Processing-time dependent profit maximization scheduling problems

with applications to star observations

Florian Fontan

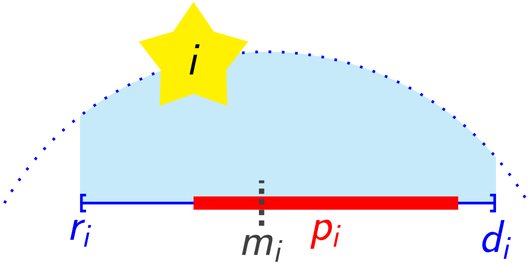
under the supervision of Nadia Brauner and Pierre Lemaire

G-SCOP

The star scheduling problem

The star scheduling problem

The star scheduling problem



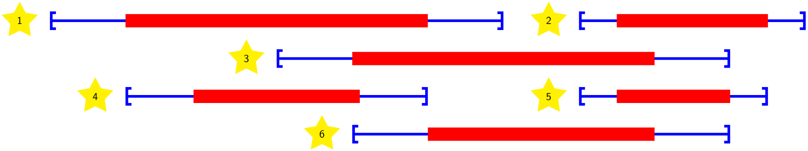
- $[r_i; d_i)$: visibility interval of star i
- p_i: required duration of the observation of star i
- w_i: scientific interest of observing star i

The meridian $m_i \in [r_i, d_i)$ is a mandatory instant of the observation.

G-SCOP

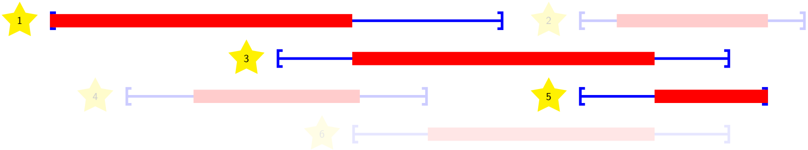
The star scheduling problem

Instance: A set $\mathcal{N}$ of stars; each star $i \in \mathcal{N}$ has an interest w_i, an observation duration p_i and a visibility window $[r_i; d_i)$



The star scheduling problem

Instance: A set $\mathcal{N}$ of stars; each star $i \in \mathcal{N}$ has an interest w_i, an observation duration p_i and a visibility window $[r_i; d_i)$

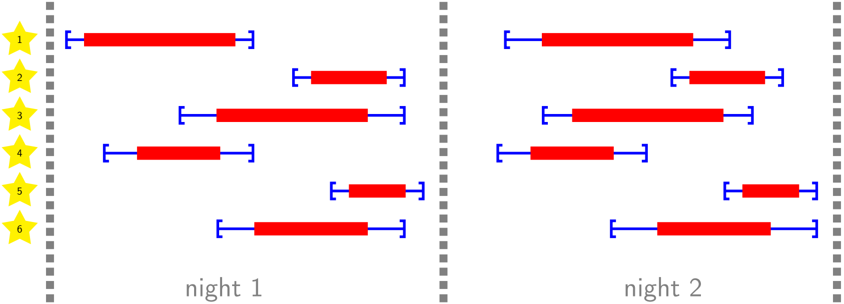


Question: find $\mathcal{N}' \subset \mathcal{N}$ as well as the starting dates of the observations $s_i, \forall i \in \mathcal{N}'$ such that

- for all $i \in \mathcal{N}'$: $[s_i; s_i + p_i) \subset [r_i; d_i)$
- for all $(i_1, i_2) \in \mathcal{N}'^2$: $[s_{i_1}; s_{i_1} + p_{i_1}) \cap [s_{i_2}; s_{i_2} + p_{i_2}) = \emptyset$
- $\sum_{i \in \mathcal{N}'} w_i$ is maximized

The star scheduling problem

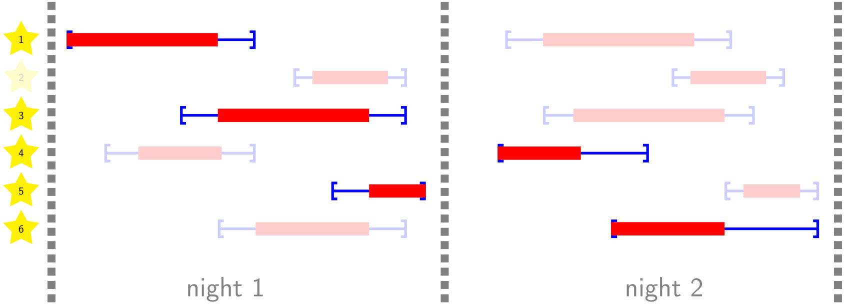
Instance: a set $\mathcal{M}$ of nights, a set $\mathcal{N}$ of stars; each star $i \in \mathcal{N}$ has an interest w_i, an observation duration p_i^j and a visibility window $[r_i^j; d_i^j)$, dependent of the night j of the observation.



The practical instances include around 800 stars for a 6 months planification.

The star scheduling problem

Instance: a set $\mathcal{M}$ of nights, a set $\mathcal{N}$ of stars; each star $i \in \mathcal{N}$ has an interest w_i, an observation duration p_i^j and a visibility window $[r_i^j ; d_i^j)$, dependent of the night j of the observation.



The practical instances include around 800 stars for a 6 months planification.

The star scheduling problem

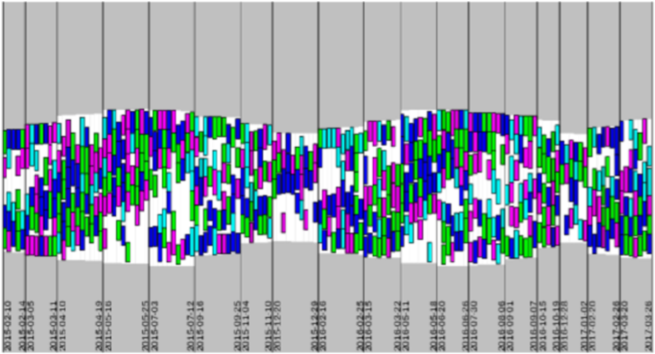
Nicolas Catusse, Hadrien Cambazard, Nadia Brauner, Pierre Lemaire, and Bernard Penz - Anne-Marie Lagrange and Pascal Rubini (IJCAI 2016) :

- Branch-and-price, local search

Model extensions:

- Observation duration depending on starting date
- Calibrations
- Night reservation
- **Variable interests**

The star scheduling problem

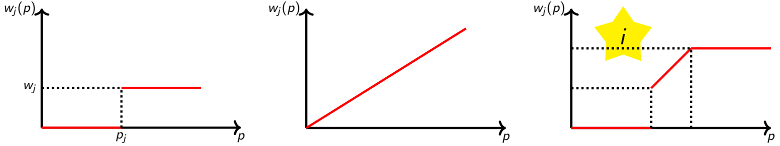


Processing-time dependent profit

Processing-time dependent profit

Instance: A set $\mathcal{N}$ of jobs. Each job $j \in \mathcal{N}$ has deadline d_j and a profit function $w_j(p_j)$, p_j the allocated processing-time of job j

Profit function for a classical scheduling problem, for the linear model and for a star observation:

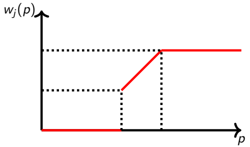


Question: find $\mathcal{N}' \subset \mathcal{N}$ such that $\sum_{j \in \mathcal{N}'} w_j(p_j)$ is maximized.

