Notes on scheduling radio-astronomical observations

Maher Mallem[1], Frédéric Vivien[1]

[1]Inria, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342, Lyon cedex 07, France

19 March 2026

Outline

Outline

Observing from the Earth

Figure: Time lapse photography (Animalia Life)

Altitude over time

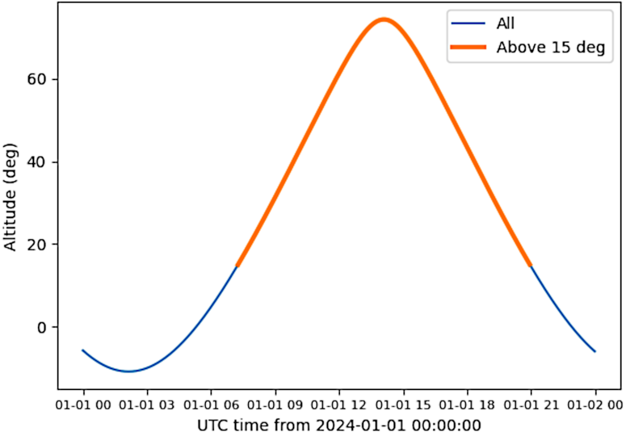


Figure: Altitude of star KELT 16 from the NenuFAR radio-telescope on 1 January 2024.

Altitude over time

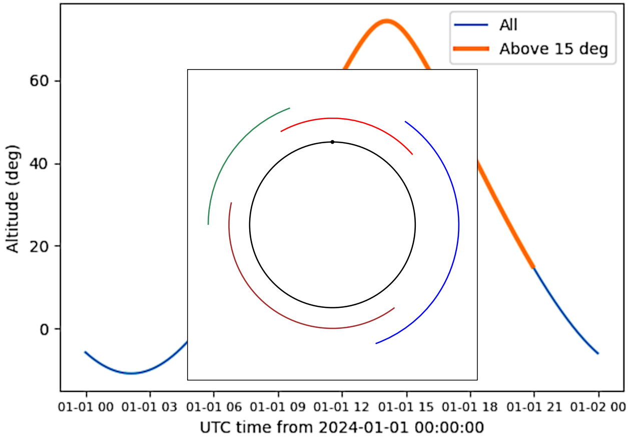


Figure: Altitude of star KELT 16 from the NenuFAR radio-telescope on 1 January 2024.

Postdoc: "Observation Scheduling"

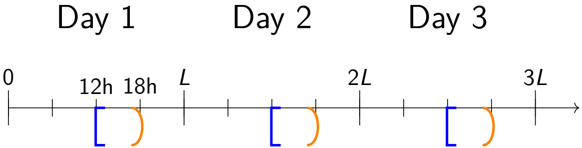
ROMA team, LIP, ENS de Lyon Nov. 2024 - Now

- *Topic:* Scheduling radio-astronomical observations.
  - Joint work with Frédéric Vivien (and Anne Benoit).
  - Partnership with the ECLAT laboratory (HPC & AI for astronomy & astrophysics).
  - Access to NenuFAR radio-telescope usage data.

Postdoc: “Observation Scheduling”

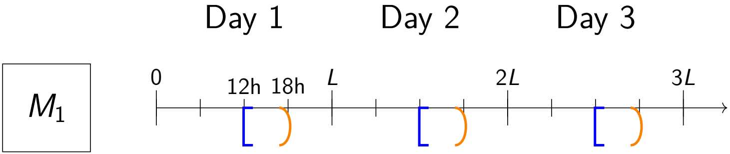
ROMA team, LIP, ENS de Lyon Nov. 2024 - Now

- *Topic:* Scheduling radio-astronomical observations.
  - Joint work with Frédéric Vivien (and Anne Benoit).
  - Partnership with the ECLAT laboratory (HPC & AI for astronomy & astrophysics).
  - Access to NenuFAR radio-telescope usage data.
- *Model:* **“Observation Scheduling”**.
  - Day length $L \in \mathbb{R}_{>0}$.
  - Processing time $p_j \in \mathbb{R}_{>0}$, daily release time $r_j \in [0, L)$ and daily deadline $d_j \in [0, L)$.

