On the Fine-grained Parameterized Complixity of Partial Scheduling

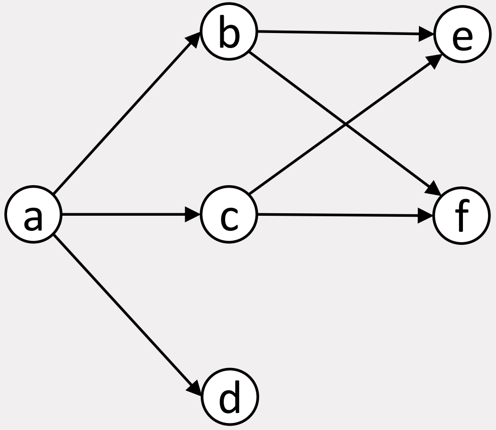
Céline Swennenhuis & Jesper Nederlof

Department of Mathematics and Computer Science, Eindhoven University of Technology

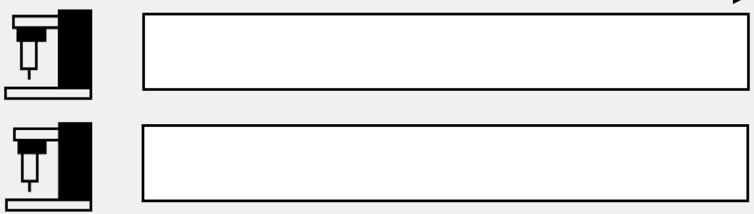
Problem Definition

- Number of machines
- Processing times for n jobs
- Precedence Graph G
- k = minimal number of jobs to schedule
- Makespan / Universal Deadline D

Job	p_j
a	1
b	2
c	4
d	5
e	6
f	3



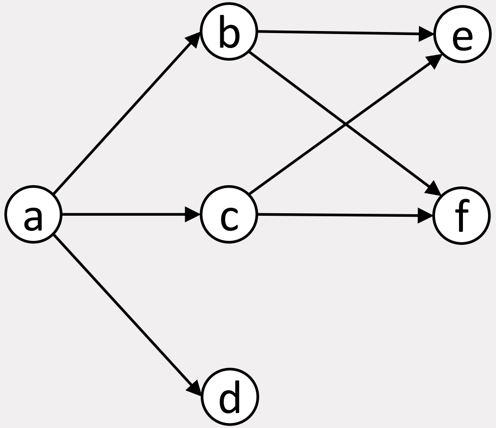
$k = 5$



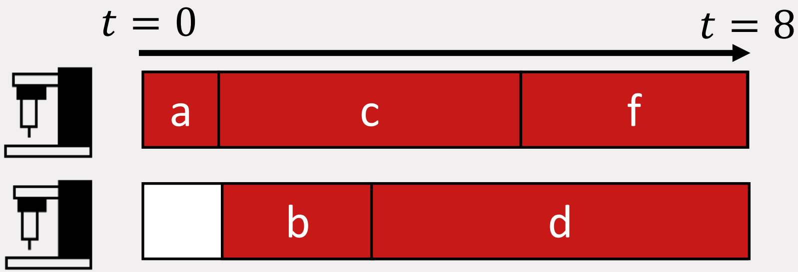
Problem Definition

- Number of machines
- Processing times for n jobs
- Precedence Graph G
- k = minimal number of jobs to schedule
- Makespan / Universal Deadline D