Processing-time dependent profit

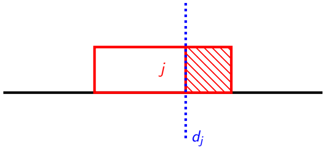
Controllable processing-times, Imprecise computation:

- Dvir Shabtay and George Steiner. A survey of scheduling with controllable processing times, 2007.



Processing-time dependent profit

- *Late work*:
  - Malgorzata Sterna. A survey of scheduling problems with late work criteria, 2011.



- *Increasing Reward with Increasing Service (IRIS)*:
  - Jayanta K Dey, James Kurose, and Don Towsley. On-line processor scheduling for class of iris (increasing reward with increasing service) real-time tasks, 1993.

Processing-time dependent profit

Goal: finding the NP-hardness limits for each model of the profit function depending on the machine environment and the constraints.

Machine environment: 1, Pm, P.

Constraints

- Release dates
- Preemption
- Identical parameters

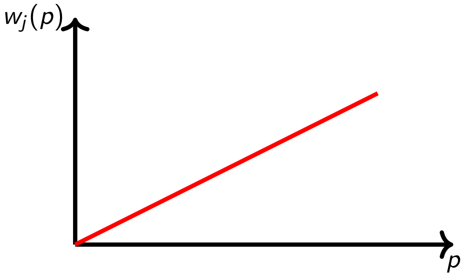
Simple examples

Simple examples

For all $j \in \mathcal{T}$:

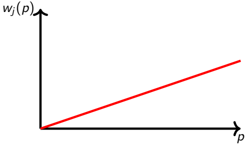
$$w_j(p) = b_j p$$

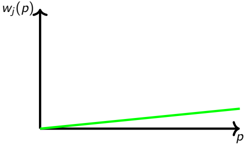
where b_j is called the growth rate of job j.

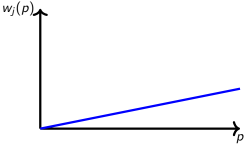


Simple examples

- No constraints, Common deadline:
  $1|w_j(p) = b_j p, d_j{=}d| - \sum w_j(p_j)$:

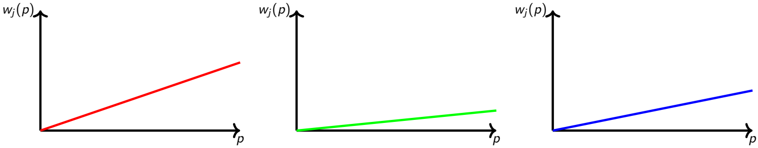


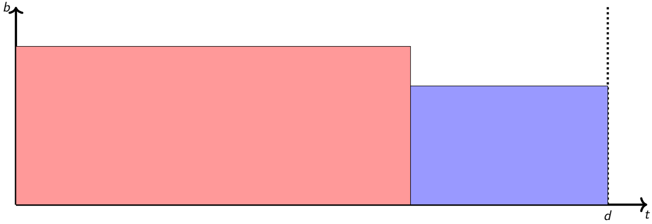




Simple examples

- No constraints, Common deadline:
  $1|w_j(p) = b_j p, d_j{=}d| - \sum w_j(p_j)$:

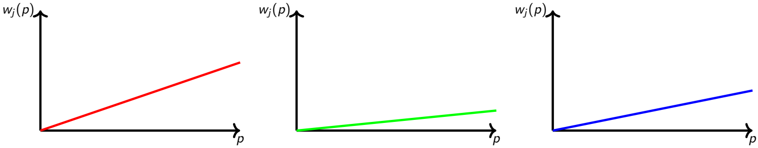


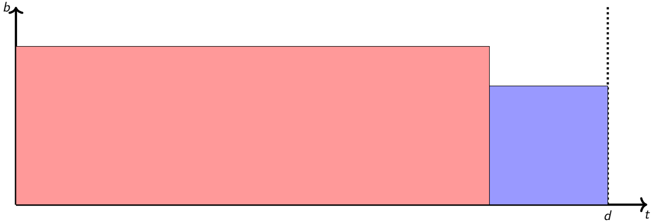


G-SCOP

Simple examples

- No constraints, Common deadline:
  $1|w_j(p) = b_j p, d_j{=}d| - \sum w_j(p_j)$:

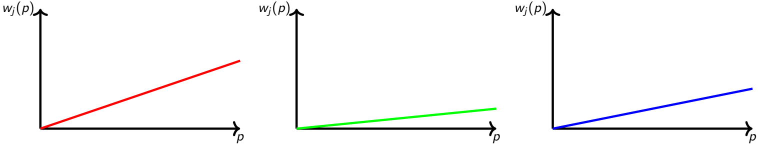




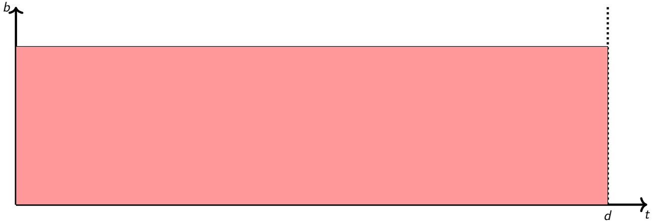
G-SCOP

Simple examples

- No constraints, Common deadline:
  $1|w_j(p) = b_j p, d_j{=}d| - \sum w_j(p_j)$:

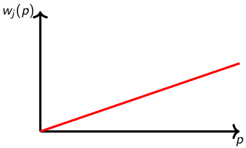


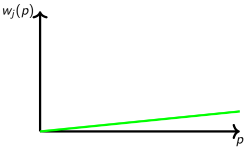
$\Longrightarrow$ Schedule only the job with the maximum growth rate b_j.

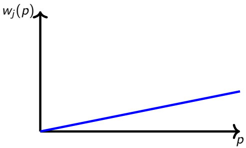


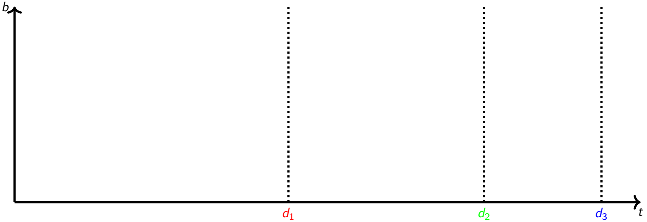
Simple examples

- Distinct deadlines $1|w_j(p) = b_j p| - \sum w_j(p_j)$:



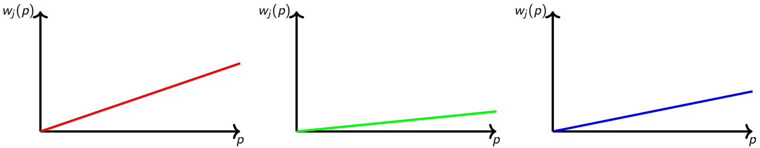


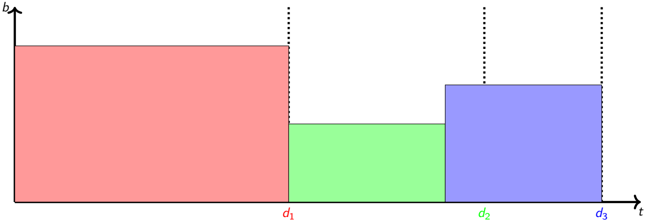




Simple examples

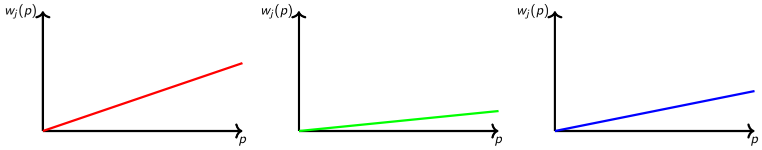
- Distinct deadlines $1|w_j(p) = b_j p| - \sum w_j(p_j)$:



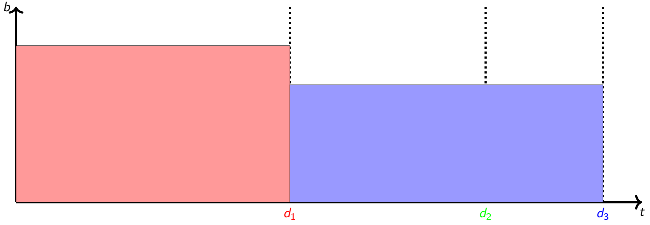


Simple examples

- Distinct deadlines $1|w_j(p) = b_j p| - \sum w_j(p_j)$:



$\Longrightarrow$ Schedule the best available job till its deadline.



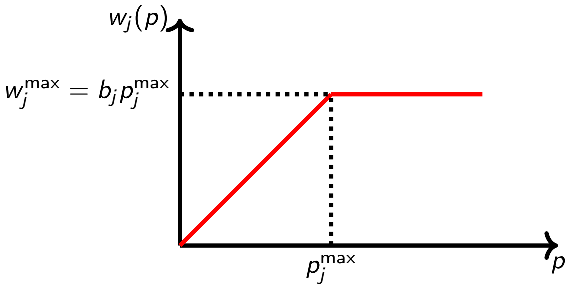
G-SCOP

Simple examples

For all $j \in \mathcal{T}$:

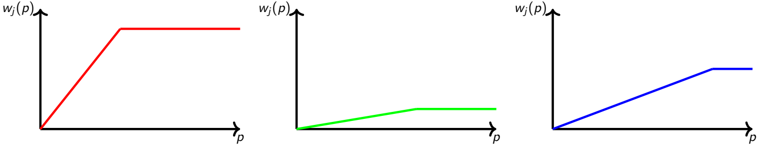
$$w_j(p) = \begin{cases} b_j p, & p \leq p_j^{\max} \\ w_j^{\max}, & p > p_j^{\max} \end{cases}$$

where $w_j^{\max}$ is called the maximum profit, b_j is called the growth rate and $p_j^{\max}$ the maximum processing-time of job j.



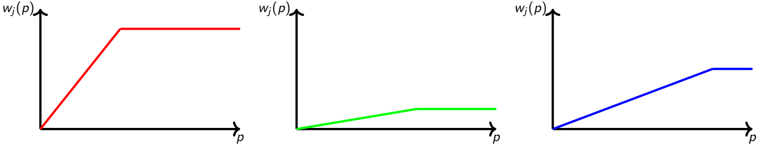
Simple examples

- No constraint, Common deadline
  $1|w_j(p) = \min\{b_j p, w_j^{\max}\}, d_j{=}d| - \sum w_j(p_j)$

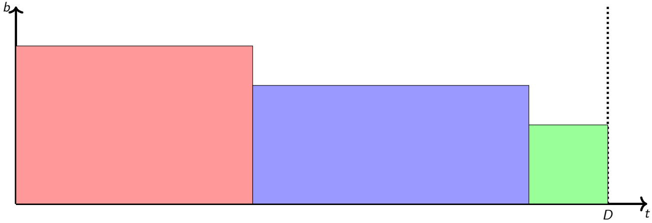


Simple examples

- No constraint, Common deadline
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}, d_j{=}d| - \sum w_j(p_j)$



$\Longrightarrow$ Schedule jobs in non-increasing order of their growth rate



G-SCOP

Simple examples

- Distinct deadlines:
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}| - \sum w_j(p_j)$

Simple examples

- Distinct deadlines:
  $1|w_j(p) = \min\{b_j p, w_j^{\max}\} | -\sum w_j(p_j)$
  - The set of solutions for which every job is scheduled and in non-decreasing order of their deadline is dominant.

G-SCOP

Simple examples

- Distinct deadlines:
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}| - \sum w_j(p_j)$
  - The set of solutions for which every job is scheduled and in non-decreasing order of their deadline is dominant.

$$\max \sum_{j=1}^{n} b_j p_j$$

$$\begin{aligned} s_j + p_j &\leq d_j & \forall j &= 1, \dots, n \\ s_{j+1} &= s_j + p_j & \forall j &= 1, \dots, n-1 \\ b_j p_j &\leq w_j^{\max} & \forall j &= 1, \dots, n \end{aligned}$$

Simple examples

- Release dates:
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}, r_j| - \sum w_j(p_j)$

G-SCOP

Simple examples

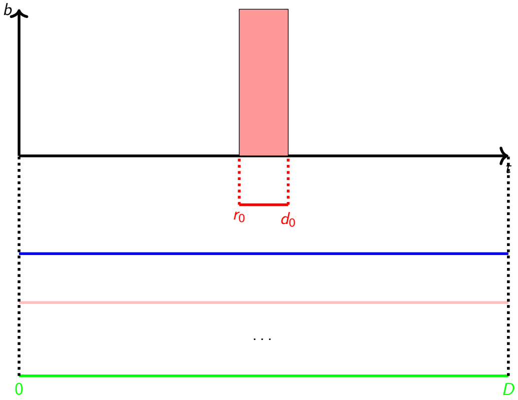
- Release dates:
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}, r_j| - \sum w_j(p_j)$

Strongly NP-complete, reduction from 3-Partition.

Simple examples

- Release dates:
  $1|w_j(p) = \min\{b_j p, w_j^{max}\}, r_j| - \sum w_j(p_j)$

Strongly NP-complete, reduction from 3-Partition.



