Kernelization Algorithms on Machine Scheduling with Time Windows

Maher Mallem[1], Claire Hanen[2,3], Alix Munier-Kordon[2]

[1]Inria, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342, Lyon cedex 07, France,
[2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France,
[3]Université Paris Nanterre, F-92000 Nanterre, France,

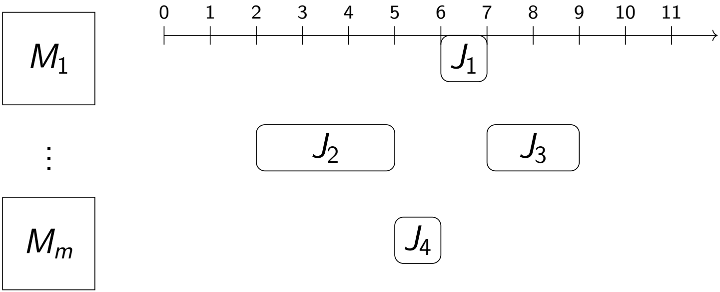
26 November 2024

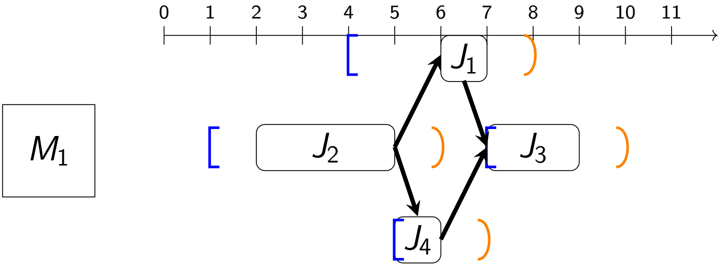
Outline

Outline

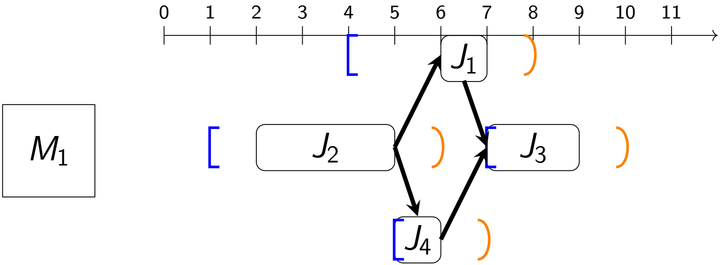
1. Introduction
   - The Problem
   - Parameterized Complexity
2. Polynomial Kernel with Vertex Cover
   - Overview
   - Deleting Jobs
   - Reducing Time Values
3. Kernels with Pathwidth
   - No Polynomial Kernel
   - Exponential Size Kernels
4. Conclusion

Machine Scheduling





$$1|prec, r_j, d_j|\star$$



$1|prec, r_j, d_j|\star$ is strongly NP-hard [LRKB 77].

Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.

- P: deterministic time $poly(|\mathcal{I}|)$.

Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Problem $(\mathcal{P}, k)$, instance $\mathcal{I}$ with parameter value $\leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.

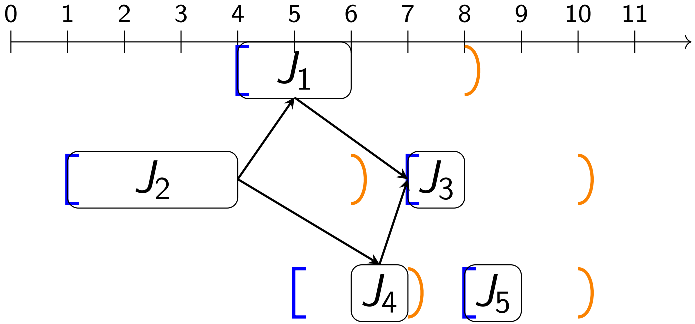
Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
    - Problem $(\mathcal{P}, k)$, instance $\mathcal{I}$ with parameter value $\leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
    - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
- Kernelization algorithm: outputs (I', k') in time $poly(|\mathcal{I}|)$ such that:
    - I' is equivalent to I.
    - $|I'| + k' \leq f(k)$ for some computable function f.

