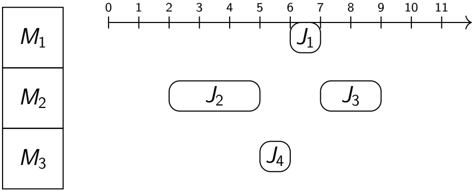
A New Structural Parameter on Single Machine Scheduling with Release Dates and Deadlines

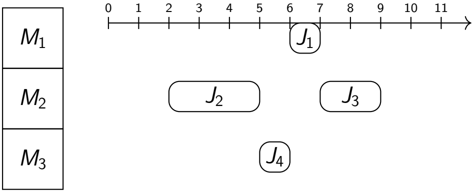
Maher Mallem, Claire Hanen, Alix Munier-Kordon

15 May 2024

Machine scheduling

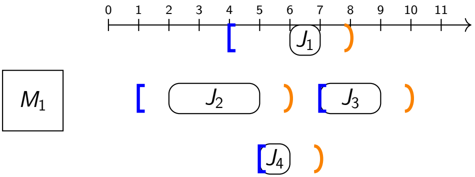


Machine scheduling



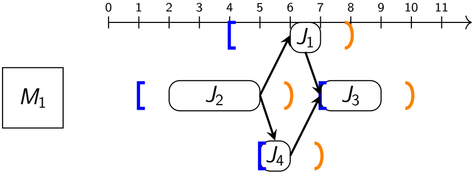
machines | job properties | optimization criterion

Machine scheduling



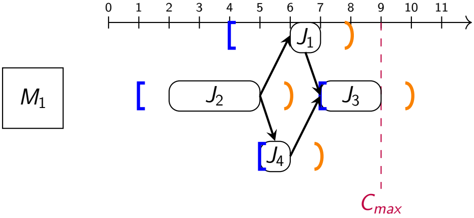
$$1|r_j, d_j|\star$$

Machine scheduling



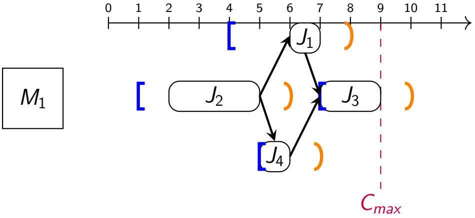
$$1|prec, r_j, d_j|\star$$

Machine scheduling



$$1|prec,r_j,d_j|C_{max}$$

Machine scheduling



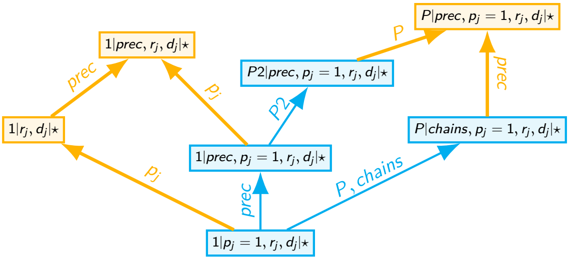
$1|prec, r_j, d_j|C_{max}$

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

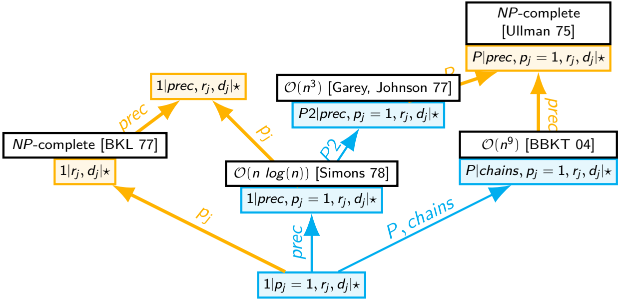
Classical complexity theory

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

Classical complexity theory



Classical complexity theory



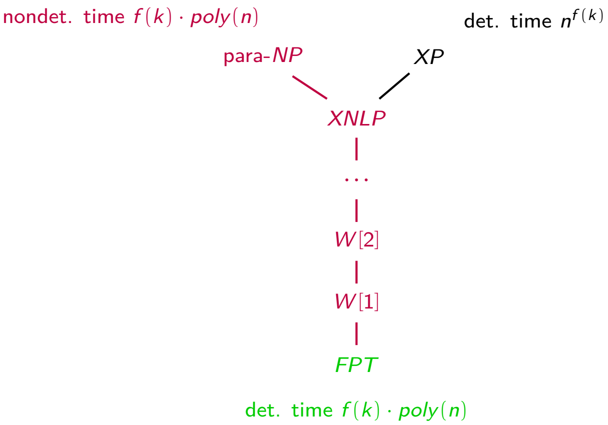
Going beyond NP-hardness

- Problem $\mathcal{P}$
- P: deterministic time $poly(n)$
- NP: nondeterministic time $poly(n)$