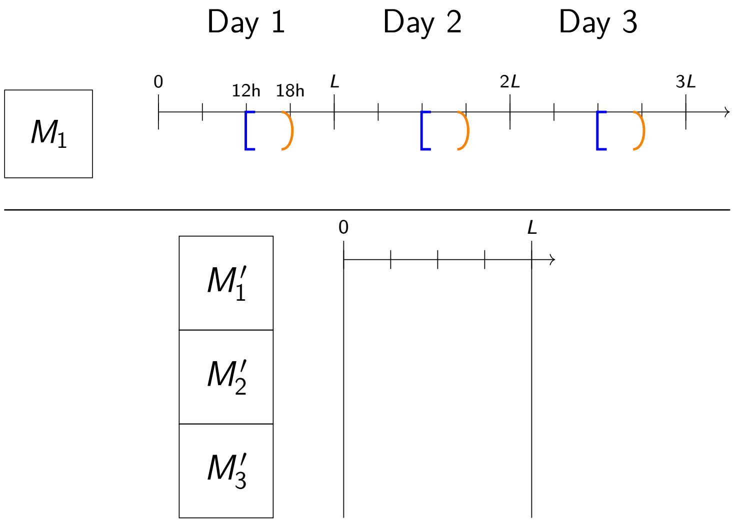


ROMA team, LIP, ENS de Lyon Nov. 2024 - Now

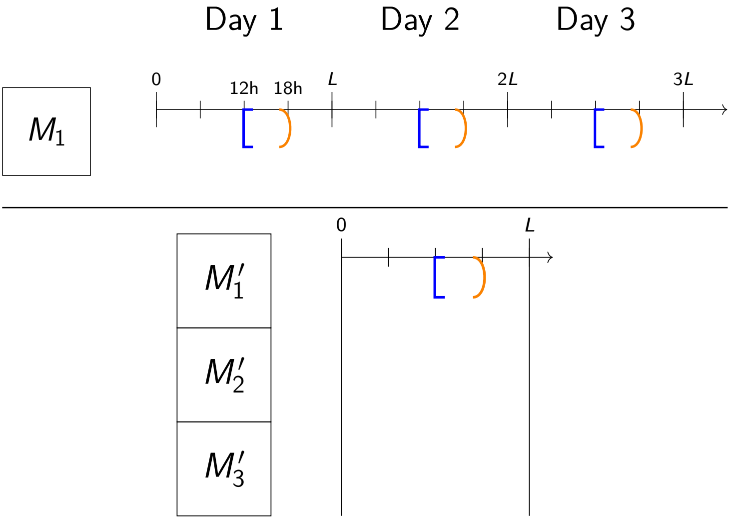
- *Topic:* Scheduling radio-astronomical observations.
  - Joint work with Frédéric Vivien (and Anne Benoit).
  - Partnership with the ECLAT laboratory (HPC & AI for astronomy & astrophysics).
  - Access to NenuFAR radio-telescope usage data.
- *Model:* **"Observation Scheduling"**.
  - Day length $L \in \mathbb{R}_{>0}$.
  - Processing time $p_j \in \mathbb{R}_{>0}$, daily release time $r_j \in [0, L)$ and daily deadline $d_j \in [0, L)$.
  - Single machine, non-preemptive, minimizing idle time (i.e., makespan).



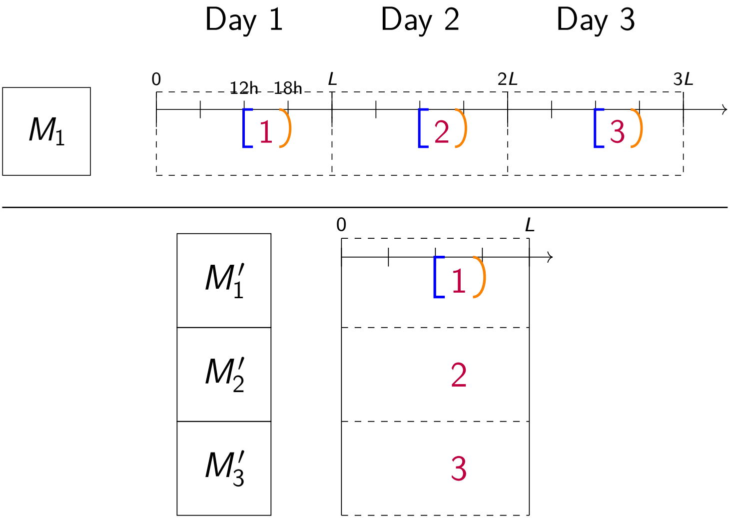
Postdoc: "Observation Scheduling" (cont.)



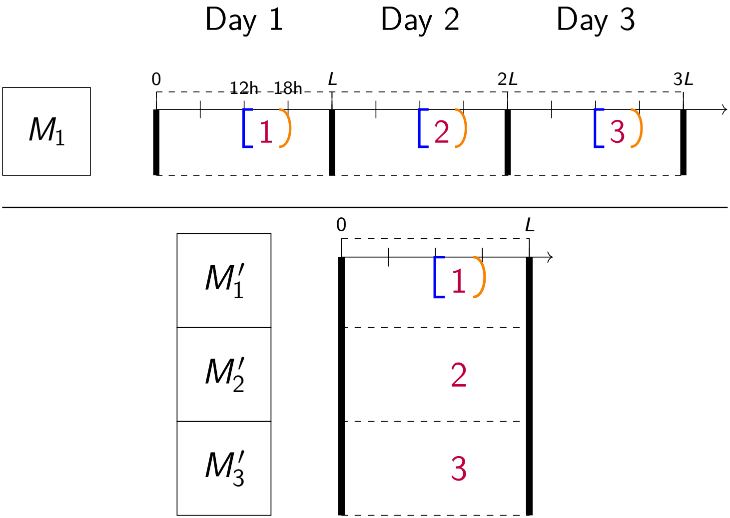
Postdoc: "Observation Scheduling" (cont.)



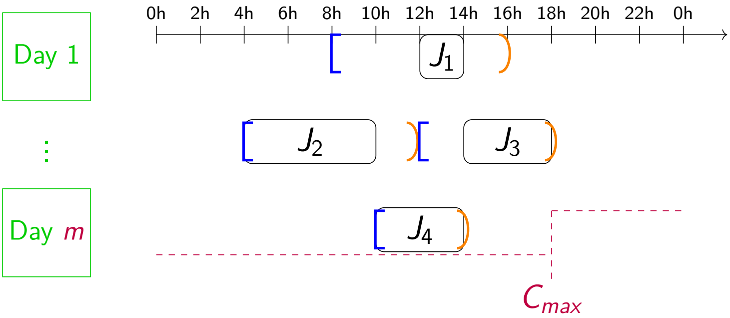
Postdoc: "Observation Scheduling" (cont.)