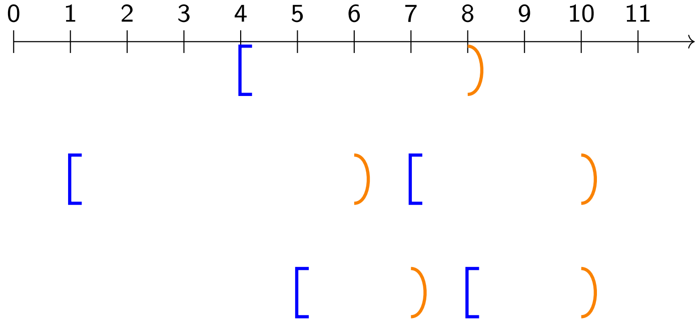
Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Problem $(\mathcal{P}, k)$, instance $\mathcal{I}$ with parameter value $\leq k$.
- P: deterministic time $poly(|\mathcal{I}|)$.
  - FPT: deterministic time $f(k) \cdot poly(|\mathcal{I}|)$.
- Kernelization algorithm: outputs (I', k') in time $poly(|\mathcal{I}|)$ such that:
  - I' is equivalent to I.
  - $|I'| + k' \leq f(k)$ for some computable function f.
- Polynomial kernel: f is polynomial.

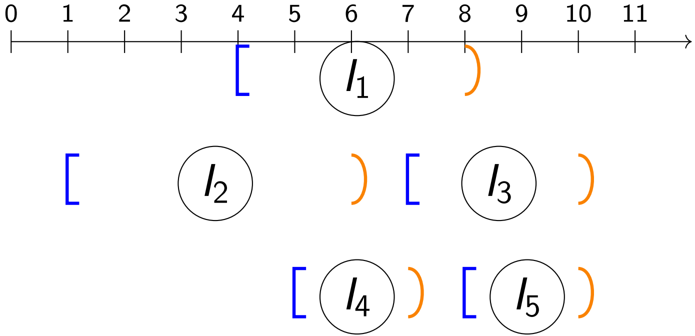
Pathwidth *pw* and vertex cover *vc*



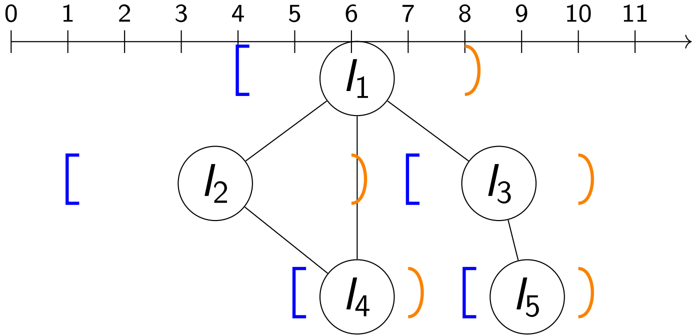
Pathwidth *pw* and vertex cover *vc*



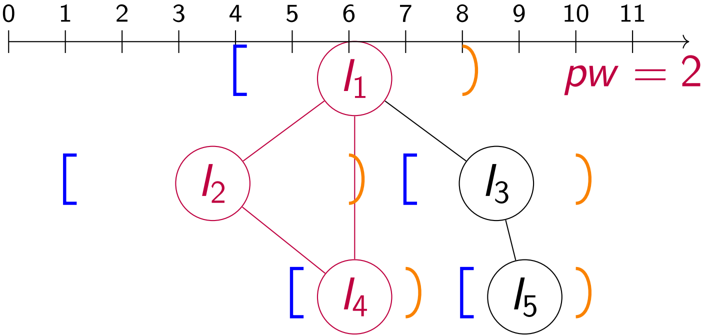
Pathwidth pw and vertex cover vc



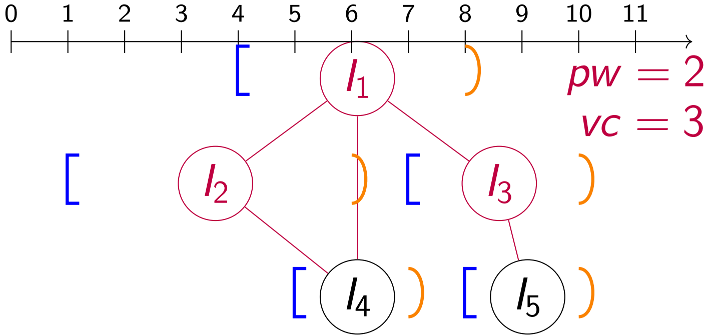
Pathwidth pw and vertex cover vc



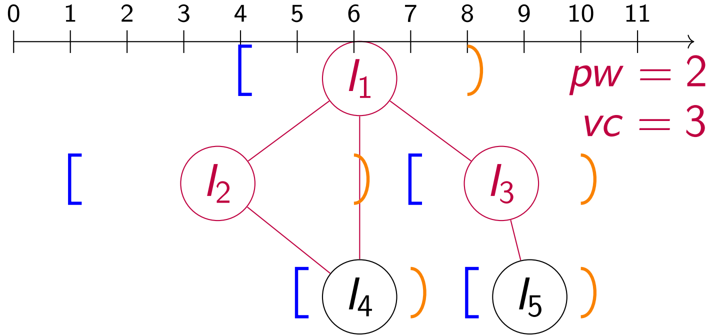
Pathwidth pw and vertex cover vc



Pathwidth pw and vertex cover vc



Pathwidth pw and vertex cover vc



Result

- $pw \leq vc$
- [Mallem, Hanen, Munier-Kordon 22]
  $1|prec, r_j, d_j|\star$ in time $\mathcal{O}((pw+1) \cdot 4^{pw} \cdot n + n \cdot log(n))$.

Outline