Going beyond NP-hardness

- Problem $\mathcal{P}$
  - Problem $(\mathcal{P}, k)$
- P: deterministic time $poly(n)$
  - FPT: deterministic time $f(k) \cdot poly(n)$
- NP: nondeterministic time $poly(n)$
  - para-NP: nondeterministic time $f(k) \cdot poly(n)$

Parameterized complexity classes



- Negative results via *FPT* reductions

Parameterized complexity in machine scheduling

- Negative results via *FPT* reductions
  - [Bodlaender, Fellows 1995]
    #machines m
    $P|prec, p_j = 1|C_{max}$ is W[2]-hard.

- Negative results via *FPT* reductions
  - [Bodlaender, Fellows 1995]
    #machines m
    $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- Mixed Integer Linear Programming

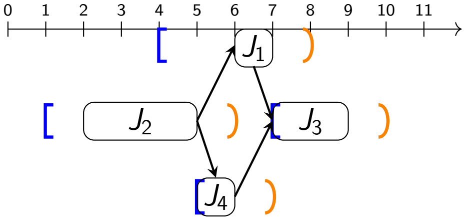
- Negative results via FPT reductions
  - [Bodlaender, Fellows 1995]
    #machines m
    $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- Mixed Integer Linear Programming
  - [Hermelin, Karhi, Pinedo, Shabtay 2018]
    #distinct d_j, #distinct p_j, #distinct w_j
    $1||\sum w_j U_j \in$ FPT with two of the three.

Parameterized complexity in machine scheduling

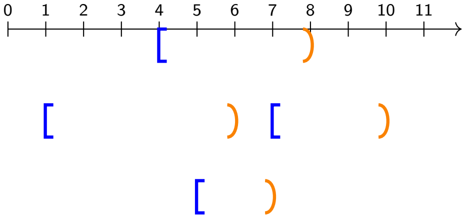
- Negative results via *FPT* reductions
  - [Bodlaender, Fellows 1995]
    #machines m
    $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- Mixed Integer Linear Programming
  - [Hermelin, Karhi, Pinedo, Shabtay 2018]
    #distinct d_j, #distinct p_j, #distinct w_j
    $1||\sum w_j U_j \in$ FPT with two of the three.
- Dynamic Programming

- Negative results via *FPT* reductions
  - [Bodlaender, Fellows 1995]
    #machines m
    $P|prec, p_j = 1|C_{max}$ is W[2]-hard.
- Mixed Integer Linear Programming
  - [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.
- Dynamic Programming
  - [Baart, He, de Weerdt 2021]
    slack $\sigma = max_j(d_j - r_j - p_j)$
    $1|r_j, s_{jk}, reject, \bar{d}_j| \sum_{j \notin R}(v_j - w_j T_j)$ in time $\mathcal{O}(\sigma^3 4^\sigma \cdot n^2)$.

