISFIABILITY
  - INPUT: a set S of variables, a boolean formula $\mathcal{F}$ of size n based on S

    QUESTION: $\exists\, \mathcal{I} : S \rightarrow \{TRUE, FALSE\}$ s.t. $\mathcal{F}_{|\mathcal{I}} = TRUE$
  - NP-hard
- Parameterized SATISFIABILITY
  - INPUT: a set S of variables, a boolean formula $\mathcal{F}$ of size n based on S
    PARAMETER: the number k of distinct variables used in $\mathcal{F}$
    QUESTION: $\exists\, \mathcal{I} : S \rightarrow \{TRUE, FALSE\}$ s.t. $\mathcal{F}_{|\mathcal{I}} = TRUE$
  - NP-hard but FPT algorithm in time $\mathcal{O}(2^k \times n)$

The W-hierarchy

- P = NP iff $\exists k$, FPT = para-NP [Flum, Grohe 1998]

The W-hierarchy

- P $=$ NP iff $\exists k$, FPT $=$ para-NP [Flum, Grohe 1998]
- W-hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots \subseteq$ para-NP

The W-hierarchy

- P = NP iff $\exists k$, FPT = para-NP [Flum, Grohe 1998]
- W-hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots \subseteq$ para-NP
  - W[1]-complete: INDEPENDENT SET, CLIQUE
  - W[2]-complete: DOMINATING SET

Graham notation

- Three fields: $\alpha|\beta|\gamma$

Graham notation

- Three fields: $\alpha|\beta|\gamma$
- α: machine description
  - 1: single machine
  - P: m parallel machines

Graham notation

- Three fields: $\alpha|\beta|\gamma$

- α: machine description
  - 1: single machine
  - P: *m* parallel machines

- β: job characteristics
  - $p_j = 1$
  - r_j: release dates
  - d_j: deadlines
  - *prec*: general precedence between jobs
  - *chains*: precedence graphs = chains
  - $l_{i,j}$: delay between a job and its successors

Graham notation

- Three fields: $\alpha|\beta|\gamma$
- α: machine description
  - 1: single machine
  - P: m parallel machines
- β: job characteristics
  - $p_j = 1$
  - r_j: release dates
  - d_j: deadlines
  - *prec*: general precedence between jobs
  - *chains*: precedence graphs = chains
  - $l_{i,j}$: delay between a job and its successors
- γ: optimization criterion
  - $\star$: existence
  - C_{max}: earliest completion of the project

Graham notation

- Three fields: $\alpha|\beta|\gamma$
- α: machine description
  - 1: single machine
  - P: m parallel machines
- β: job characteristics
  - $p_j = 1$
  - r_j: release dates
  - d_j: deadlines
  - *prec*: general precedence between jobs
  - *chains*: precedence graphs = chains
  - $l_{i,j}$: delay between a job and its successors
- γ: optimization criterion
  - $\star$: existence
  - C_{max}: earliest completion of the project
- Example: $1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$

A few examples in scheduling

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

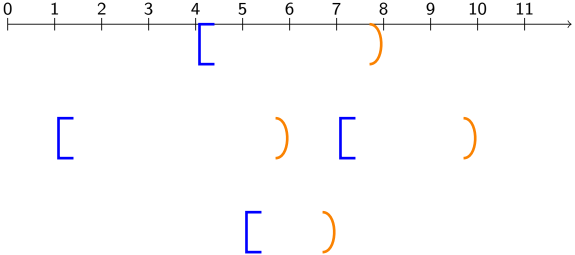
A few examples in scheduling

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

A few examples in scheduling

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

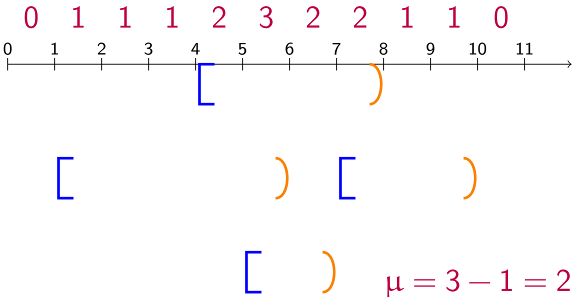
Pathwidth μ



Definition: pathwidth

$$\mu \stackrel{\text{def}}{=} \max_{0 \le t < \max_j d_j} |\{j \mid r_j \le t < d_j\}| - 1$$