Polynomial kernel overview

Proposition

$1|prec, r_j, d_j|\star$ has a polynomial kernel w.r.t. parameter vc.

Polynomial kernel overview

Proposition

$1|prec, r_j, d_j|\star$ has a polynomial kernel w.r.t. parameter vc.

Instance I		Instance I'		Instance I''
n jobs	$\xrightarrow{(1)}$	$\mathcal{O}(vc^2)$ jobs	$\xrightarrow{(2)}$	$\mathcal{O}(vc^2)$ jobs
size $\mathcal{O}(n \cdot log(max_j(d_j)))$		size $\mathcal{O}(vc^2 \cdot log(max_j(d_j)))$		size $\mathcal{O}(vc^8)$

Polynomial kernel overview

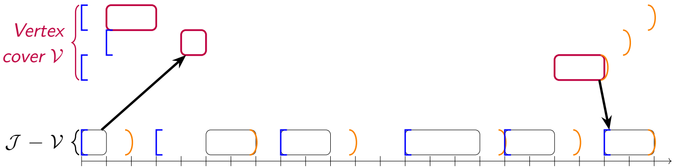
Proposition

$1|prec, r_j, d_j|\star$ has a polynomial kernel w.r.t. parameter vc.

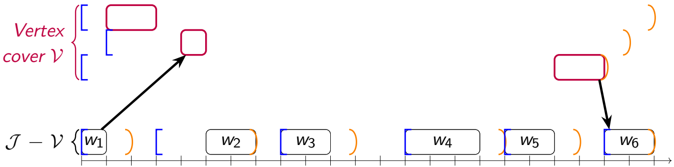
Instance I		Instance I'		Instance I''
n jobs	$\xrightarrow{(1)}$	$\mathcal{O}(vc^2)$ jobs	$\xrightarrow{(2)}$	$\mathcal{O}(vc^2)$ jobs
size $\mathcal{O}(n \cdot log(max_j(d_j)))$		size $\mathcal{O}(vc^2 \cdot log(max_j(d_j)))$		size $\mathcal{O}(vc^8)$

- (1) Delete *most* jobs.
  - Use a reduction rule.
  - Time $\mathcal{O}(n^2)$.
- (2) Reduce time values.
  - Translate into a BIP and use a theorem from [Frank, Tardos 87].
  - Time $\mathcal{O}(n^{18})$.