Job	p_j
a	1
b	2
c	4
d	5
e	6
f	3



$k = 5$



Motivation

More jobs available than we want to process

- Close-horizon approach
- Other jobs can be outsourced

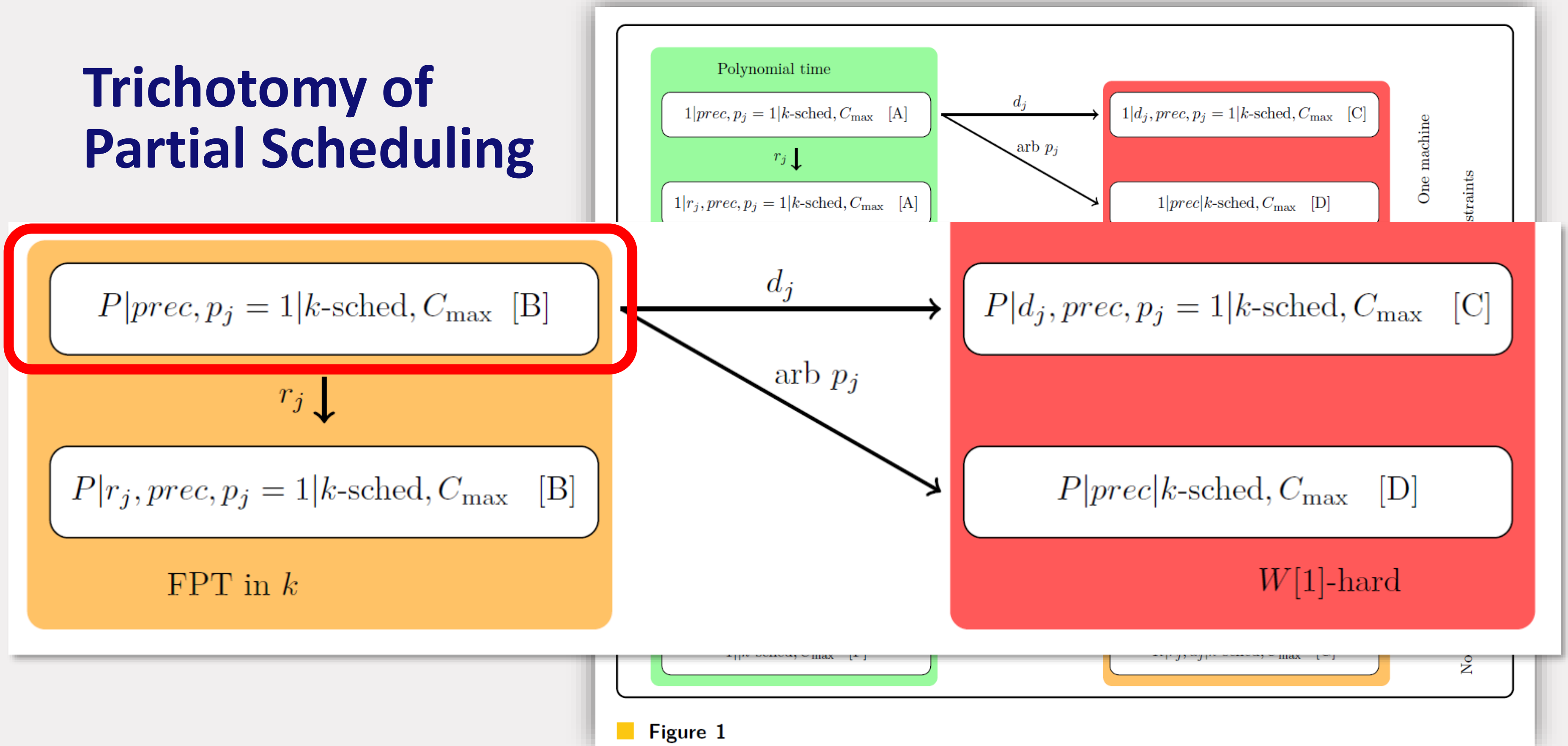
Parameterized Complexity

If $k = n$, i.e. we want to schedule all jobs, the problem is NP-hard*.

Can we do better if k is small?

*Except for some special cases

Trichotomy of Partial Scheduling

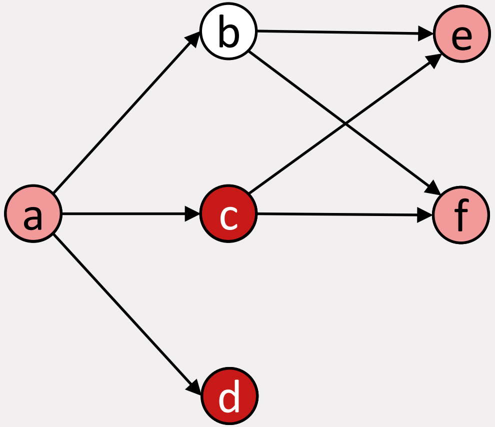


Today:

Theorem: The problem $P|prec, p_j = 1|k - sched, C_{max}$ is fixed-parameter tractable (FPT) in k.

To do so, we will give an algorithm which runs in time $O(8^k \cdot poly(n))$.