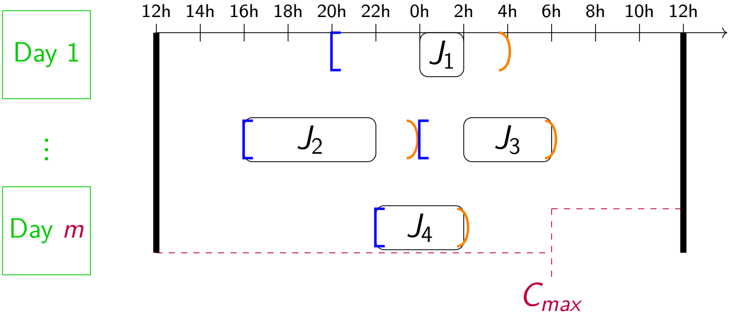


Postdoc: "Observation Scheduling" (cont.)

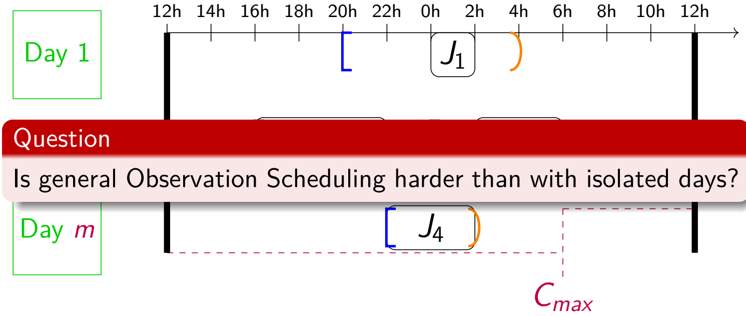


Scheduling observations





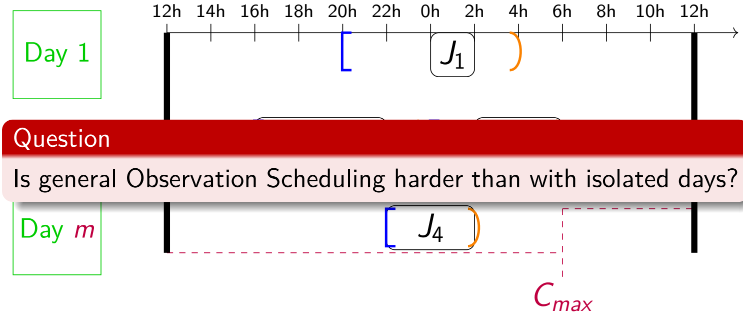
With isolated days: $P|r_j, d_j|\min(m)$.



Question

Is general Observation Scheduling harder than with isolated days?

With isolated days: $P|r_j, d_j|\min(m)$.



Question

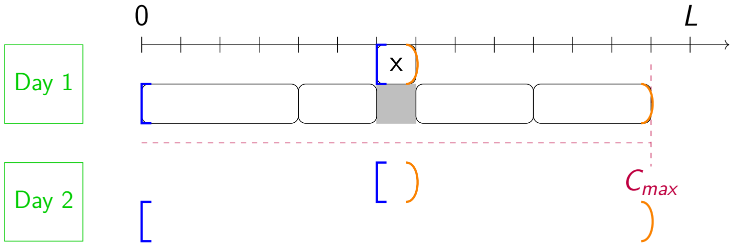
Is general Observation Scheduling harder than with isolated days?

With isolated days: $P|r_j, d_j|\mathrm{min}(m)$.

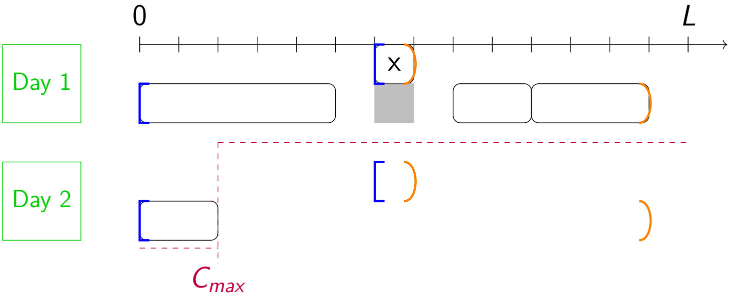
One day: strongly NP-complete [Lenstra et al., 1977].

Outline

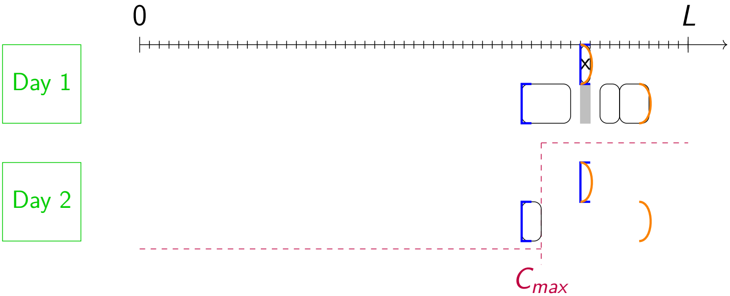
$(2-\varepsilon)$-inapproximability



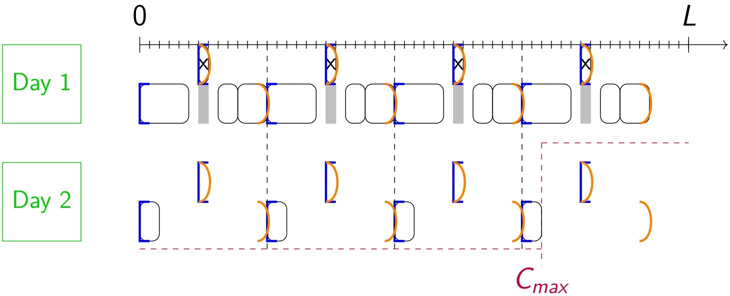
$(2-\varepsilon)$-inapproximability



$(2 - \varepsilon)$-inapproximability



$(2-\varepsilon)$-inapproximability



Observation Scheduling with isolated days.