(1) Viewing non-vertex covering jobs as gaps

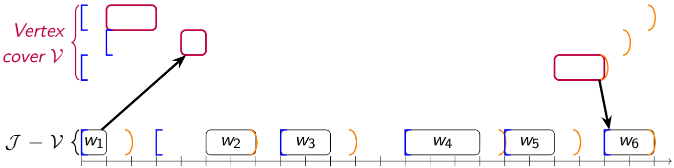


(1) Viewing non-vertex covering jobs as gaps



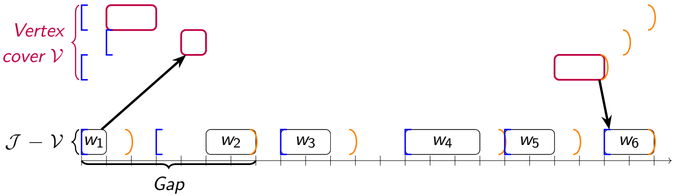
- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.

(1) Viewing non-vertex covering jobs as gaps



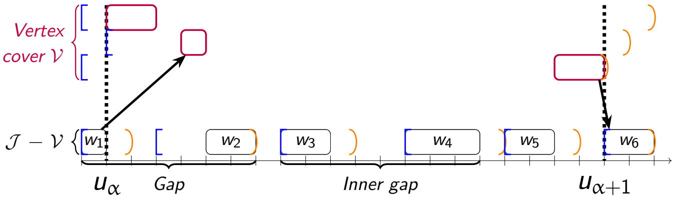
- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.
- Gap $=$ two consecutive jobs w_i, w_{i+1}.

(1) Viewing non-vertex covering jobs as gaps



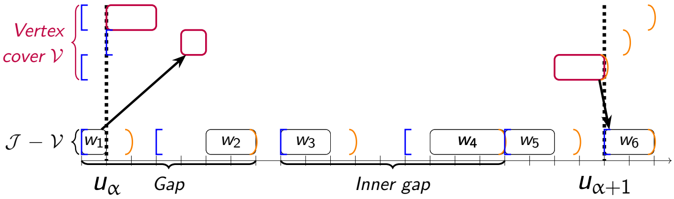
- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.
- Gap $=$ two consecutive jobs w_i, w_{i+1}.

(1) Viewing non-vertex covering jobs as gaps



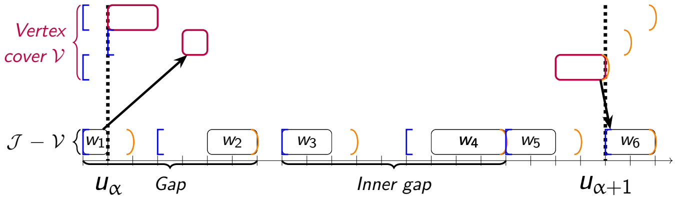
- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.
- Gap $=$ two consecutive jobs w_i, w_{i+1}.
  - Inner gap: time windows fully included in $[u_\alpha, u_{\alpha+1})$.

(1) Viewing non-vertex covering jobs as gaps



- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.
- Gap $=$ two consecutive jobs w_i, w_{i+1}.
  - Inner gap: time windows fully included in $[u_\alpha, u_{\alpha+1})$.
- Maximum capacity: w_i at the earliest, w_{i+1} at the latest.

(1) Viewing non-vertex covering jobs as gaps

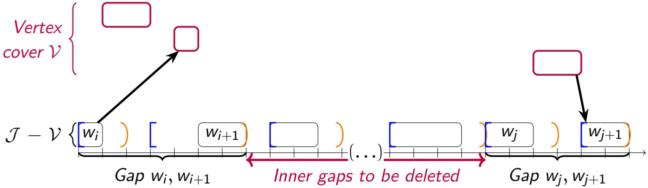


- $\mathcal{J} - \mathcal{V}$: jobs $w_1, w_2, \ldots, w_{|\mathcal{J}-\mathcal{V}|}$.
- Gap = two consecutive jobs w_i, w_{i+1}.
  - Inner gap: time windows fully included in $[u_\alpha, u_{\alpha+1})$.
- Maximum capacity: w_i at the earliest, w_{i+1} at the latest.
- $vc_\alpha = |\{J_j \in \mathcal{V}, [u_\alpha, u_{\alpha+1}) \subseteq [r_j, d_j)\}|$.