G-SCOP

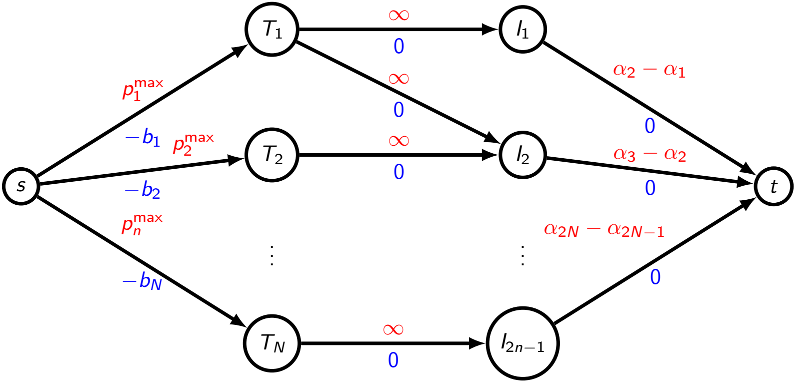
Simple examples

- Release dates and preemption
  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}, r_j, \text{pmtn}| - \sum w_j(p_j)$

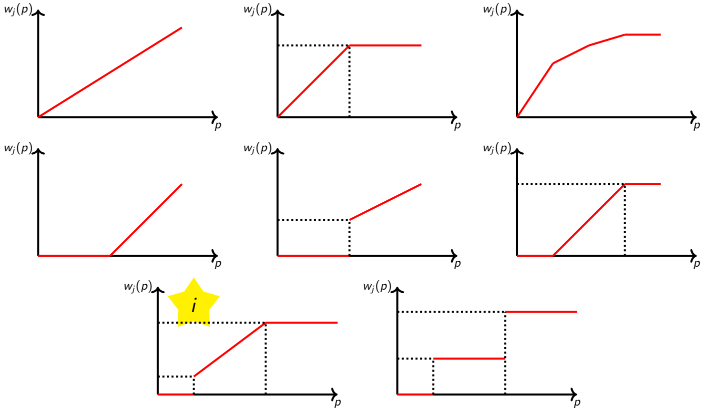
Simple examples

- Release dates and preemption

  $1|w_j(p) = \min\left\{b_j p, w_j^{\max}\right\}, r_j, \text{pmtn}| - \sum w_j(p_j)$



Simple examples



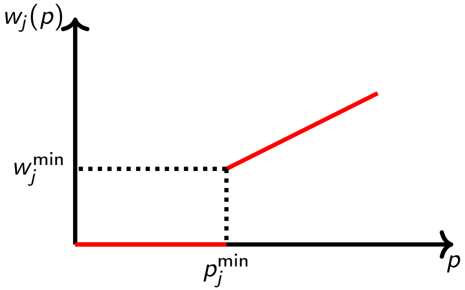
$P \mid \mathrm{LPSTIP}, p_j^{\mathsf{min}}{=}p^{\mathsf{min}}, d_j{=}d$

$P \mid \mathrm{LPSTIP}, p_j^{\min} = p^{\min}, d_j = d$

For all $j \in \mathcal{T}$:

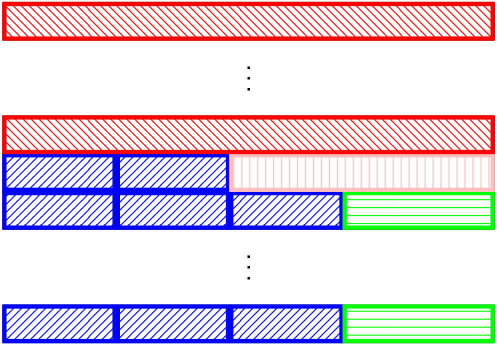
$$w_j(p) = \begin{cases} 0 & p < p_j^{\min} \\ w_j^{\min} + b_j(p - p_j^{\min}), & p \geq p_j^{\min} \end{cases}$$

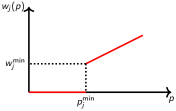
where $w_j^{\min}$ is called the minimum profit, b_j is called the growth rate and $p_j^{\min}$ the minimum processing-time of job j.



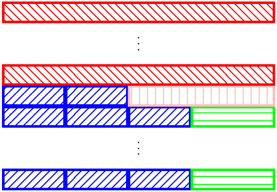
$P \mid \mathrm{LPSTIP}, p_j^{\min}{=}p^{\min}, d_j{=}d$

Dominant structure of solutions:





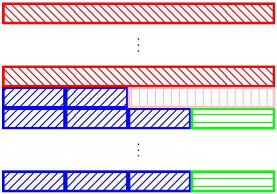
$P \mid \mathrm{LPSTIP}, p_j^{\min}{=}p^{\min}, d_j{=}d$



$$
\begin{aligned}
U_1(S) &= \{j \in S, \quad p_j = D\} \\
U_2(S) &= \{j \in S, \quad T_j \text{ is a special job}\} \\
U_3(S) &= \{j \in S, \quad p_j = p^{\min} + r\} \\
U_4(S) &= \{j \in S, \quad p_j = p^{\min}\} \\
U_0(S) &= \{j \notin S\}
\end{aligned}
$$

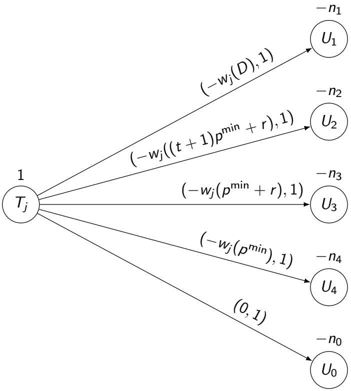
$$
t(S) = \begin{cases} 0, & \text{if } U_2(S) = \emptyset \\ \frac{D - p_j}{p^{\min}}, & \text{if } U_2(S) = \{T_j\} \end{cases}
$$

$P \mid \mathrm{LPSTIP}, p_j^{\min}=p^{\min}, d_j=d$



$$U_1(S) = \{j \in S, \quad p_j = D\}$$
$$U_2(S) = \{j \in S, \quad T_j \text{ is a special job}\}$$
$$U_3(S) = \{j \in S, \quad p_j = p^{\min} + r\}$$
$$U_4(S) = \{j \in S, \quad p_j = p^{\min}\}$$
$$U_0(S) = \{j \notin S\}$$

$$t(S) = \begin{cases} 0, & \text{if } U_2(S) = \emptyset \\ \frac{D-p_j}{p^{\min}}, & \text{if } U_2(S) = \{T_j\} \end{cases}$$



$P \mid \mathrm{LPSTIP}, p_j^{\mathbf{min}}{=}p^{\mathbf{min}}, d_j{=}d$

$$0 \leq N_1(S) \leq m \qquad 0 \leq t(S) \leq n-1$$

$P \mid \mathrm{LPSTIP}, p_j^{\min}=p^{\min}, d_j=d$

$$0 \leq N_1(S) \leq m \qquad 0 \leq t(S) \leq n-1$$

$$N_2(S) = \begin{cases} 0, & \text{if } t(S) = 0 \\ 1, & \text{otherwise} \end{cases}$$

$$N_3(S) = m - N_1(S) - N_2(S)$$

$$N_4(S) = (q-1)N_3(S) + t(S)$$

$$N_0(S) = N - N_1(S) - N_2(S) - N_3(S) - N_4(S)$$

$P \mid \mathrm{LPSTIP}, p_j^{\min} = p^{\min}, d_j = d$

$$0 \le N_1(S) \le m \qquad 0 \le t(S) \le n-1$$

$$N_2(S) = \begin{cases} 0, & \text{if } t(S) = 0 \\ 1, & \text{otherwise} \end{cases}$$