- Goal: minimize the number of days.
  - $P|r_j, d_j|\min(m)$.

Observation Scheduling with isolated days.

- Goal: minimize the number of days.
  - $P|r_j, d_j|\min(m)$.
- $\mathcal{O}(\log(n))$-approximation algorithm. [Cieliebak et al., 2004]

Observation Scheduling with isolated days.

- Goal: minimize the number of days.
  - $P|r_j, d_j|\min(m)$.
- $\mathcal{O}(\log(n))$-approximation algorithm. [Cieliebak et al., 2004]

Job Interval Selection problem.

- Goal: maximize the number of scheduled jobs.

Observation Scheduling with isolated days.

- Goal: minimize the number of days.
  - $P|r_j, d_j|\min(m)$.
- $\mathcal{O}(\log(n))$-approximation algorithm. [Cieliebak et al., 2004]

Job Interval Selection problem.

- Goal: maximize the number of scheduled jobs.
- 1/2-approximation: "Earliest Completion Time". [Spieksma, 1999]

$\mathcal{O}(\log(n))$-approximation (cont.)

Remark

W.l.o.g: $\forall j, p_j \leq L$.

Proposition

If $L \leq OPT$, then any f-approximation algorithm with isolated days is a $[(2 + L/OPT) \cdot f]$-approximation algorithm for Observation Scheduling.

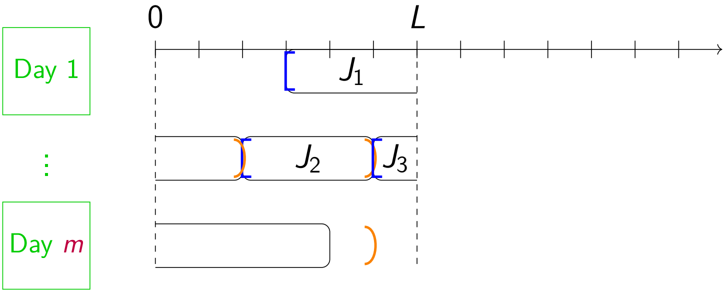
$\mathcal{O}(\log(n))$-approximation (cont.)

Remark

W.l.o.g: $\forall j, p_j \leq L$.

Proposition

If $L \leq OPT$, then any f-approximation algorithm with isolated days is a $[(2 + L/OPT) \cdot f]$-approximation algorithm for Observation Scheduling.



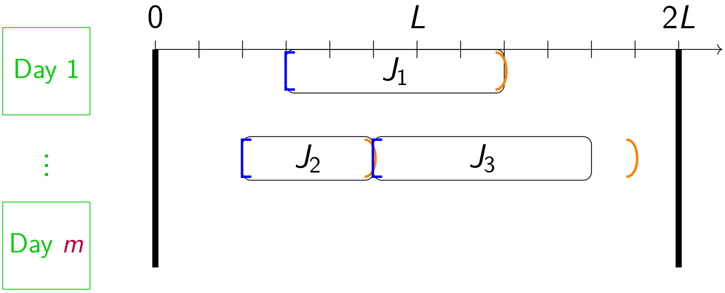
$\mathcal{O}(\log(n))$-approximation (cont.)

Remark

W.l.o.g: $\forall j, p_j \leq L$.

Proposition

If $L \leq OPT$, then any f-approximation algorithm with isolated days is a $[(2 + L/OPT) \cdot f]$-approximation algorithm for Observation Scheduling.



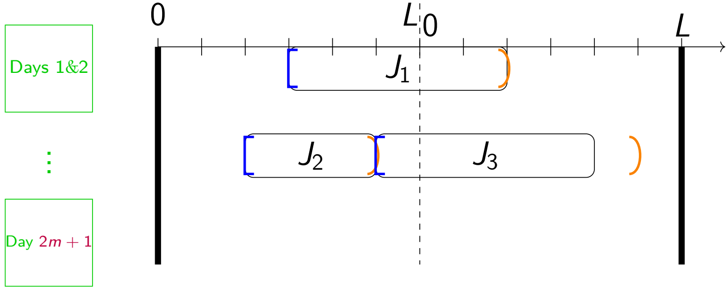
$\mathcal{O}(\log(n))$-approximation (cont.)

Remark

W.l.o.g: $\forall j, p_j \leq L$.

Proposition

If $L \leq OPT$, then any f-approximation algorithm with isolated days is a $[(2 + L/OPT) \cdot f]$-approximation algorithm for Observation Scheduling.



$\mathcal{O}(\log(n))$-approximation (cont.)

Remark

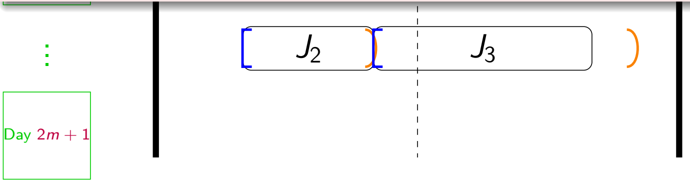
W.l.o.g: $\forall j, p_j \leq L$.

Proposition

If $L \leq OPT$, then any f-approximation algorithm with isolated days is a $[(2 + L/OPT) \cdot f]$-approximation algorithm for Observation Scheduling.

Question

Constant approximation algorithm?



Outline

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
- P: deterministic time $poly(|\mathcal{I}|)$.
- NP: nondeterministic time $poly(|\mathcal{I}|)$.