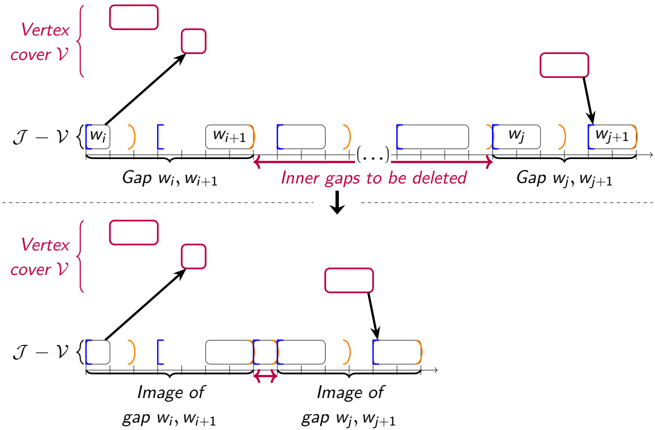
Reduction rule

In each interval $[u_\alpha, u_{\alpha+1})$, only keep the $3vc_\alpha$ inner gaps with highest maximum capacity.

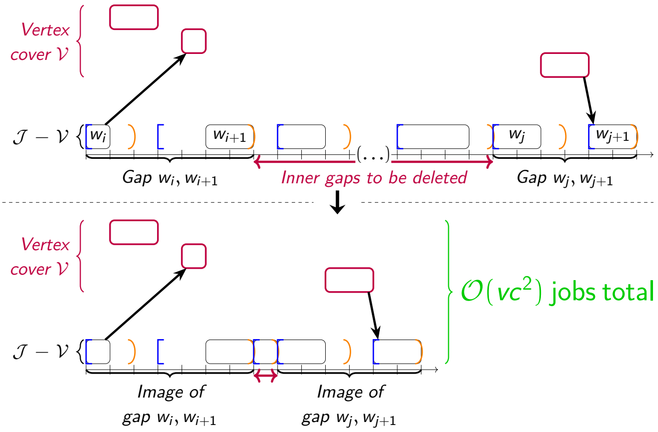
(1) Deleting unnecessary inner gaps



(1) Deleting unnecessary inner gaps



(1) Deleting unnecessary inner gaps



(2) Reducing time values

(2) Reducing time values

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w'_1, \ldots, w'_d)$ of positive integers such that:

- $w'_i \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w'_i x_i \geq 0.$$

(2) Reducing time values

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w'_1, \ldots, w'_d)$ of positive integers such that:

- $w'_i \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w'_i x_i \geq 0.$$

- [Lasserre, Queyranne 92] Binary IP equivalent to $1|prec, r_j, d_j|\star$.
  - At most $n + 2$ variables equal to one in each equation of any solution.

(2) Reducing time values

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w_1', \ldots, w_d')$ of positive integers such that:

- $w_i' \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w_i' x_i \geq 0.$$

- [Lasserre, Queyranne 92] Binary IP equivalent to $1|prec, r_j, d_j|\star$.
  - At most $n + 2$ variables equal to one in each equation of any solution.
- Use the theorem on instance I' with $\tilde{n} = \mathcal{O}(vc^2)$ jobs.
  - $d = 3\tilde{n}$, $\mathtt{w} = (p_1, r_1, d_1, \ldots, p_{\tilde{n}}, r_{\tilde{n}}, d_{\tilde{n}})$, $N = \tilde{n} + 3$.

(2) Reducing time values

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \dots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w'_1, \dots, w'_d)$ of positive integers such that:

- $w'_i \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \dots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w'_i x_i \geq 0.$$

- [Lasserre, Queyranne 92] Binary IP equivalent to $1|prec, r_j, d_j|\star$.
  - At most $n + 2$ variables equal to one in each equation of any solution.
- Use the theorem on instance I' with $\tilde{n} = \mathcal{O}(vc^2)$ jobs.
  - $d = 3\tilde{n}$, $\mathtt{w} = (p_1, r_1, d_1, \dots, p_{\tilde{n}}, r_{\tilde{n}}, d_{\tilde{n}})$, $N = \tilde{n} + 3$.
  - Time $\mathcal{O}(vc^{18}) = \mathcal{O}(n^{18})$.
  - Each reduced time value taking space $\mathcal{O}(vc^6)$.