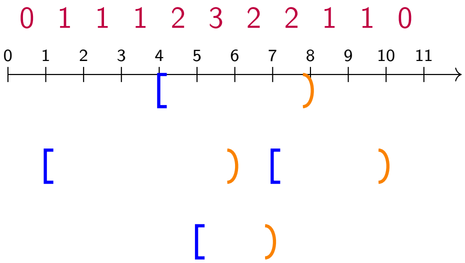
Pathwidth μ



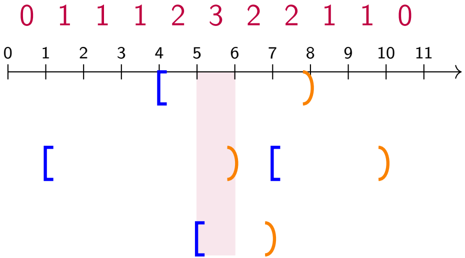
Pathwidth μ



Pathwidth μ

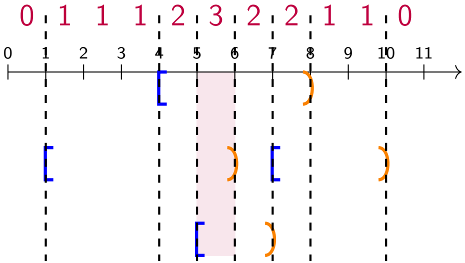


Pathwidth μ



$\mu = 3$

Pathwidth μ



$\mu = 3$

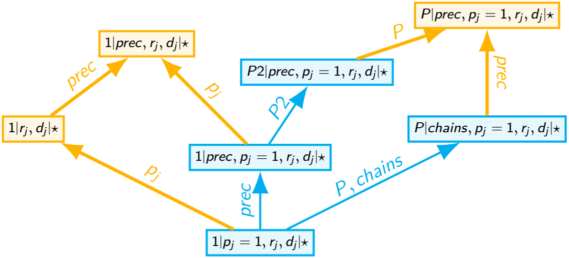
Results with parameter μ

- [Munier 21]
  - $P|prec, p_j = 1, r_j, d_j|C_{max}$
  - Exact resolution in time $\mathcal{O}(16^{\mu} \cdot n^4)$.
- [Baart, He, de Weerdt 21]
  - $1|r_j, s_{jk}, reject, \bar{d}_j| \sum_{j \notin R}(v_j - w_j T_j)$
  - $(1 - \epsilon)$-approximation in time $\mathcal{O}(\mu^2\ 2^{\mu} \cdot \frac{n^3}{\epsilon^2})$.

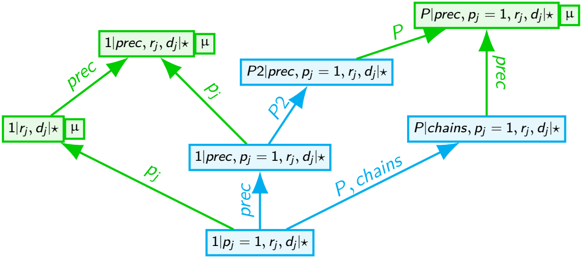
Results with parameter μ

- [Munier 21]
  - $P|prec, p_j = 1, r_j, d_j|C_{max}$
  - Exact resolution in time $\mathcal{O}(16^{\mu} \cdot n^4)$.
- [Baart, He, de Weerdt 21]
  - $1|r_j, s_{jk}, reject, \bar{d}_j| \sum_{j \notin R}(v_j - w_j T_j)$
  - $(1-\epsilon)$-approximation in time $\mathcal{O}(\mu^2 \, 2^{\mu} \cdot \frac{n^3}{\epsilon^2})$.
- [MAPSP 22]
  - $1|prec, r_j, d_j|C_{max}$
  - Exact resolution in time $\mathcal{O}(\mu^2 \, 4^{\mu} \cdot n + n^2)$.

Parameter μ

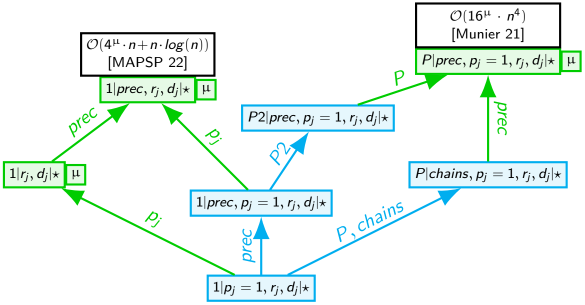

$P|prec, p_j = 1, r_j, d_j|\star$
P
$1|prec, r_j, d_j|\star$
$P2|prec, p_j = 1, r_j, d_j|\star$
$prec$
$prec$
p_j
$P2$
$1|r_j, d_j|\star$
$P|chains, p_j = 1, r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
p_j
$P, chains$
$prec$
$1|p_j = 1, r_j, d_j|\star$