Definitions



Precendence Constraints Graph G

Notice:

- G is acyclic
- $i < j$ if $i \rightarrow j$

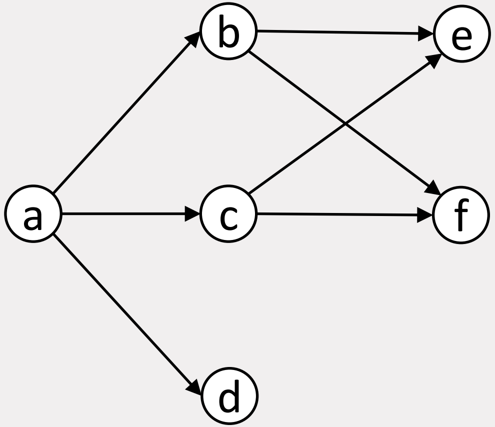
Def: Let A be a set of jobs.
pred$(A) = \{x \mid \exists\, a \in A\ s.t.\ x \leq a\}$
comp$(A) = \{x \mid \exists\, a \in A\ s.t.\ x \leq a$ or $x \geq a\}$

Let $A = \{c, d\}$, then:

- pred$(A) = \{a, c, d\}$
- comp$(A) = \{a, c, d, e, f\}$

Definitions

Def: An *antichain* is a set A whose elements are pairwise incomparable.



Precendence Constraints Graph G

Ex. of antichains in G	Corresponding pred(A)
✓ $\{b, c, d\}$	$\Longrightarrow \{a, b, c, d\}$
✓ $\{b, c\}$	$\Longrightarrow \{a, b, c\}$
✓ $\{d, f\}$	$\Longrightarrow \{a, b, c, d, f\}$

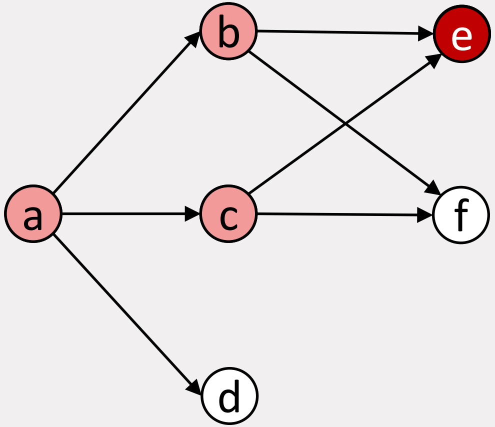
No antichain:

× $\{b, f\}$

× $\{a, e\}$

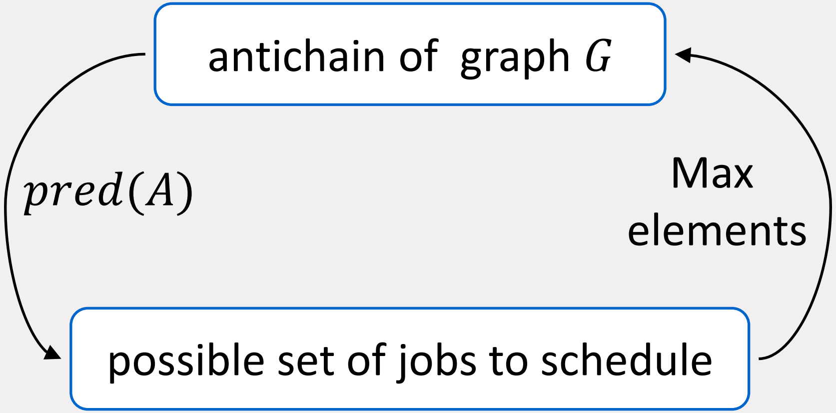
Definitions

Def: An *antichain* is a set A whose elements are pairwise incomparable.



Precendence Constraints Graph G

There is a one-to-one relation between:



Dynamic Program

$$S(A,t) = \begin{cases} true, & \text{if } pred(A) \text{ can be done before or at } t, \\ false, & \text{else} \end{cases}$$

If $S(A,D) = true$, for some A with $|\text{pred}(A)| \geq k$, then return YES.