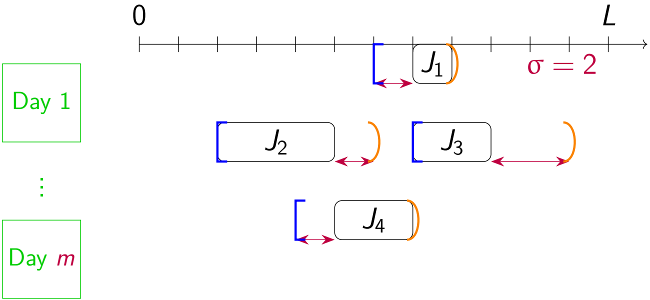
Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k}$, instance $\mathcal{I}$ with parameter value $\leq k$.
- 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.

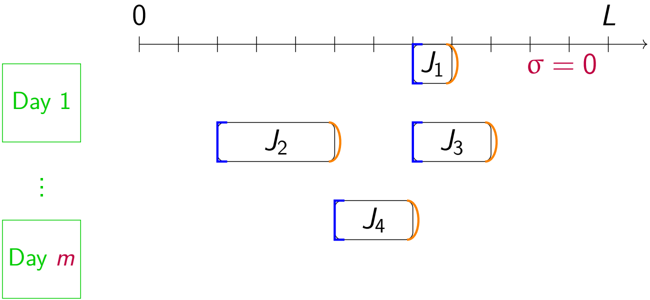
Going beyond NP-hardness

- Problem $\mathcal{P}$, instance $\mathcal{I}$.
  - Parameter $\bar{k}$, instance $\mathcal{I}$ with parameter value $\leq k$.
- 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.

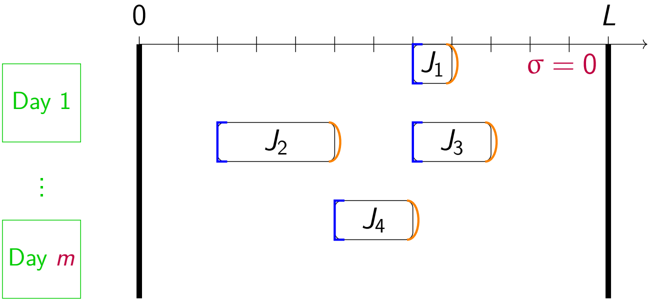
Scheduling with tight windows



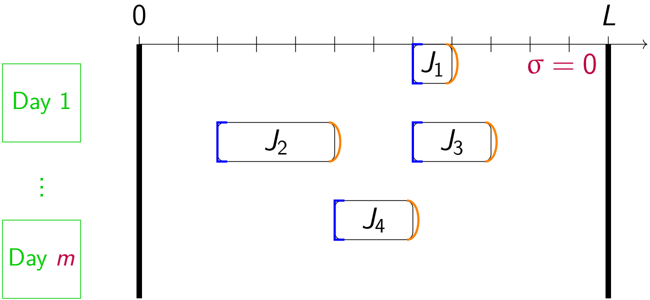
Scheduling with tight windows



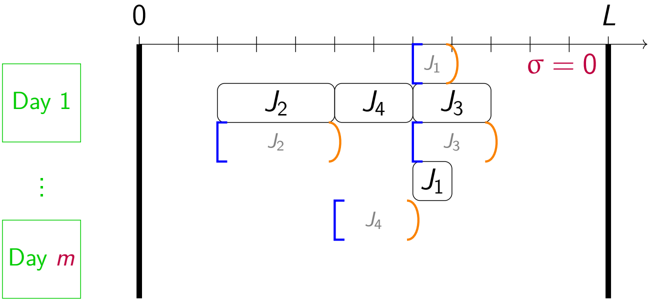
Scheduling with tight windows



Scheduling with tight windows

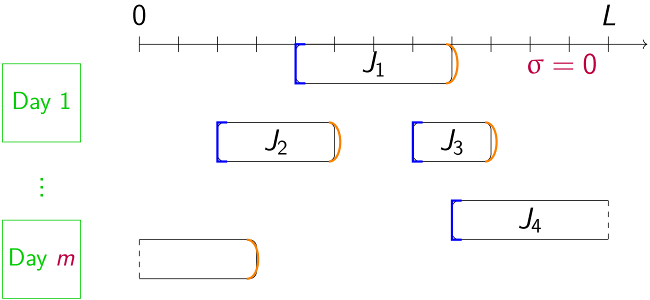


Interval graph coloring: $\mathcal{O}(n \cdot \log(n))$ time.



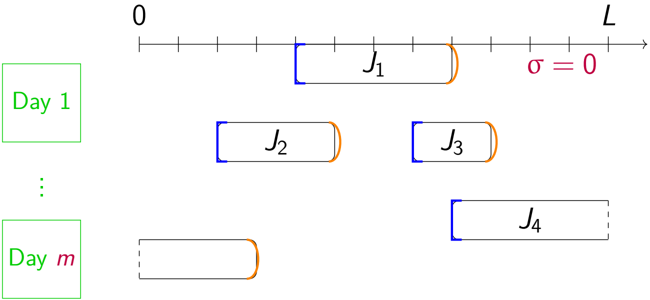
Interval graph coloring: $\mathcal{O}(n \cdot \log(n))$ time.

Scheduling with tight windows



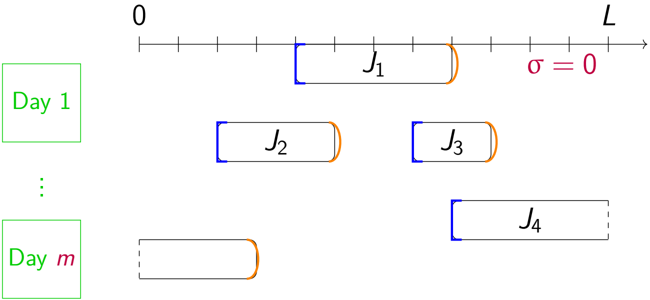
Interval graph coloring: $\mathcal{O}(n \cdot \log(n))$ time.