Parameter μ

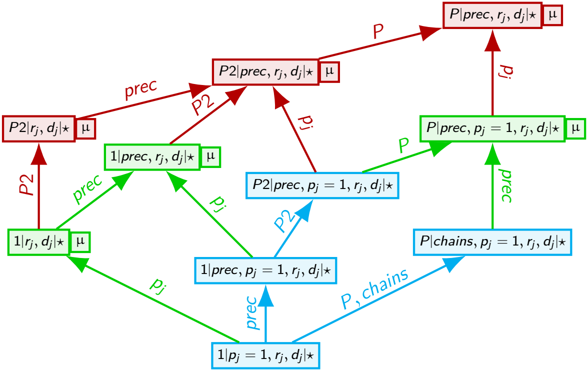


$P|prec, p_j = 1, r_j, d_j|\star$
μ
$1|prec, r_j, d_j|\star$
μ
P
$P2|prec, p_j = 1, r_j, d_j|\star$
prec
prec
p_j
$P2$
$1|r_j, d_j|\star$
μ
$P|chains, p_j = 1, r_j, d_j|\star$
$1|prec, p_j = 1, r_j, d_j|\star$
p_j
P, chains
prec
$1|p_j = 1, r_j, d_j|\star$

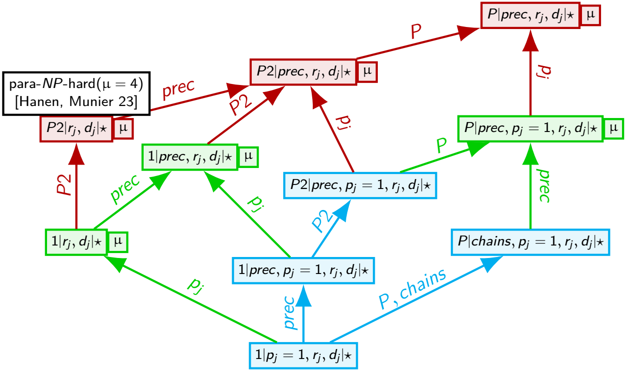
Parameter μ


$\mathcal{O}(4^{\mu} \cdot n + n \cdot log(n))$
[MAPSP 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
P, chains
prec
$1|p_j = 1, r_j, d_j|\star$