Outline

1. Introduction
   - The Problem
   - Parameterized Complexity
2. Polynomial Kernel with Vertex Cover
   - Overview
   - Deleting Jobs
   - Reducing Time Values
3. Kernels with Pathwidth
   - No Polynomial Kernel
   - Exponential Size Kernels
4. Conclusion

No polynomial kernel w.r.t. pathwidth pw

Proposition

Unless $NP \subseteq coNP/poly$ there is no polynomial kernel of $1|prec, r_j, d_j|\star$ w.r.t. pathwidth pw.

Proposition

Unless $NP \subseteq coNP/poly$ there is no polynomial kernel of $1|prec, r_j, d_j|\star$ w.r.t. pathwidth pw.

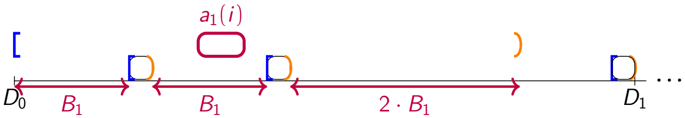
- *Proof:* Cross-composition from PARTITION.
  - $(P_1, \ldots, P_t) \mapsto I$
  - $pw \leq poly(max_i(|P_i|) + log(t))$

No polynomial kernel w.r.t. pathwidth pw

Proposition

Unless $NP \subseteq coNP/poly$ there is no polynomial kernel of $1|prec, r_j, d_j|\star$ w.r.t. pathwidth pw.

- *Proof:* Cross-composition from PARTITION.
  - $(P_1, \ldots, P_t) \mapsto I$
  - $pw \leq poly(max_i(|P_i|) + log(t))$

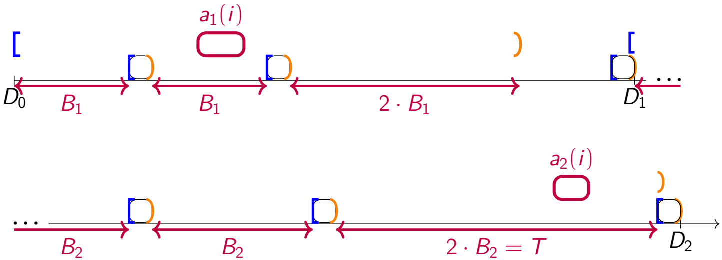


No polynomial kernel w.r.t. pathwidth pw

Proposition

Unless $NP \subseteq coNP/poly$ there is no polynomial kernel of $1|prec, r_j, d_j|\star$ w.r.t. pathwidth pw.

- *Proof:* Cross-composition from PARTITION.
  - $(P_1, \ldots, P_t) \mapsto I$
  - $pw \leq poly(max_i(|P_i|) + log(t))$

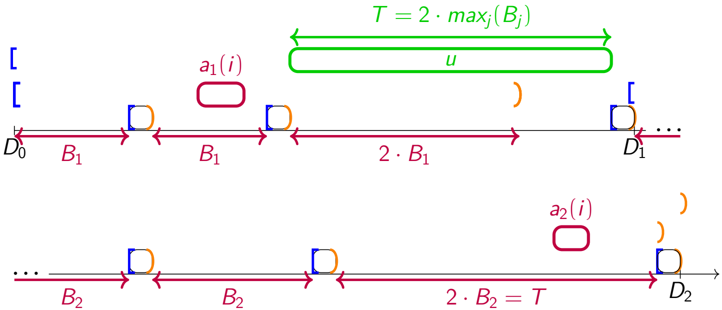


No polynomial kernel w.r.t. pathwidth *pw*

Proposition

Unless $NP \subseteq coNP/poly$ there is no polynomial kernel of $1|prec, r_j, d_j|\star$ w.r.t. pathwidth pw.