For $t = 0$

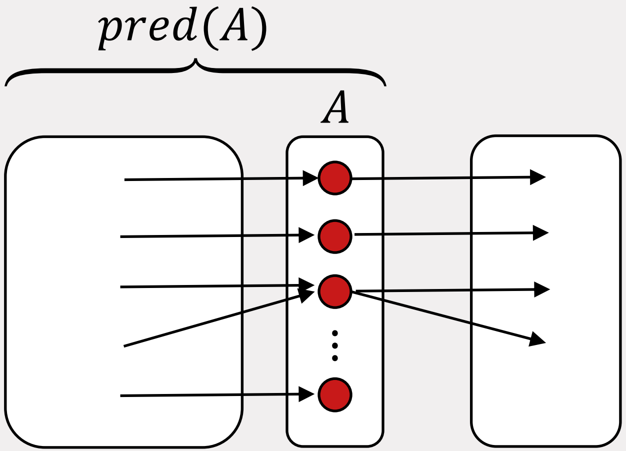
$$S(A,0) = \begin{cases} true, & \text{if } A = \emptyset, \\ false, & \text{else} \end{cases}$$

Dynamic Program

$$S(A,t) = \begin{cases} true, & \text{if } pred(A) \text{ can be done before or at } t, \\ false, & \text{else} \end{cases}$$

Only $a \in A$ can be processed at time t.

So, $\binom{|A|}{m} \leq 2^{|A|}$ possibilities.



For each possibility:

Process subset at timeslot $[t-1, t]$.

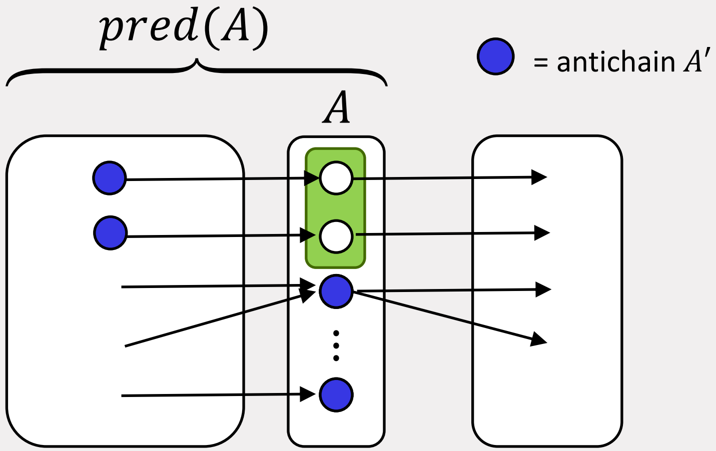
Check if leftover jobs can be finished at $t-1$.
i.e. Check whether $S(A', t-1) = true$

Dynamic Program

$$S(A,t) = \begin{cases} true, & \text{if } pred(A) \text{ can be done before or at } t, \\ false, & \text{else} \end{cases}$$

Only $a \in A$ can be processed at time t.

So, $\binom{|A|}{m} \leq 2^{|A|}$ possibilities.



For each possibility:

Process subset at timeslot $[t-1, t]$.

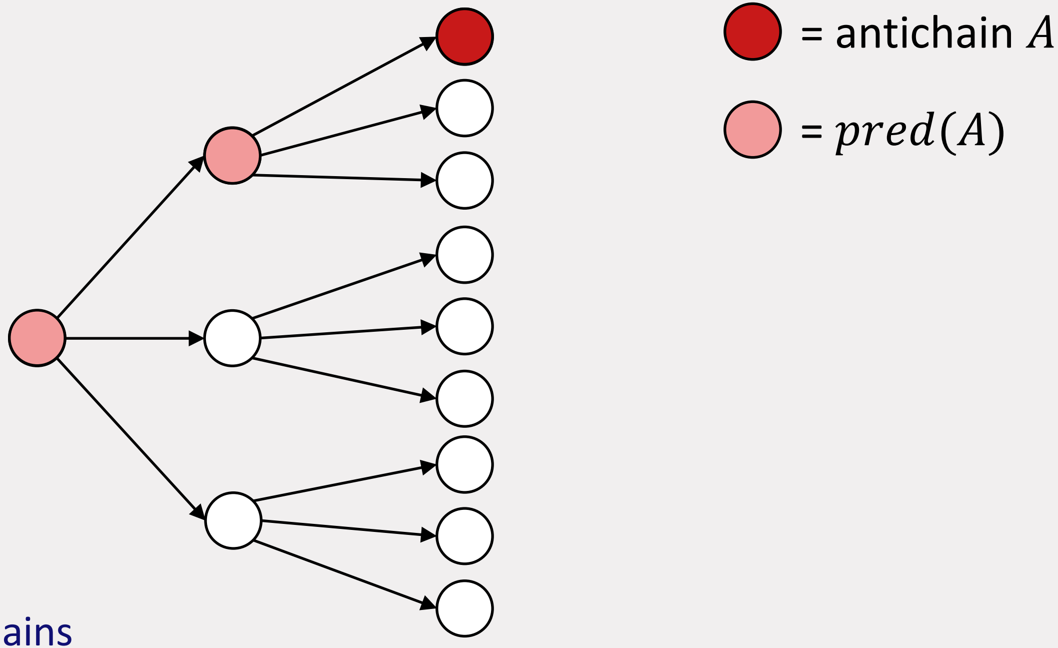
Check if leftover jobs can be finished at $t-1$.
i.e. Check whether $S(A', t-1) = true$