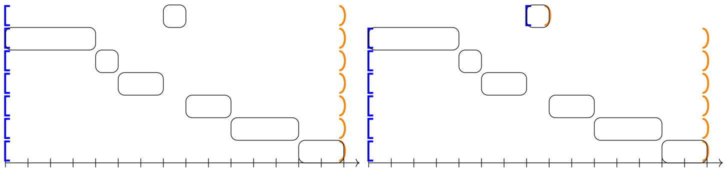
Parameter μ


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

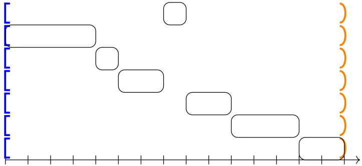
Parameter μ



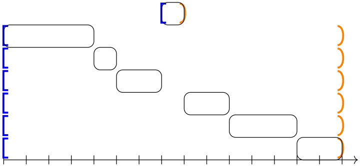
Pathwidth limitation



Pathwidth limitation

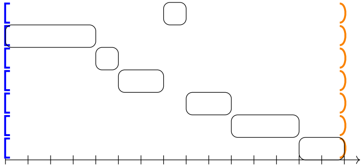


$\mu = n$

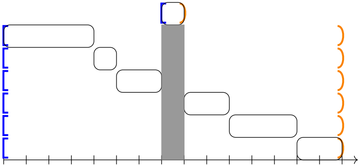


$\mu = n$

Pathwidth limitation

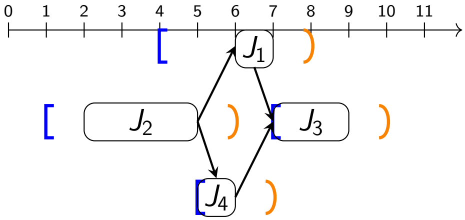


$\mu = n$

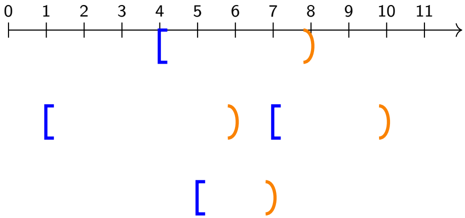


$\mu = n$

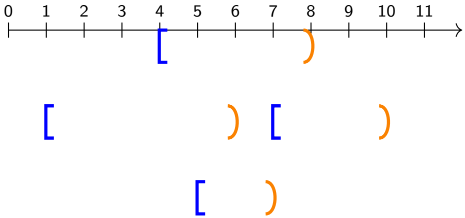
q-Proper Level q



q-Proper Level q



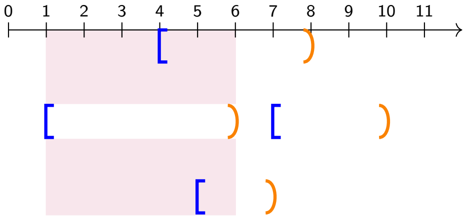
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

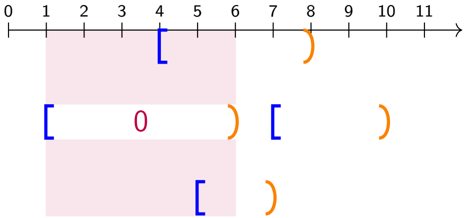
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

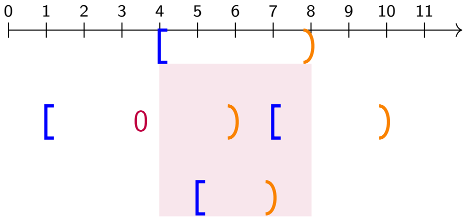
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

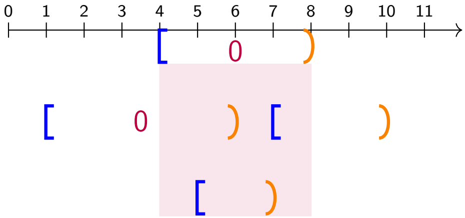
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \land d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

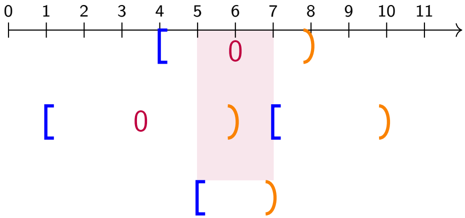
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

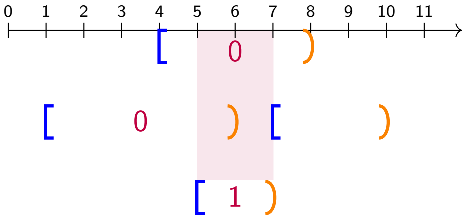
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \; |\Pi_i|$

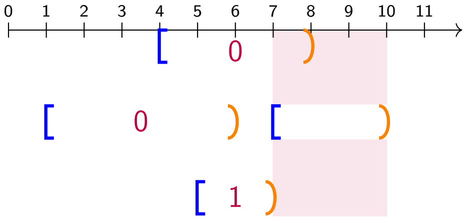
q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

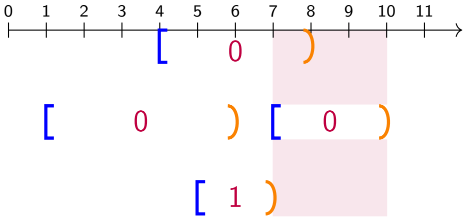
q-Proper Level q



Definition: Proper sets and q-proper level

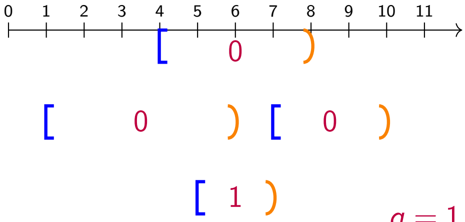
- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

q-Proper Level q



Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \land d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

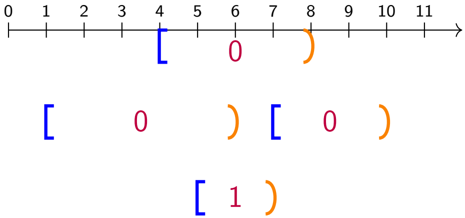


$q = 1$

Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$

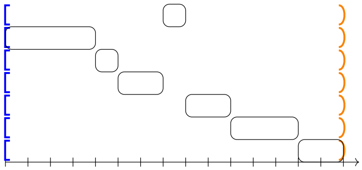
q-Proper Level q



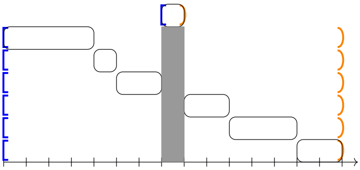
Definition: Proper sets and q-proper level