- *Proof:* Cross-composition from PARTITION.
  - $(P_1, \dots, P_t) \mapsto I$
  - $pw \leq poly(max_i(|P_i|) + log(t))$



Algorithm 1 NaiveKernel($1|prec, r_j, d_j|\star$) with pathwidth pw

Input: an instance I of $1|prec, r_j, \bar{d}_j|\star$.
Compute pw in time $\mathcal{O}(n \cdot log(n))$.
if $|I| \leq 2^{pw}$ **then**
  return I
else
  Use the *FPT* algorithm from [MHMK 22] in time $\mathcal{O}((pw+1)^2 \cdot 4^{pw} \cdot n + n \cdot log(n)) = \mathcal{O}(|I|^2 \cdot log(|I|) \cdot n)$.
  if the output is YES **then**
    return a dummy YES instance of size $\mathcal{O}(1)$.
  else
    return a dummy NO instance of size $\mathcal{O}(1)$.

An exponential size kernel w.r.t. pathwidth pw

Algorithm 1 NaiveKernel($1|prec, r_j, d_j|\star$) with pathwidth pw

Input: an instance I of $1|prec, r_j, \bar{d}_j|\star$.
Compute pw in time $\mathcal{O}(n \cdot log(n))$.
if $|I| \leq 2^{pw}$ **then**
    return I
else
    Use the *FPT* algorithm from [MHMK 22] in time $\mathcal{O}((pw+1)^2 \cdot 4^{pw} \cdot n + n \cdot log(n)) = \mathcal{O}(|I|^2 \cdot log(|I|) \cdot n)$.
    if the output is YES **then**
        return a dummy YES instance of size $\mathcal{O}(1)$.
    else
        return a dummy NO instance of size $\mathcal{O}(1)$.

An exponential size kernel w.r.t. pathwidth pw (cont.)

Algorithm 2 Kernel($1|prec, r_j, d_j|\star$) with pathwidth pw

```
Input: an instance I of 1|prec, r_j, \bar{d}_j|⋆.
Compute pw and vc in time O(n · log(n)).
if vc ≤ 2^pw then
    return the polynomial kernel of I w.r.t. vc.
else
    Use the FPT algorithm from [MHMK 22] in time
    O((pw + 1)^2 · 4^pw · n + n · log(n)) = O(vc^2 · log(vc) · n).
    if the output is YES then
        return a dummy YES instance of size O(1).
    else
        return a dummy NO instance of size O(1).
```

Outline

1. Introduction
   - The Problem
   - Parameterized Complexity
2. Polynomial Kernel with Vertex Cover
   - Overview
   - Deleting Jobs
   - Reducing Time Values
3. Kernels with Pathwidth
   - No Polynomial Kernel
   - Exponential Size Kernels
4. Conclusion

Conclusion

Parameter	$1 \| prec, r_j, d_j \| \star$	
pw	FPT [Mallem, Hanen, Munier-Kordon 22] Kernel of exponential size No polynomial kernel	
vc	FPT Polynomial kernel	

Conclusion

Parameter	$1\|prec, r_j, d_j\|\star$	$P2\|r_j, d_j\|\star$
pw	FPT [Mallem, Hanen, Munier-Kordon 22] Kernel of exponential size No polynomial kernel	NP-hard when $pw = 3$ [Hanen, Munier-Kordon 23]
vc	FPT Polynomial kernel	Open

Future work

- Investigate other scheduling problems with time windows.

Parameter	$1\|prec, r_j, d_j\|\star$	$P2\|r_j, d_j\|\star$
pw	FPT [Mallem, Hanen, Munier-Kordon 22] Kernel of exponential size No polynomial kernel	NP-hard when $pw = 3$ [Hanen, Munier-Kordon 23]
vc	FPT Polynomial kernel	Open

Future work

- Investigate other scheduling problems with time windows.
- Consider parameters between pathwidth and vertex cover.
  - $pw \longrightarrow$ treedepth $\longrightarrow$ vertex integrity $\longrightarrow vc$

Conclusion

Parameter	$1\|prec, r_j, d_j\|\star$	$P2\|r_j, d_j\|\star$
pw	FPT [Mallem, Hanen, Munier-Kordon 22] Kernel of exponential size No polynomial kernel	NP-hard when $pw = 3$ [Hanen, Munier-Kordon 23]
vc	FPT Polynomial kernel	Open

Future work

- Investigate other scheduling problems with time windows.
- Consider parameters between pathwidth and vertex cover.
  - $pw \longrightarrow$ treedepth $\longrightarrow$ vertex integrity $\longrightarrow vc$
- Improve the existing kernels.