Runtime Analysis

- Amount of $S(A, t)$ to compute:
  - $t \leq k$,
  - Number of antichains A s.t. $|pred(A)| \leq k \leq \binom{n}{k} \leq n^k$,
- Computing $S(A, t)$: Trying all possible subsets of A at $[t - 1, t]$
  - Assuming $|A| \leq k$, at most 2^k possibilities

Number of Antichains (Example)

Let $k = 3$



So at least k^{k-1} possible antichains

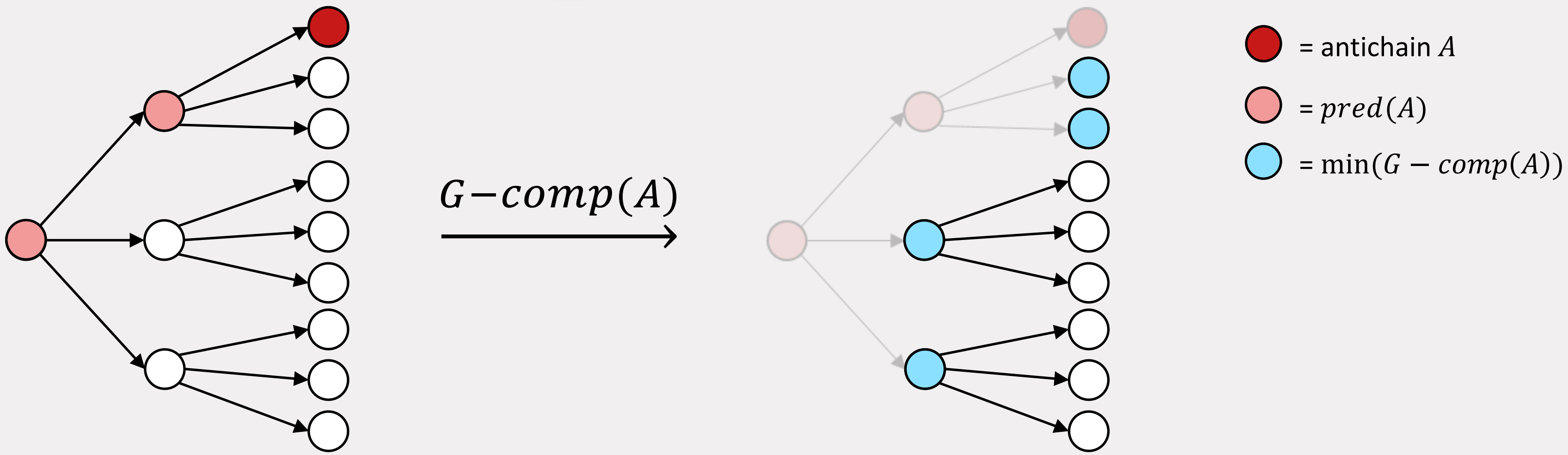
Definition of depth

Definition: The *depth* of an antichain A is defined as:

$$d(A) = |pred(A)| + \left|\min\left(G - comp(A)\right)\right|$$

3

4



TU/e

Definition of depth

Definition: The *depth* of an antichain A is defined as:

$$d(A) = |pred(A)| + \left|\min\left(G - comp(A)\right)\right|$$

<u>*Idea:*</u>

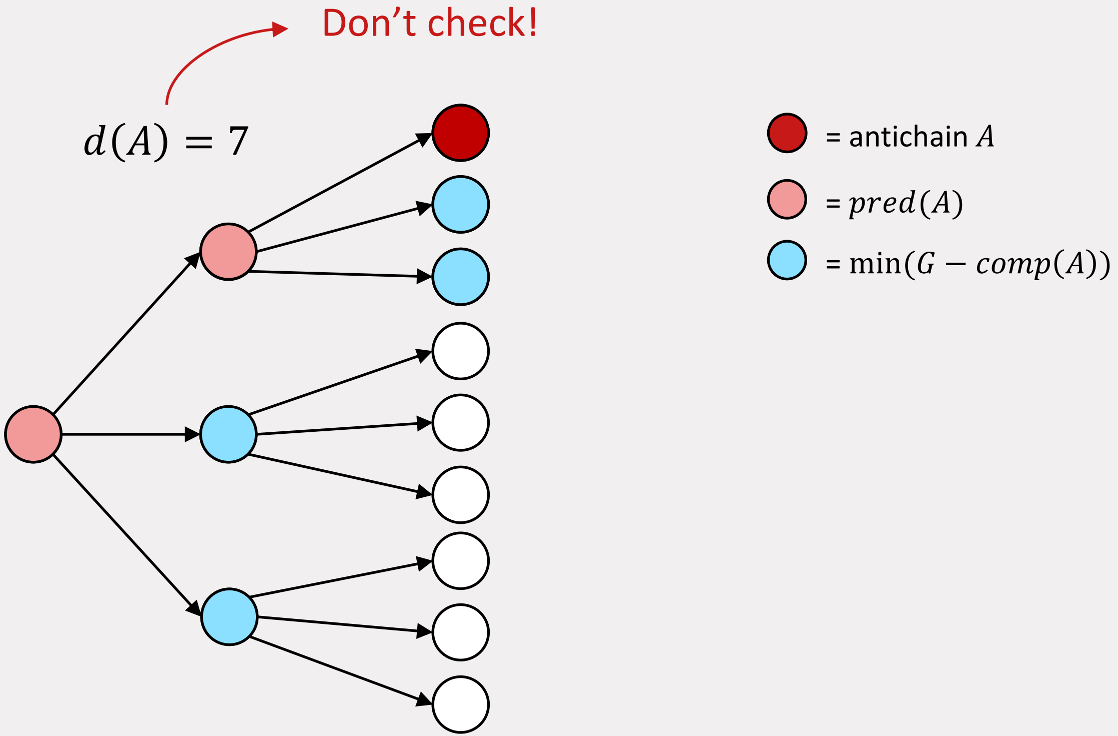
Only check antichains with $d(A) \leq k$.

Theorem: There are at most 4^{k} *antichains with depth* $d(A) \leq k$.

Is that enough?