$$N_3(S) = m - N_1(S) - N_2(S)$$

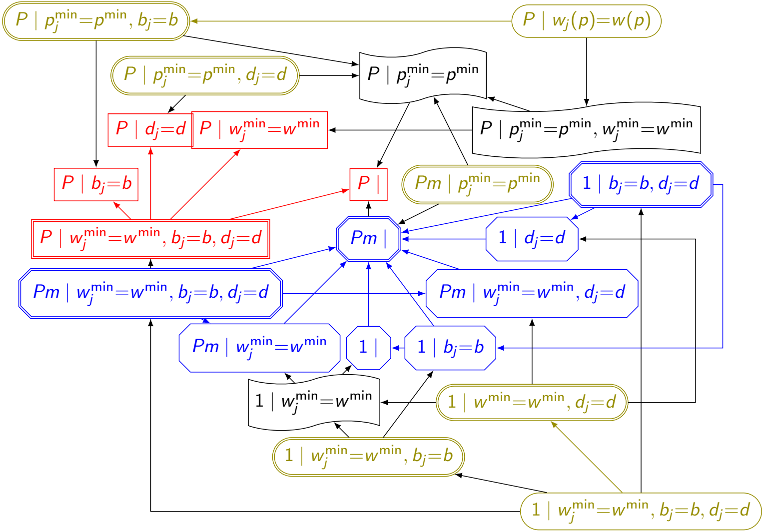
$$N_4(S) = (q-1)N_3(S) + t(S)$$

$$N_0(S) = N - N_1(S) - N_2(S) - N_3(S) - N_4(S)$$

$\Longrightarrow$ The problem can be solved in polynomial time by solving a polynomial number of minimum cost flows

G-SCOP

$P \mid \text{LPSTIP}, p_j^{\min}=p^{\min}, d_j=d$



G-SCOP

$P \mid \mathrm{LPSTIP}, p_j^{\min}{=}p^{\min}, d_j{=}d$

Problem	Compl.	Ref
$P \mid \mathrm{LP}$	P	Th. 1
$P \mid \mathrm{LBP}, b_j{=}b, d_j{=}d$	sNPc	Th. 3
$Pm \mid \mathrm{LBP}, b_j{=}b, d_j{=}d$	wNPc	Th. 3
$P \mid \mathrm{LBP}, w_j^{\max}{=}w^{\max}, d_j{=}d$	sNPc	Th. 3
$Pm \mid \mathrm{LBP}, w_j^{\max}{=}w^{\max}, d_j{=}d$	wNPc	Th. 3
$1 \mid \mathrm{LBP}$	P	Th. 2
$P \mid \mathrm{LBP}, w_j(p){=}w(p), d_j{=}d$	P	Th. 4
$P \mid \mathrm{LPST}, b_j{=}b$	P	Th. 5
$P \mid \mathrm{LPSTIP}, w_j^{\min}{=}w^{\min}, b_j{=}b, d_j{=}d$	sNPc	Th. 6
$Pm \mid \mathrm{LPSTIP}, w_j^{\min}{=}w^{\min}, b_j{=}b, d_j{=}d$	wNPc	Th. 6
$1 \mid \mathrm{LPSTIP}, b_j{=}b, d_j{=}d$	wNPc	Th. 6
$P \mid \mathrm{LPSTIP}, p_j^{\min}{=}p^{\min}, d_j{=}d$	P	Th. 7
$P \mid \mathrm{LPSTIP}, p_j^{\min}{=}p^{\min}, b_j{=}b$	P	Th. 8
$P \mid \mathrm{LBPST}, w_j^{\max}{=}w^{\max}, p_j^{\min}{=}p^{\min}, d_j{=}d$	sNPc	Th. 9
$Pm \mid \mathrm{LBPST}, w_j^{\max}{=}w^{\max}, p_j^{\min}{=}p^{\min}, d_j{=}d$	wNPc	Th. 9
$P \mid \mathrm{LBPST}, p_j^{\max}{=}p^{\max}, d_j{=}d$	P	Th. 11
$1 \mid \mathrm{LBPST}, b_j{=}b, d_j{=}d$	wNPc	Th. 10
$1 \mid \mathrm{LBPST}, b_j{=}b, p_j^{\min}{=}p^{\min}, d_j{=}d$	P	
$P \mid \mathrm{LBPSTIP}, w_j^{\min}{=}w^{\min}, b_j{=}b, p_j^{\max}{=}p^{\max}, w_j^{\max}{=}w^{\max}, d_j{=}d$	sNPc	Th. 12
$Pm \mid \mathrm{LBPSTIP}, w_j^{\min}{=}w^{\min}, b_j{=}b, p_j^{\max}{=}p^{\max}, w_j^{\max}{=}w^{\max}, d_j{=}d$	wNPc	Th. 12
$1 \mid \mathrm{LBPSTIP}, b_j{=}b, w_j^{\max}{=}w^{\max}, d_j{=}d$	wNPc	Th. 13
$1 \mid \mathrm{LBPSTIP}, p_j^{\max}{=}p^{\max}, w_j^{\max}{=}w^{\max}, d_j{=}d$	wNPc	Th. 13
$Pm \mid \mathrm{PLP}, p_j^k{=}p^k$	P	Th. 14
$1 \mid \mathrm{PLP}, w_j^k{=}w^k, b_j^k{=}b$	P	Th. 15
$1 \mid \mathrm{PLP}, w_j^k{=}w^k, b_j^k{=}b_j, d_j{=}d$	P	Th. 16

G-SCOP

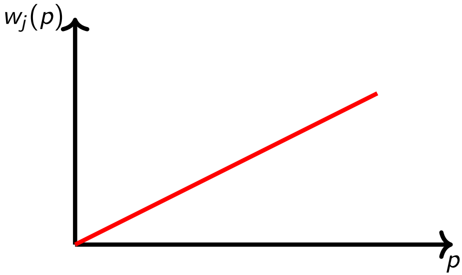
Linear model with release dates

Linear model with release dates

For all $j \in \mathcal{T}$:

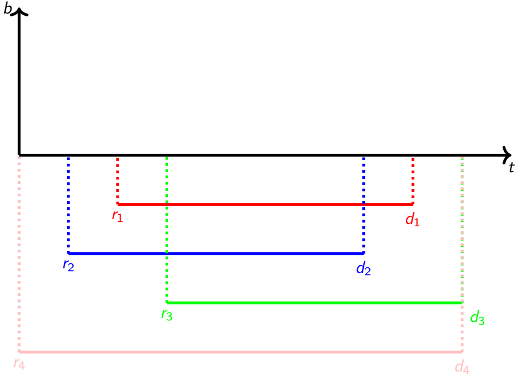
$$w_j(p) = b_j p$$

where b_j is called the growth rate of job j.



Linear model with release dates

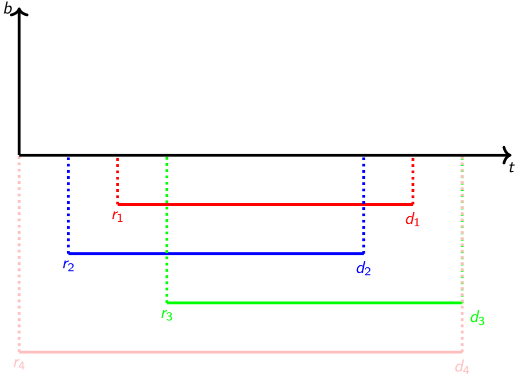
Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).



Linear model with release dates

Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).