(2) Reducing time values (with the BIP)

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w_1', \ldots, w_d')$ of positive integers such that:

- $w_i' \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w_i' x_i \geq 0.$$

(2) Reducing time values (with the BIP)

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w'_1, \ldots, w'_d)$ of positive integers such that:

- $w'_i \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w'_i x_i \geq 0.$$

- [Lasserre, Queyranne 92] Binary IP equivalent to $1|prec, r_j, d_j|\star$.

$$BIP(I): \begin{cases} \text{One job per position} & \sum_{j=1}^{n} x_{i,j} = 1 \text{ for all } 1 \leq i \leq n. \\ \text{One position per job} & \sum_{i=1}^{n} x_{i,j} = 1 \text{ for all } 1 \leq j \leq n. \\ \text{Precedence constraints} & \sum_{i=1}^{i_0} (x_{i,j_1} - x_{i,j_2}) \geq 0 \text{ for all } 1 \leq i_0 \leq n,\ (j_1, j_2) \in A. \\ \text{Time window checks} & \sum_{j=1}^{n} r_j \cdot x_{i_1,j} + \sum_{i=i_1}^{i_2} \sum_{j=1}^{n} p_j \cdot x_{i,j} \leq \sum_{j=1}^{n} \bar{d}_j \cdot x_{i_2,j} \\ & \text{for all } 1 \leq i_1 \leq i_2 \leq n. \end{cases}$$

(2) Reducing time values (with the BIP)

Theorem [Frank, Tardos 87]

Let $\mathtt{w} = (w_1, \ldots, w_d)$ be a vector of rational numbers and let N be a positive integer. Then in time $\mathcal{O}(d^8 \cdot (d + log(N)))$ one can build a vector $\mathtt{w}' = (w'_1, \ldots, w'_d)$ of positive integers such that:

- $w'_i \leq 2^{4d^3} \cdot N^{d(d+2)}$ for all $1 \leq i \leq d$,
- for every $\mathtt{x} = (x_1, \ldots, x_d)$ vector of integers such that $\sum_{i=1}^{d} |x_i| \leq N - 1$:

$$\sum_{i=1}^{d} w_i x_i \geq 0 \text{ if and only if } \sum_{i=1}^{d} w'_i x_i \geq 0.$$

- [Lasserre, Queyranne 92] Binary IP equivalent to $1|prec, r_j, d_j|\star$.

$$BIP(I) : \begin{cases} \text{One job per position} & \sum_{j=1}^{n} x_{i,j} = 1 \text{ for all } 1 \leq i \leq n. \\ \text{One position per job} & \sum_{i=1}^{n} x_{i,j} = 1 \text{ for all } 1 \leq j \leq n. \\ \text{Precedence constraints} & \sum_{i=1}^{i_0} (x_{i,j_1} - x_{i,j_2}) \geq 0 \text{ for all } 1 \leq i_0 \leq n,\ (j_1, j_2) \in A. \\ \text{Time window checks} & \sum_{j=1}^{n} r_j \cdot x_{i_1,j} + \sum_{i=i_1}^{i_2} \sum_{j=1}^{n} p_j \cdot x_{i,j} \leq \sum_{j=1}^{n} \bar{d}_j \cdot x_{i_2,j} \\ & \text{for all } 1 \leq i_1 \leq i_2 \leq n. \end{cases}$$

- Use the theorem on instance I' with $\tilde{n} = \mathcal{O}(vc^2)$ jobs.
  - $d = 3\tilde{n}$, $\mathtt{w} = (p_1, r_1, d_1, \ldots, p_{\tilde{n}}, r_{\tilde{n}}, d_{\tilde{n}})$, $N = \tilde{n} + 3$.