Scheduling with tight windows

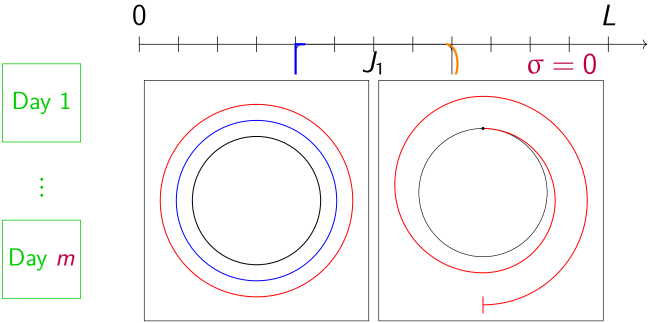


Circular-arc graph coloring: NP-complete [Garey et al., 1980].

Scheduling with tight windows

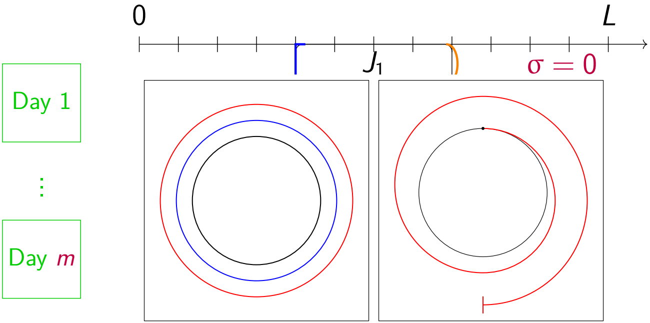


~~Circular-arc graph coloring: NP-complete [Garey et al., 1980]~~.



~~Circular-arc graph coloring: NP-complete [Garey et al., 1980]~~.

Scheduling with tight windows



~~Circular-arc graph coloring: NP-complete [Garey et al., 1980]~~.

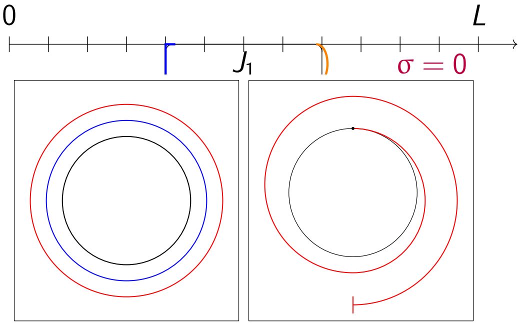
Multi-slot Just-in-time: $\mathcal{O}(n\log(n)^2)$ time [Dereniowski and Kubiak, 2010].

Scheduling with tight windows

Day 1

⋮

Day m



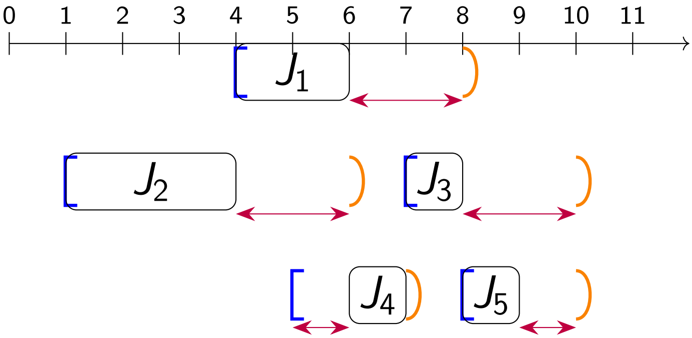
~~Circular-arc graph coloring: NP-complete [Garey et al., 1980]~~.

Multi-slot Just-in-time: $\mathcal{O}(n\log(n)^2)$ time [Dereniowski and Kubiak, 2010].

→ TSP with a Gilmore-Gomory distance matrix.

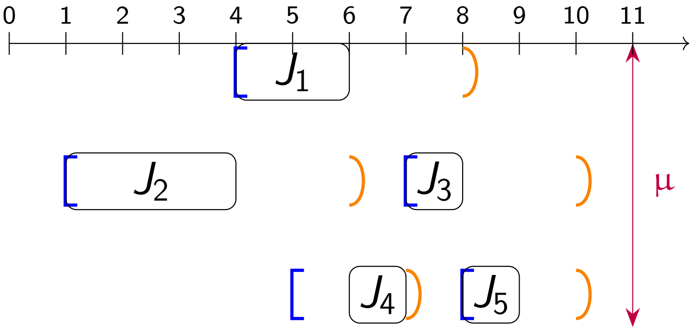
[Gilmore and Gomory, 1964, Vairaktarakis, 2003]

Width μ



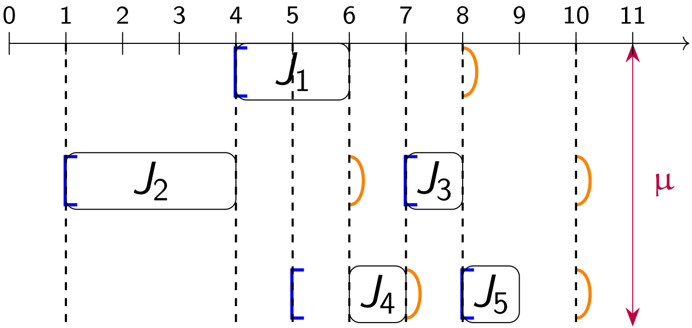
$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