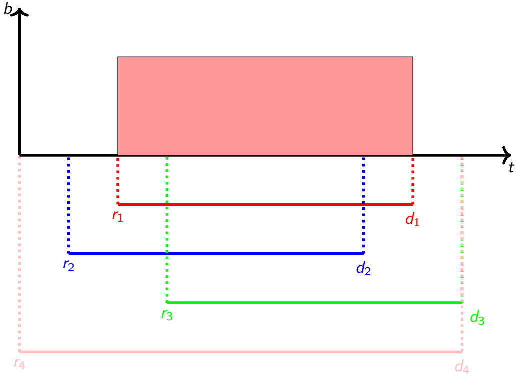
Schedule jobs in non-increasing order of their growth rates?



Linear model with release dates

Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).

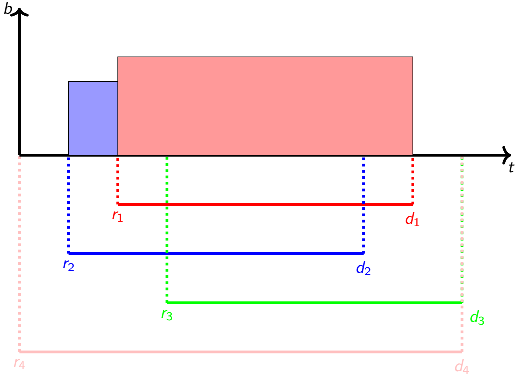
Schedule jobs in non-increasing order of their growth rates?



G-SCOP

Linear model with release dates

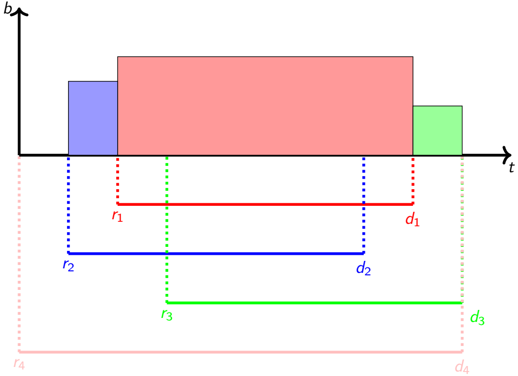
Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).
Schedule jobs in non-increasing order of their growth rates?



Linear model with release dates

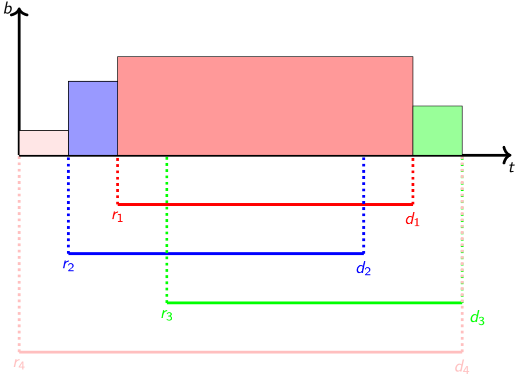
Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).

Schedule jobs in non-increasing order of their growth rates?



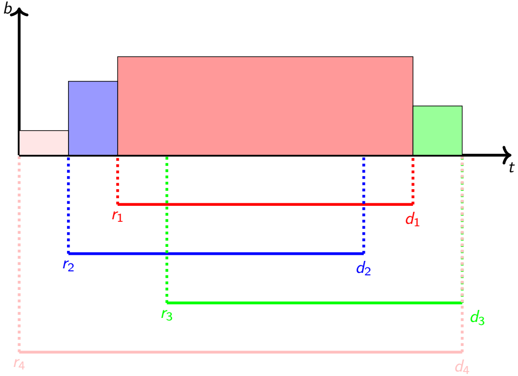
Linear model with release dates

Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarly its own).
Schedule jobs in non-increasing order of their growth rates?



Linear model with release dates

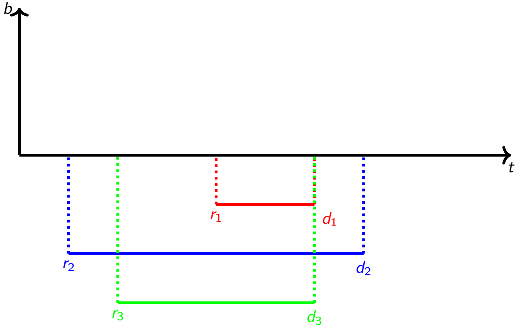
Dominance: every scheduled job starts and ends on a r_j or a d_j (but not necessarily its own).
Schedule jobs in non-increasing order of their growth rates?



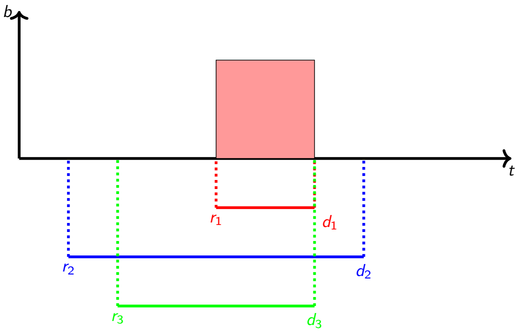
$\Longrightarrow$ Optimal solution for this instance