Pathwidth μ

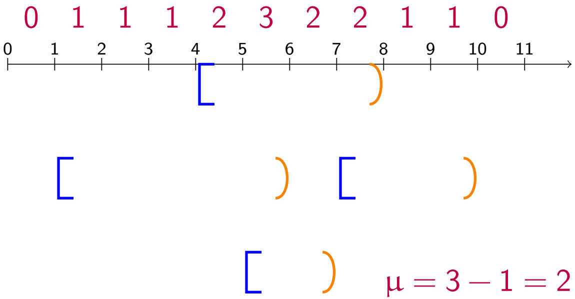


$\mu = 3 - 1 = 2$

Definition: pathwidth

$$\mu \overset{\text{def}}{=} \max_{0 \leq t < \max_j d_j} |\{j \mid r_j \leq t < d_j\}| - 1$$

Pathwidth μ



Definition: pathwidth

$$\mu \overset{\text{def}}{=} \max_{0 \leq t < \max_j d_j} |\{j \mid r_j \leq t < d_j\}| - 1$$

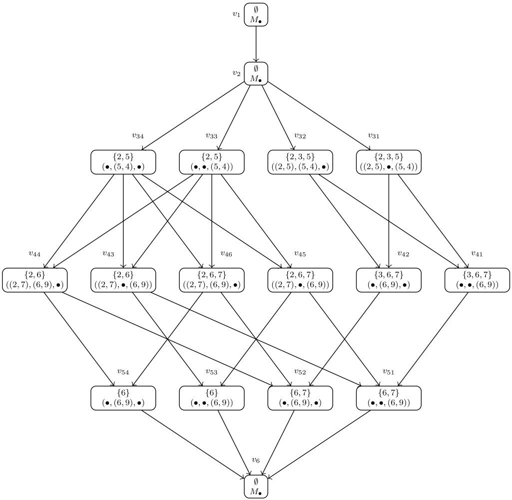
- = Pathwidth of the interval graph induced by the time windows
  - Nodes = time windows
  - there is an edge if they intersect

Pathwidth μ (cont.)

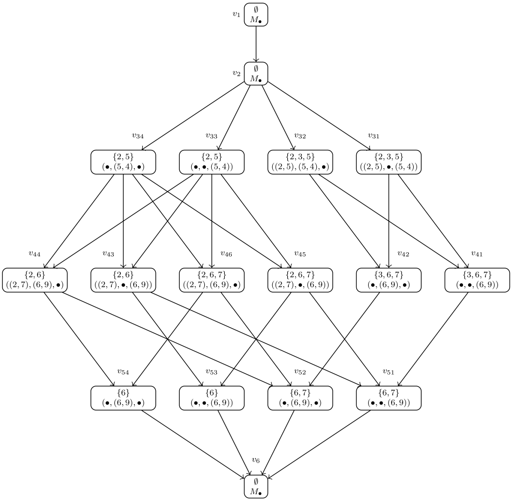
Recent work:

- $1|r_j, s_{jk}, reject, \tilde{d}_j| \sum_{j \notin R} v_j - \sum_{j \notin R} w_j T_j \in$ FPT (μ+slack) [de Weerdt et al. 2020]
- $P|prec, p_j = 1, r_j|L_{max} \in$ FPT (μ) [Munier-Kordon, Tang 2021]
- $P|M_j(type), prec, r_j|L_{max} \in$ FPT ($\mu + p_{max}$) [Hanen, Munier-Kordon 2022]

Pathwidth μ (cont.)



Pathwidth μ (cont.)