- $\forall i \in \mathcal{J}, \Pi_i = \{j \in \mathcal{J}, r_j < r_i \wedge d_i < d_j\}$ [Erschler, Fontan, Merce 85]
- $q = max_{i \in \mathcal{J}} \ |\Pi_i|$ [Proskurowski, Telle 99]

Pathwidth vs q-proper level

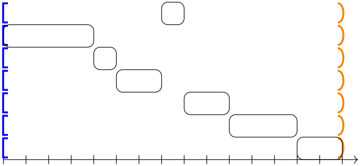


$\mu = n$



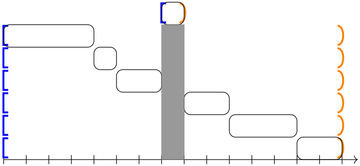
$\mu = n$

Pathwidth vs q-proper level



$$\mu = n$$

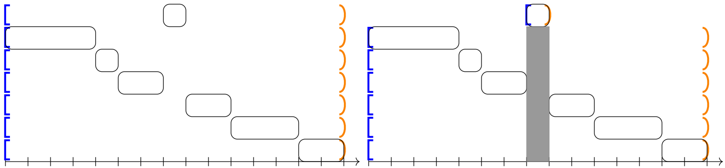
$$q = 0$$



$$\mu = n$$

$$q = n - 1$$

Pathwidth vs q-proper level



$\mu = n$

$q = 0$

$\mu = n$

$q = n - 1$

Result

$$q \leq \mu - 1$$

Extending the Earliest Deadline rule

- σ optimal schedule on $1|r_j, d_j|C_{max}$ (= job ordering).
- $\prec$ = lexicographic order (non decreasing d_j, non decreasing r_j) on $\mathcal{J}$.

Extending the Earliest Deadline rule

- σ optimal schedule on $1|r_j, d_j|C_{max}$ (= job ordering).
- $\prec$ = lexicographic order (non decreasing d_j, non decreasing r_j) on $\mathcal{J}$.

Definition: Earliest Deadline rule (ED) [Lawler, Moore 69]

Schedule σ follows the (ED) rule if:

$$\forall (i,j) \in \mathcal{J}^2, i \prec j \implies \sigma(i) < \sigma(j).$$

Extending the Earliest Deadline rule (cont.)

Definition: weak Earliest Deadline rule (wED)

Schedule σ follows the (wED) rule if:

$$\forall (i,j) \in \mathcal{J}^2, i \prec j \implies [\sigma(i) < \sigma(j)] \vee [j \in \Pi_i] \vee [\exists h \prec i, \sigma(j) < \sigma(h) < \sigma(i)].$$

Extending the Earliest Deadline rule (cont.)

Definition: weak Earliest Deadline rule (wED)

Schedule σ follows the (wED) rule if:

$$\forall (i,j) \in \mathcal{J}^2, i \prec j \implies [\sigma(i) < \sigma(j)] \vee [j \in \Pi_i] \vee [\exists h \prec i, \sigma(j) < \sigma(h) < \sigma(i)].$$

Definition: Summit

A summit is a job i in $\mathcal{J}$ such that: $\forall j \in \mathcal{J}, i \notin \Pi_j$.

Extending the Earliest Deadline rule (cont.)

Definition: weak Earliest Deadline rule (wED)

Schedule σ follows the (wED) rule if:

$$\forall (i,j) \in \mathcal{J}^2, i \prec j \implies [\sigma(i) < \sigma(j)] \vee [j \in \Pi_i] \vee [\exists h \prec i, \sigma(j) < \sigma(h) < \sigma(i)].$$

Definition: Summit

A summit is a job i in $\mathcal{J}$ such that: $\forall j \in \mathcal{J}, i \notin \Pi_j$.

- $s_1 \prec \ldots \prec s_N$ summits of the instance ($N \leq n$).