G-SCOP

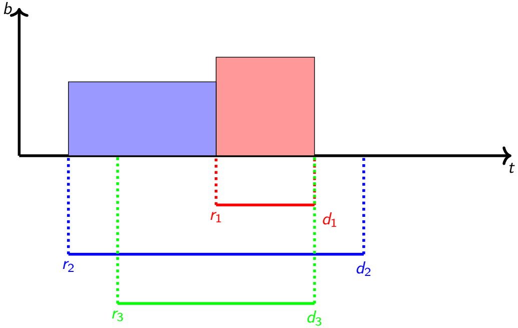
Linear model with release dates



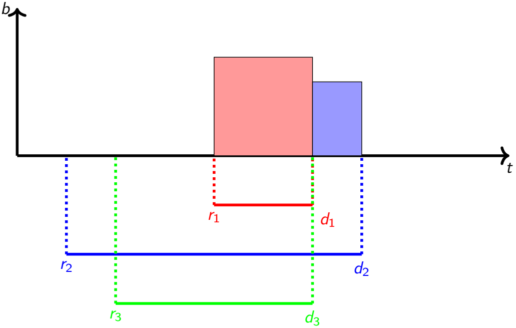
Linear model with release dates



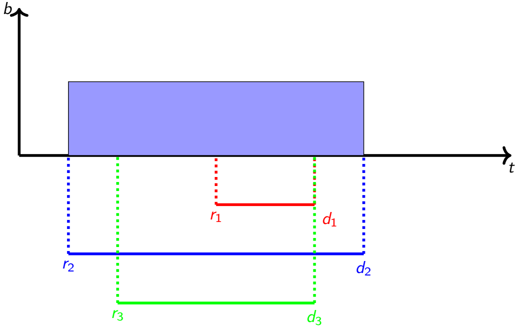
Linear model with release dates



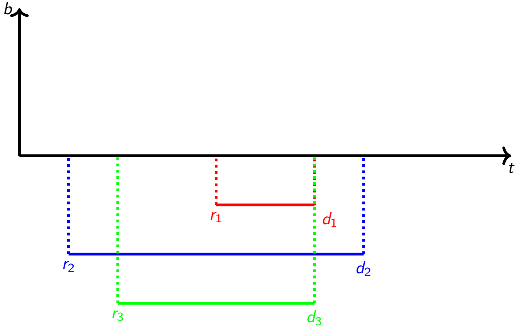
Linear model with release dates



Linear model with release dates

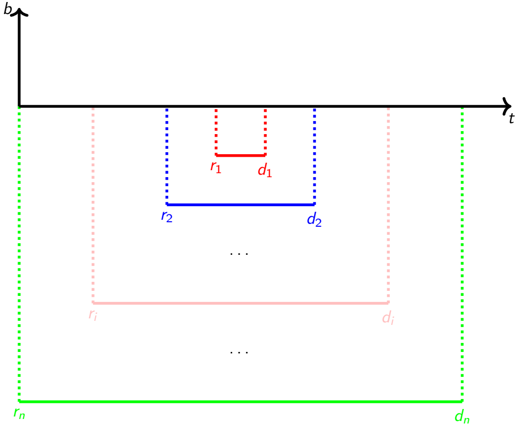


Linear model with release dates

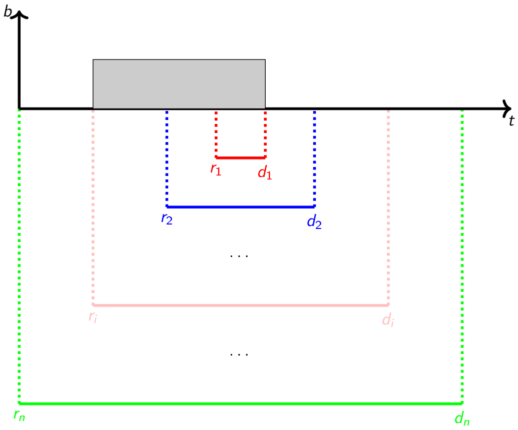


$\Longrightarrow$ a job which is the best on its time-window is either scheduled on its whole time-window, or not scheduled at all.

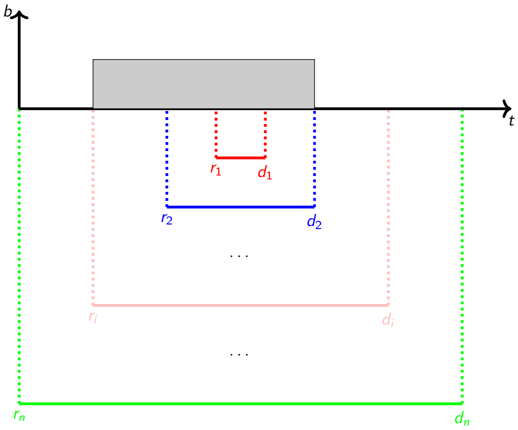
Linear model with release dates



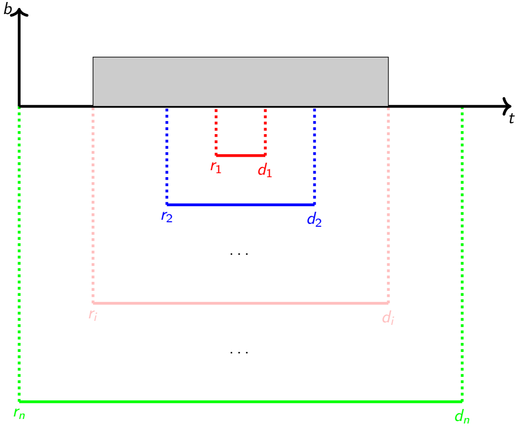
Linear model with release dates



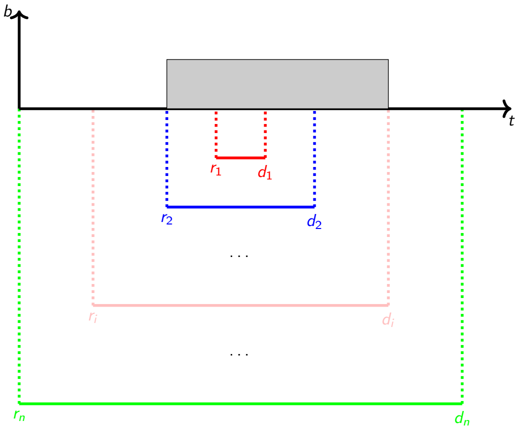
Linear model with release dates



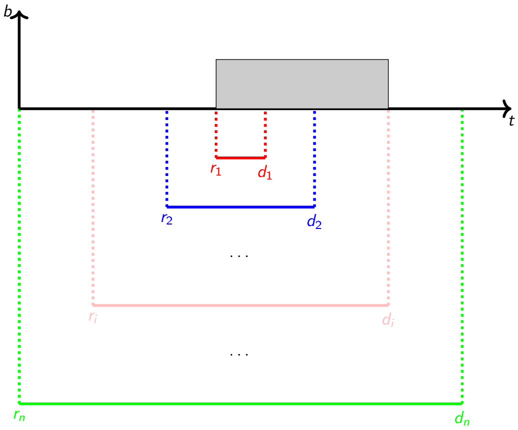
Linear model with release dates



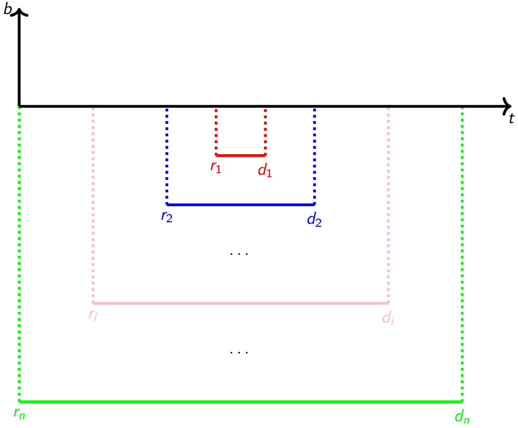
Linear model with release dates



Linear model with release dates

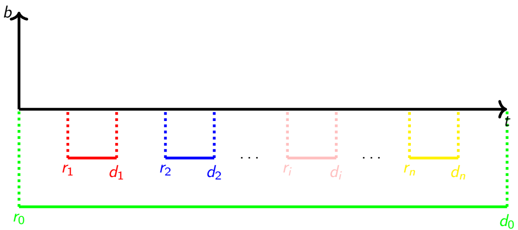


Linear model with release dates

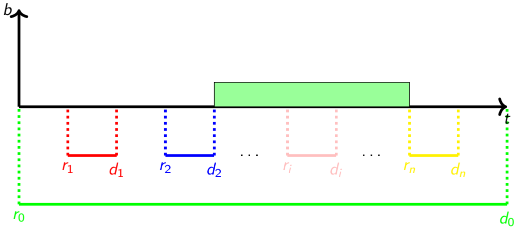


At step j, $O(j^2)$ possible states
$\Longrightarrow$ solved in polynomial time by a shortest path algorithm

Linear model with release dates

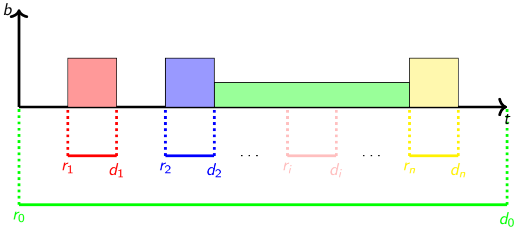


Linear model with release dates



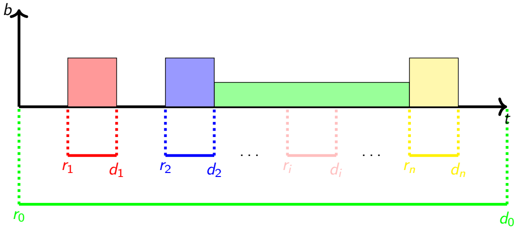
$O(n^2)$ possible schedules for job T_0.