- FPT algorithm in time $\mathcal{O}(f(\mu) \times [n^4 + (\mu^{\mu+2} \times !\mu)n])$ with $f(\mu) = [(\mu+1)^{\mu^2} \times 2^\mu]^2$

$1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS

- Decision problem: $1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$
- EXACT DELAYS: $J_i \prec J_j \implies \sigma(j) = \sigma(i) + 1 + l_{i,j}$

$1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS

- Decision problem: $1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$
- EXACT DELAYS: $J_i \prec J_j \implies \sigma(j) = \sigma(i) + 1 + l_{i,j}$
- Goal: prove para-NP-hardness

$1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS

- Decision problem: $1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$
- EXACT DELAYS: $J_i \prec J_j \implies \sigma(j) = \sigma(i) + 1 + l_{i,j}$
- Goal: prove para-NP-hardness
  - prove NP-hardness for a fixed value of the parameter [Flum, Grohe 1998]

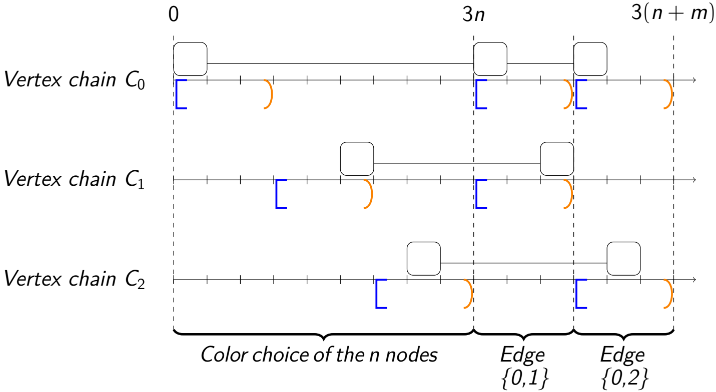
$1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS

- Decision problem: $1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$
- EXACT DELAYS: $J_i \prec J_j \implies \sigma(j) = \sigma(i) + 1 + l_{i,j}$
- Goal: prove para-NP-hardness
  - prove NP-hardness for a fixed value of the parameter [Flum, Grohe 1998]
  - Example: k-COLORING, parameter $k = \#colors$
    - 3-COLORING is NP-hard

$1|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS, pathwidth 1

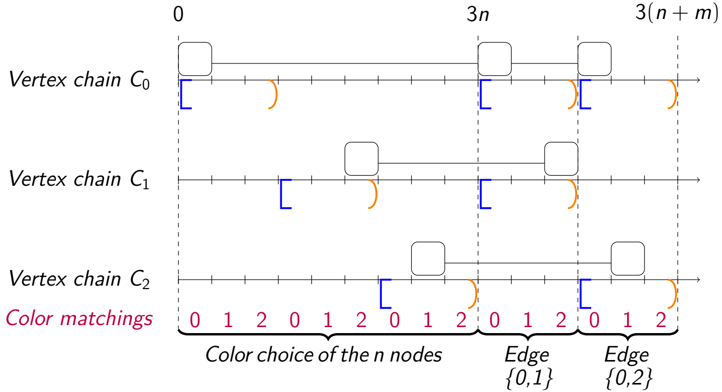
- Reduction from 3-COLORING: $G = (V, E), n = |V|, m = |E|$

Example: $V = \{0, 1, 2\}, E = (\{0, 1\}, \{0, 2\})$



- Reduction from 3-COLORING: $G = (V, E)$, $n = |V|$, $m = |E|$

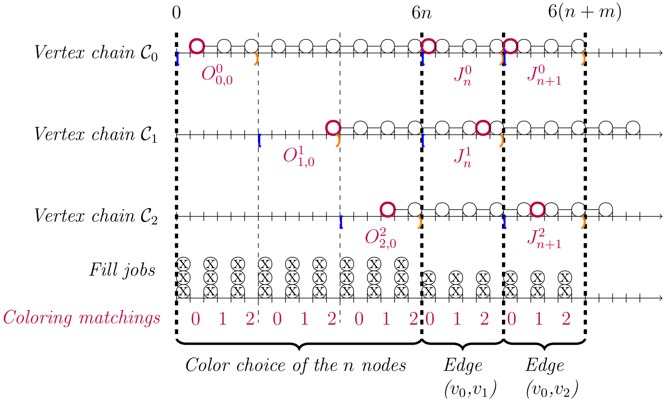
Example: $V = \{0, 1, 2\}$, $E = (\{0, 1\}, \{0, 2\})$



$P|chains(l_{i,j}), p_j = 1, r_j, d_j|\star$ EXACT DELAYS, max delay 1

- Reduction from 3-COLORING: $G = (V, E), n = |V|, m = |E|$

Example: $V = \{0, 1, 2\}, E = (\{0, 1\}, \{0, 2\})$, #machines $= n = 3$



Thank you!