Extending the Earliest Deadline rule (cont.)

Definition: weak Earliest Deadline rule (wED)

Schedule σ follows the (wED) rule if:

$$\forall (i,j) \in \mathcal{J}^2, i \prec j \implies [\sigma(i) < \sigma(j)] \vee [j \in \Pi_i] \vee [\exists h \prec i, \sigma(j) < \sigma(h) < \sigma(i)].$$

Definition: Summit

A summit is a job i in $\mathcal{J}$ such that: $\forall j \in \mathcal{J}, i \notin \Pi_j$.

- $s_1 \prec \ldots \prec s_N$ summits of the instance ($N \leq n$).

Lemma 1

In any feasible schedule σ which follows the (wED) rule:

(i) for all k in $[1, N]$ and $i \prec s_k$ we have $\sigma(i) < \sigma(s_k)$,

(ii) for all k in $[1, N]$ and $j \succ s_k$ if $\sigma(j) < \sigma(s_k)$ then $j \in \Pi_{s_k}$.

Summit-based dominance

Lemma 2

Any feasible schedule σ which follows the (wED) rule is of the form $T_1 s_1 \dots T_N s_N T_{N+1}$ where:

a. every job in sequence T_1 is in set Π_{s_1},

b. for all k in $[2, N]$ every job in sequence T_k is in set $\Pi_{s_{k-1}} \cup \Pi_{s_k}$, and

c. every job in sequence T_{N+1} is in set Π_{s_N}.

- TODO: Gantt chart

Summit-based dominance (cont.)

Definition: Summit-ordered schedule

A feasible schedule $\sigma = T_1 s_1 \dots T_N s_N T_{N+1}$ is summit-ordered if it follows Lemmas 1 and 2, and:

a. all jobs in sequence T_1 are in Π_{s_1} and scheduled in nondecreasing order of their release date.

b. for all k in $[1, N-1]$ $T_{k+1} = C_k A_{k+1} B_{k+1}$ where:
   - all jobs in sequence C_k are in Π_{s_k}, not in $\Pi_{s_{k+1}}$ and scheduled in nondecreasing order of their deadline,
   - all jobs in sequence A_{k+1} are in $\Pi_{s_k} \cap \Pi_{s_{k+1}}$ and scheduled in any order (say in their lexicographic order),
   - all jobs in sequence B_{k+1} are in $\Pi_{s_{k+1}}$, not in Π_{s_k} and scheduled in nondecreasing order of their release date.

c. all jobs in sequence T_{N+1} are in Π_{s_N} and scheduled in nondecreasing order of their deadline.

Summit-based dominance (cont.)

Definition: Summit-ordered schedule

A feasible schedule $\sigma = T_1 s_1 \dots T_N s_N T_{N+1}$ is summit-ordered if it follows Lemmas 1 and 2, and:

a. all jobs in sequence T_1 are in Π_{s_1} and scheduled in nondecreasing order of their release date.

b. for all k in $[1, N-1]$ $T_{k+1} = C_k A_{k+1} B_{k+1}$ where:

- all jobs in sequence C_k are in Π_{s_k}, not in $\Pi_{s_{k+1}}$ and scheduled in nondecreasing order of their deadline,
- all jobs in sequence A_{k+1} are in $\Pi_{s_k} \cap \Pi_{s_{k+1}}$ and scheduled in any order (say in their lexicographic order),
- all jobs in sequence B_{k+1} are in $\Pi_{s_{k+1}}$, not in Π_{s_k} and scheduled in nondecreasing order of their release date.

c. all jobs in sequence T_{N+1} are in Π_{s_N} and scheduled in nondecreasing order of their deadline.

Proposition

On $1|r_j, d_j|C_{max}$ the summit-ordered schedules are dominant.

Conclusion

- New structural parameter on single machine scheduling with job time windows.

Conclusion

- New structural parameter on single machine scheduling with job time windows.

- *FPT* algorithm on $1|prec, r_j, d_j|C_{max}$ derived from a new dominance rule.

Conclusion

- New structural parameter on single machine scheduling with job time windows.
- FPT algorithm on $1|prec, r_j, d_j|C_{max}$ derived from a new dominance rule.
- $P|prec, p_j = 1, r_j, d_j|\star$ parameterized by q?