Linear model with release dates



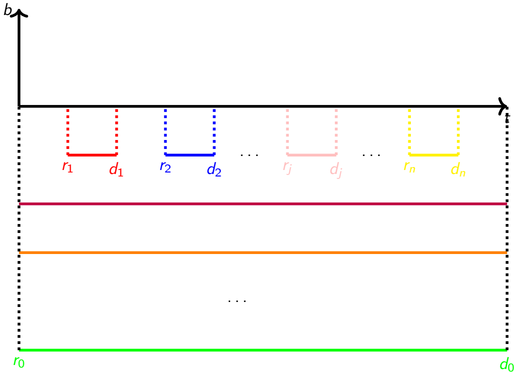
$O(n^2)$ possible schedules for job T_0.
Once the schedule of T_0 is fixed, the problem can be trivially solved in polynomial time.

Linear model with release dates



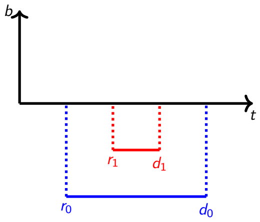
$O(n^2)$ possible schedules for job T_0.
Once the schedule of T_0 is fixed, the problem can be trivially solved in polynomial time.
$\Longrightarrow$ solved in polynomial time

Linear model with release dates



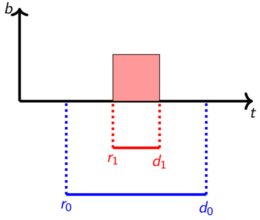
Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\implies O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)



Linear model with release dates

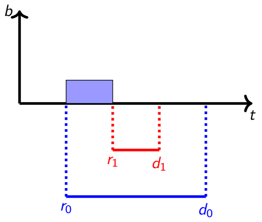
- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\implies O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)



G-SCOP

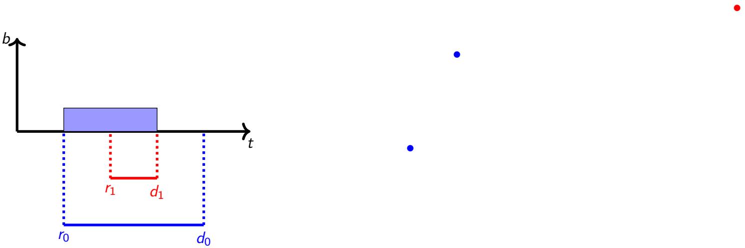
Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\implies O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)



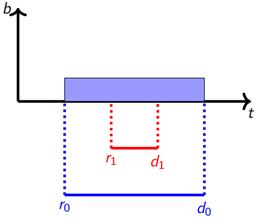
Linear model with release dates

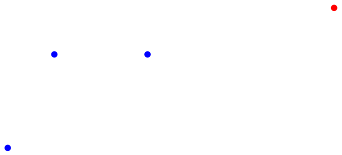
- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\implies O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)



Linear model with release dates

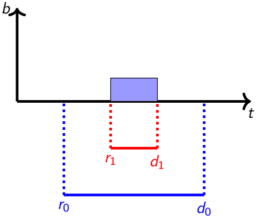
- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\Longrightarrow \ O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)

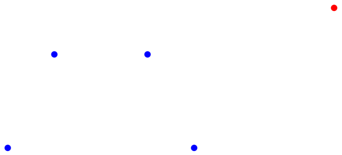




Linear model with release dates

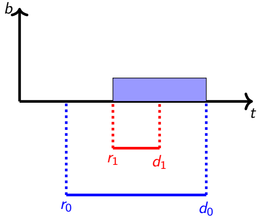
- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\implies O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)

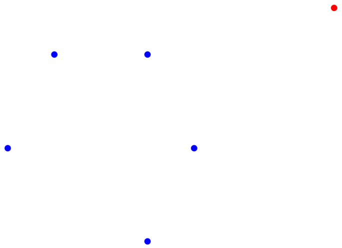




Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\Longrightarrow \quad O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)

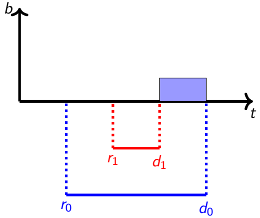


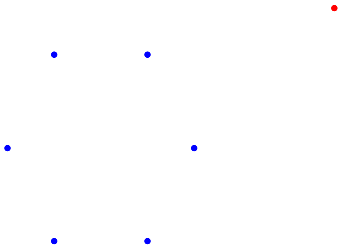


G-SCOP

Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\Longrightarrow \quad O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)

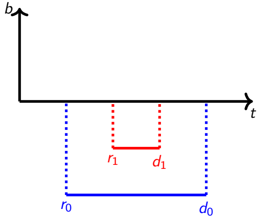


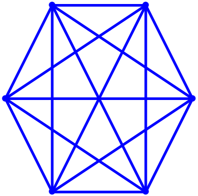


G-SCOP

Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\Longrightarrow O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)

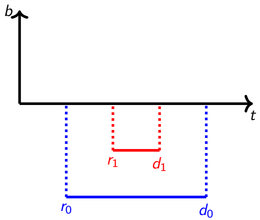


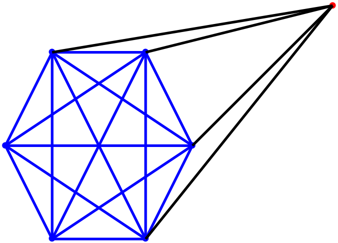


G-SCOP

Linear model with release dates

- A vertex for each possible schedule of each job with weight equal to the profit of the schedule
  $\Longrightarrow \ O(n^3)$ vertices
- An edge between two vertices iff the corresponding schedules are not compatible (same job or common instant)





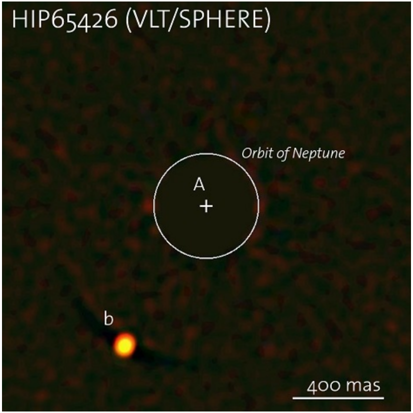
Linear model with release dates

G-SCOP

Conclusion

- Interesting class of scheduling problems
- Various resolution methods: list algorithms, linear programming, flows, classical reductions. . . (good exercices for students!)
- Some "basic" cases still open

- Integration in the practical problem
- Integration in other problem models to improve accuracy or speed up resolution.



Questions?