Non-preemptively scheduling interval-constrained jobs: few machines, small looseness, and small slack

René van Bevern[1,2]

joint work with

Rolf Niedermeier[3] and Ondřej Suchý[4]

[1]Novosibirsk State University [2]Sobolev Institute of Mathematics

[3]TU Berlin [4]Czech Technical University in Prague

DOOR 2016, September 19–23, 2016, Vladivostok, Russian Federation

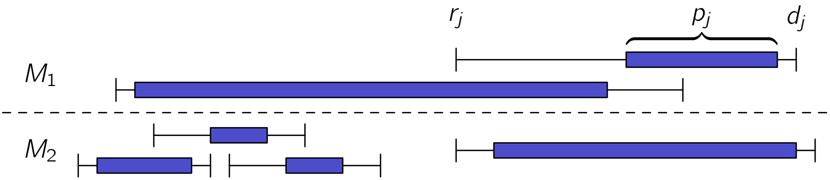
1 Introduction

Problem 1.1 (Interval-Constrained Scheduling (ICS)).

Input: m parallel identical machines, n jobs, each job j with a *release time* $r_j \in \mathbb{N}$, a *deadline* $d_j \in \mathbb{N}$, and a *processing time* $p_j \in \mathbb{N}$.

Question: Can we process each job j for p_j consecutive time units such that

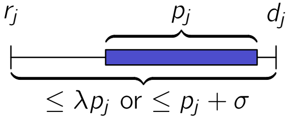
1. each job is processed only on one machine,
2. each machine processes at most one job at a time, and
3. each job j starts no earlier than r_j and is finished by d_j.



1.1 Looseness and slack

Study influence of the following constraints for various σ and λ:

- $|d_j - r_j| \leq \lambda p_j$ (looseness λ) and
- $|d_j - r_j| \leq p_j + \sigma$ (slack σ).



Why is this interesting?

- If $|d_j - r_j| = p_j$ for all $j \rightsquigarrow$ m-coloring interval graphs: $O(n \log n)$ time.
- If $r_j = 0$ and $d_j = D$ for all $j \rightsquigarrow$ Bin Packing: NP-complete but trivial if $|d_j - r_j| < 2p_j$ for all j: each item needs its own bin.

Cieliebak, Erlebach, Hennecke, Weber, and Widmayer, 2004:

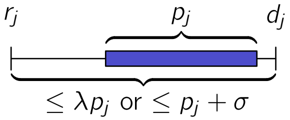
- NP-complete for any constant $\lambda > 1$ if σ and m are unbounded,
- NP-complete for any constant $\sigma \geq 2$ if λ and m are unbounded.

1.2 Our results

van Bevern, Niedermeier, Suchý, Journal of Scheduling, 2016:

Reconsider problem with small number m of machines:

1. Remains weakly NP-hard for $m = 2$ machines and any constant $\lambda > 1$.
2. There is not even a $f(m) \cdot \text{poly}(n)$-time algorithm for any constant $\lambda > 1$, unless FPT = W[1], even if all processing times are polynomial.
3. Pseudo-polynomial time if both m and λ are constant.
4. $O(n \log n)$-time if both m and σ are constant $\rightsquigarrow$ quasi-linear!



2 Small slack σ and few machines m

Definition 2.1. Consider instance *height* $h := \max_{t\in\mathbb{N}} |S_t|$, where

$$S_t := \{j \in J \mid t \in [r_j, d_j)\}$$

are the jobs whose time window contains time t.

Lemma 2.2. If all jobs can be processed on m machines, then $h \leq (2\sigma+1)m$.

If our input instance has $h > (2\sigma + 1)m$, then we say no. Otherwise:

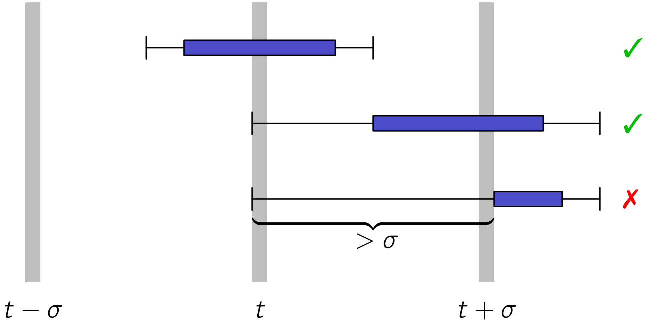
Proposition 2.3 (Cieliebak et al. (2004))**.** ICS is solvable in $O(n \cdot (\sigma + 1)^h \cdot h \log h)$-time.

Theorem 2.4. ICS is solvable in $O((\sigma + 2)^{(2\sigma+1)m} \cdot n + n \log n)$ time.

Lemma 2.2. If all jobs can be processed on m machines, then $h \leq (2\sigma+1)m$.

Proof. For any time t, consider set S_t of jobs whose time window contains t.

- ▷ There are $(2\sigma+1)m$ time units on m machines in interval $[t-\sigma, t+\sigma]$.
- ▷ Each job in S_t uses at least one of these units $\rightsquigarrow |S_t| \leq (2\sigma+1)m$.



□

3 Pseudo-polynomial time for fixed m and λ

Lemma 3.1. If jobs can be processed on m machines, then $h \in O(m\lambda \log \ell)$, where $\ell = \max_{j\in J} |d_j - t_j|$ is the length of the longest time window.

If our input instance has too large height h, then we say no.

Otherwise, use new dynamic programming algorithm:

Lemma 3.2. ICS is solvable in $O(2^h \cdot (2\ell + 2)^m \cdot n\ell + n \log n)$ time.

Why we need a new dynamic programming algorithm:

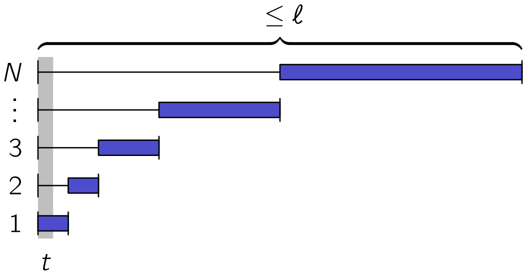
- ▷ The dependence on h in Cieliebak et al. (2004) is $(\sigma + 1)^h$,
- ▷ we have $2^h \rightsquigarrow$ pseudo-poly for $h \in O(m\lambda \log \ell)$ and constant λm.

Theorem 3.3. ICS can be solved in $\ell^{O(\lambda m)} n + O(n \log n)$ time.

Lemma 3.1. If all jobs can be processed on m machines, then $h \in O(m\lambda \log \ell)$.

Proof sketch for $\lambda = 2$ and $m = 1$. $\rightsquigarrow$ show $h \in O(\log \ell)$.

- Consider set S_t of jobs whose time window contains time t.
- Number N of jobs of S_t that can be processed at some time $t' \geq t$:



Observe: $\ell \geq |d_N - r_N| \sim 2^N \quad \rightsquigarrow \quad N \sim \log \ell$.

- Same number N of jobs can be processed at some time $t' \leq t$. □

4 Conclusion

We have seen: if the number m of machines is constant, then ICS is

- ▷ solvable in quasi-linear time, i. e. $O(n \log n)$ time, for constant slack σ,
- ▷ solvable in pseudo-polynomial time for constant looseness λ.

These results are tight in the following sense:

- ▷ constant σ or λ leave ICS strongly NP-hard (Cieliebak et al., 2004),
- ▷ $m = 1$ leaves ICS strongly NP-hard (Garey and Johnson, 1979),
- ▷ $f(m) \cdot \text{poly}(n)$ is impossible for constant λ, unless $\text{FPT} = \text{W}[1]$.