Width μ



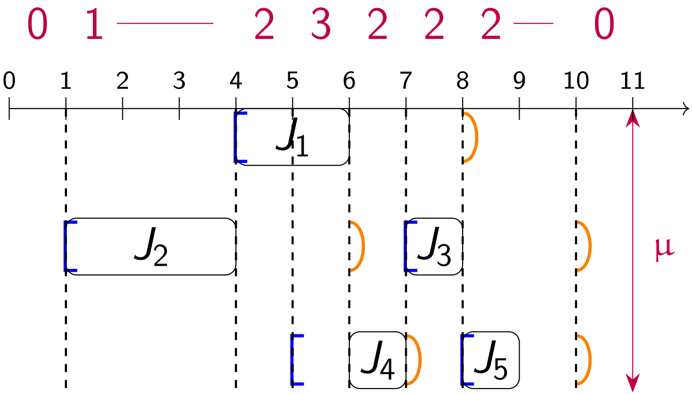
$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

Width μ



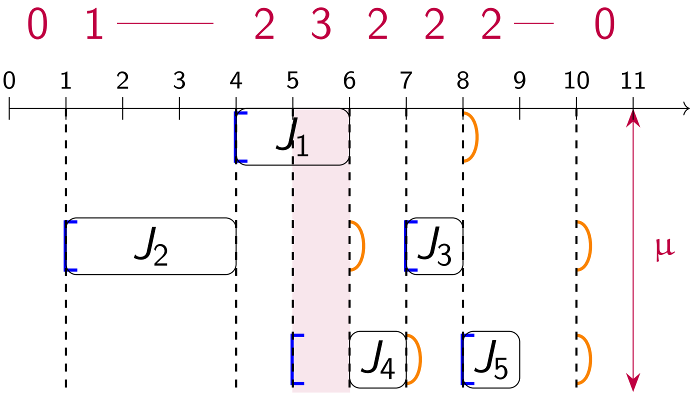
$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

Width μ



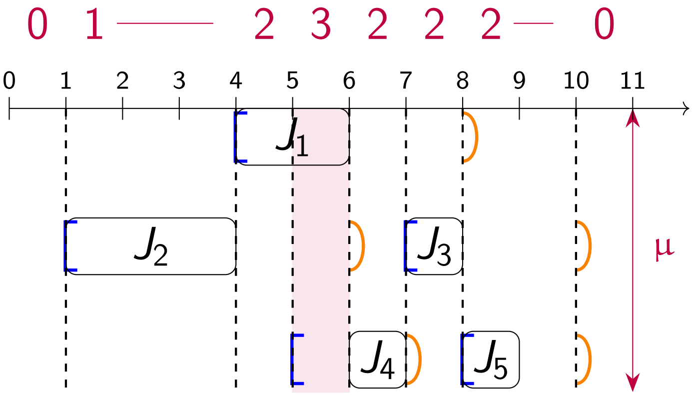
$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

Width μ



$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

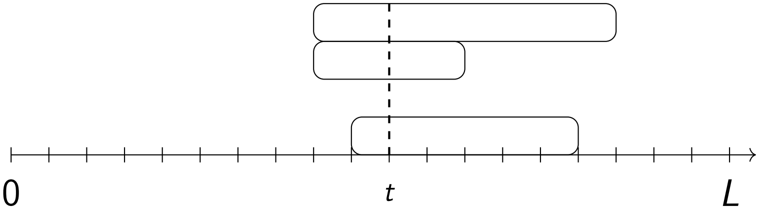
Width μ



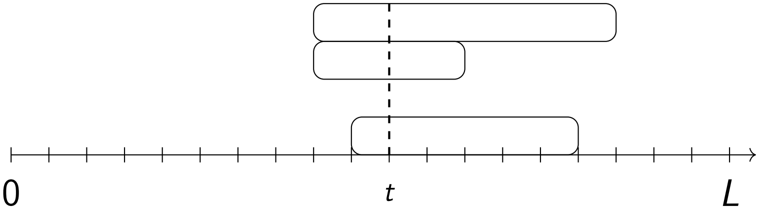
$\sigma = 2$: NP-complete [Cieliebak et al., 2004].

$\mu = 4$: NP-complete [Hanen and Munier Kordon, 2023].

FPT membership with $(\mu + \sigma)$: border schedules

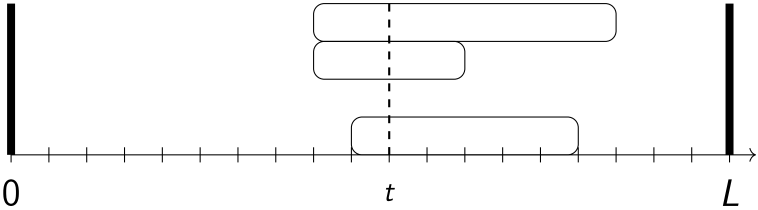


FPT membership with $(\mu + \sigma)$: border schedules



- $\mathcal{O}([\mu(\sigma+1)+1]^m)$ border schedules associated to t mod L.

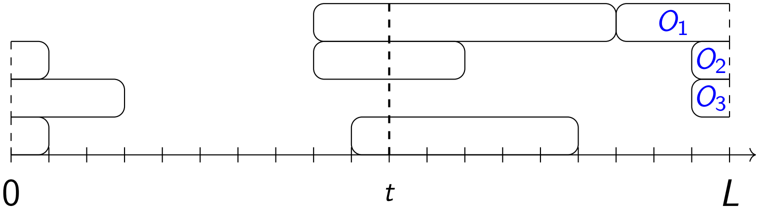
FPT membership with $(\mu + \sigma)$: border schedules



- $\mathcal{O}([\mu(\sigma+1)+1]^m)$ border schedules associated to t mod L.
- Isolated days: in FPT$(\mu + \sigma)$.