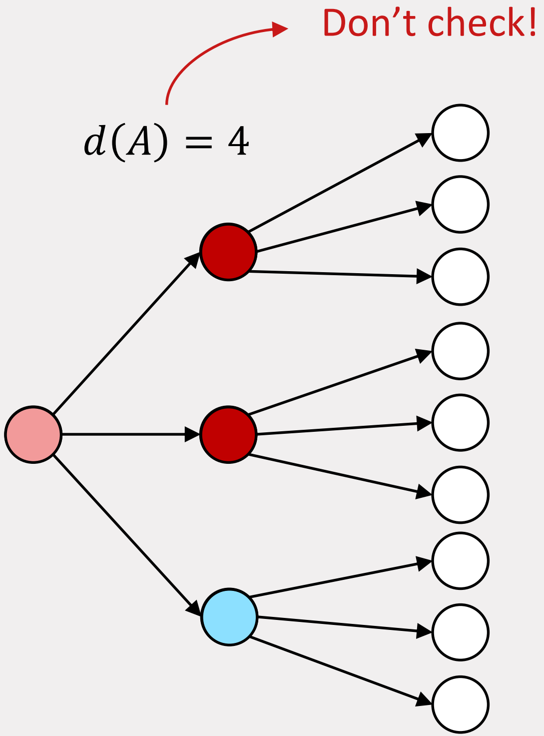
Restricting Depth

Let $k = 3$



Restricting Depth

Let $k = 3$



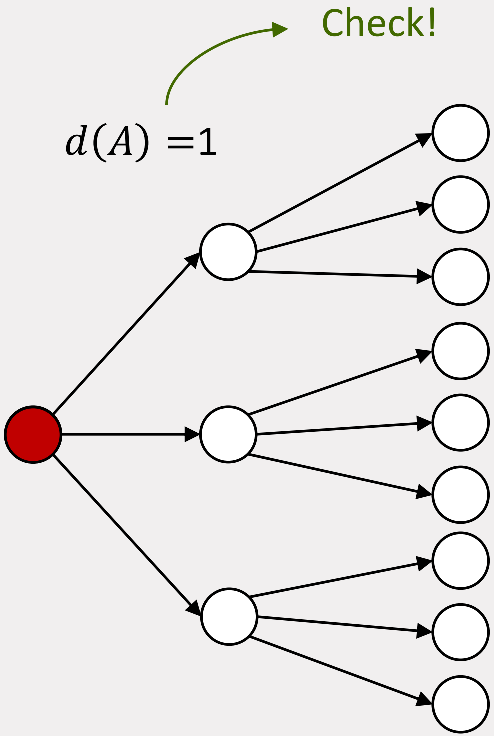
= antichain A

= $pred(A)$

= $\min(G - comp(A))$

Restricting Depth

Let $k = 3$



= antichain A

= $pred(A)$

= $\min(G - comp(A))$

Extending the Dynamic Program

Each time $S(A, t) = true$, ask:

Can we extend the given schedule to k jobs before deadline D?

Greedily schedule available jobs between t and D.

Restricting Depth

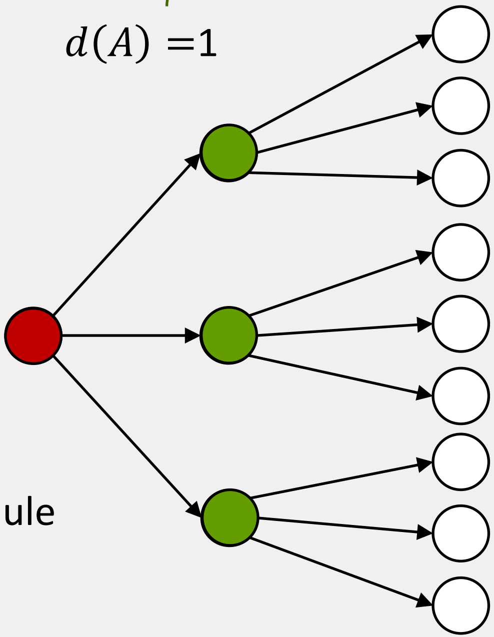
Let $k = 3$

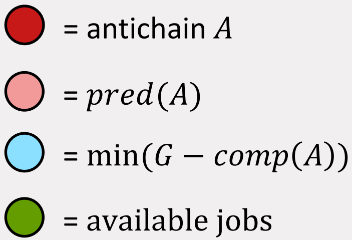
and

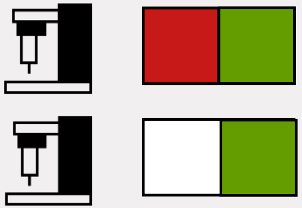
$m = 2, D = 2$

Check!

$d(A)$ =1







$S(A, 1) = true$

- Find available jobs
- Greedily extend schedule
- Return YES

TU/e

Runtime Analysis

- Amount of $S(A,t)$ to compute:
  - $t \leq k$,
  - ~~#{antichains A s.t. $|pred(A)| \leq k\} \leq \binom{n}{k} \leq n^k$~~,
    #{antichains A s.t. $d(A) \leq k\} \leq 4^k$
- Computing $S(A,t)$: Trying all possible subsets of A at $[t-1,t]$
  - Assuming $|A| \leq k$, at most 2^k possibilities
  - Expanding schedule in polynomial time

So runtime of $O(8^k \cdot poly(n))$.

Summary / Open problems

- Partial Scheduling with parameter k
- Trichotomy of the parameterized complexity
- $P|prec, p_j = 1|k - sched, C_{max}$
  - Dynamic Program
  - *Depth* of an antichain
  - Runtime of $O(8^k \cdot poly(n))$

Open problem

- Can we improve this bound?

Theorem: There are at most 4^{k} *antichains with depth* $d(A) \leq k$.