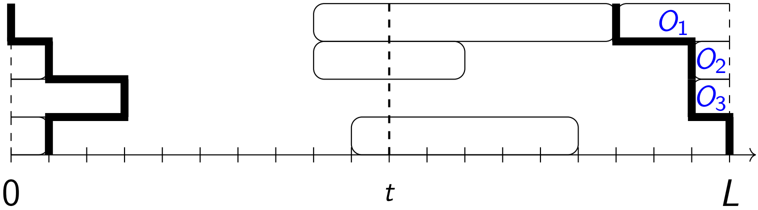
  [Cieliebak et al., 2004, Hanen and Munier Kordon, 2023]

FPT membership with $(\mu + \sigma)$: border schedules



- $\mathcal{O}([\mu(\sigma+1)+1]^m)$ border schedules associated to t mod L.
- Isolated days: in FPT$(\mu+\sigma)$.
  [Cieliebak et al., 2004, Hanen and Munier Kordon, 2023]

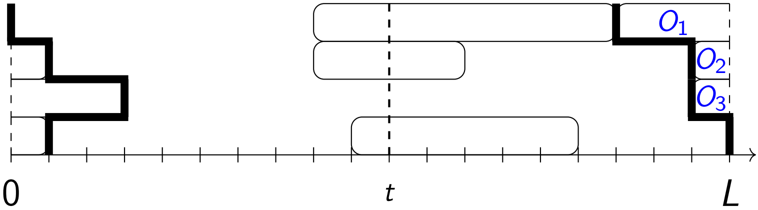
FPT membership with $(\mu + \sigma)$: border schedules



- $\mathcal{O}([\mu(\sigma + 1) + 1]^m)$ border schedules associated to t mod L.
- Isolated days: in FPT$(\mu + \sigma)$.

  [Cieliebak et al., 2004, Hanen and Munier Kordon, 2023]

FPT membership with $(\mu + \sigma)$: border schedules



- $\mathcal{O}([\mu(\sigma + 1) + 1]^m)$ border schedules associated to t mod L.
- Isolated days: in FPT$(\mu + \sigma)$.
  [Cieliebak et al., 2004, Hanen and Munier Kordon, 2023]

Idea

For each of the $\mathcal{O}([\mu(\sigma + 1) + 1]^{m-1})$ border schedules associated to 0 mod L, solve an instance of "$P|r_j, d_j|\min(m)$ with machine availabilities".

Outline

Results: same as with isolated days

- $\mathcal{O}(\log(n))$-approximation algorithm.
- No $(2-\varepsilon)$-approximation algorithm (unless P = NP).
- Fixed-parameter tractability: σ, μ and $(m+\sigma)$.

Conclusion

Results: same as with isolated days

- $\mathcal{O}(\log(n))$-approximation algorithm.
- No $(2-\varepsilon)$-approximation algorithm (unless P = NP).
- Fixed-parameter tractability: σ, μ and $(m+\sigma)$.

Future work

- Constant approximation algorithm?
- Improve the fixed-parameter algorithms.
  - $\mathcal{O}([\mu(\sigma+1)+1]^{3m-1} \cdot 2^{3\mu} \cdot m^{\mu} \cdot \mu \cdot n + n \cdot \log(n))$ time.
- “Earliest Start Time” performance guarantee?
- Refine the model: unary/fixed L; limited preemption.

Conclusion

Results: same as with isolated days

- $\mathcal{O}(\log(n))$-approximation algorithm.
- No $(2-\varepsilon)$-approximation algorithm (unless P = NP).
- Fixed-parameter tractability: σ, μ and $(m+\sigma)$.

Future work

- Constant approximation algorithm?
- Improve the fixed-parameter algorithms.
  - $\mathcal{O}([\mu(\sigma+1)+1]^{3m-1} \cdot 2^{3\mu} \cdot m^{\mu} \cdot \mu \cdot n + n \cdot \log(n))$ time.
- "Earliest Start Time" performance guarantee?
- Refine the model: unary/fixed L; limited preemption.
- **Other applications?**

References I

Cieliebak, M., Erlebach, T., Hennecke, F., Weber, B., and Widmayer, P. (2004).
Scheduling with release times and deadlines on a minimum number of machines.
In Exploring New Frontiers of Theoretical Informatics, pages 209–222, Boston, MA. Springer.

Dereniowski, D. and Kubiak, W. (2010).
Makespan minimization of multi-slot just-in-time scheduling on single and parallel machines.
Journal of Scheduling, 13:479–492.

Garey, M. R., Johnson, D. S., Miller, G. L., and Papadimitriou, C. H. (1980).
The complexity of coloring circular arcs and chords.
SIAM Journal on Algebraic Discrete Methods, 1(2):216–227.

Gilmore, P. C. and Gomory, R. E. (1964).
Sequencing a one state-variable machine: A solvable case of the traveling salesman problem.
Operations research, 12(5):655–679.

References II

Hanen, C. and Munier Kordon, A. (2023).
Fixed-parameter tractability of scheduling dependent typed tasks subject to release times and deadlines.
Journal of Scheduling, pages 1–15.

Lenstra, J., Rinnooy Kan, A., and Brucker, P. (1977).
Complexity of machine scheduling problems.
Ann. of Discrete Math., 1:343–362.

Spieksma, F. C. (1999).
On the approximability of an interval scheduling problem.
Journal of Scheduling, 2(5):215–227.

Vairaktarakis, G. L. (2003).
Simple algorithms for gilmore–gomory's traveling salesman and related problems.
Journal of Scheduling, 6(6):499–520.