Makespan Scheduling of Unit Jobs with Precedence Constraints in $O(1.995^n)$ time

Jesper Nederlof

UU

Céline Swennenhuis

TU/e

Karol Węgrzycki

Saarbrücken

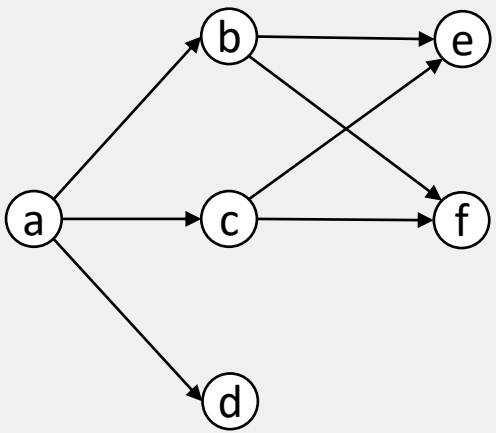
Department of Mathematics and Computer Science, Eindhoven University of Technology

$P|prec, p_j = 1|C_{\max}$

Given:

- n jobs of length 1
- m (identical, parallel) machines
- A precedence graph G
- $T \in \mathbb{N}$

Q: Is there a schedule of makespan T?



time →

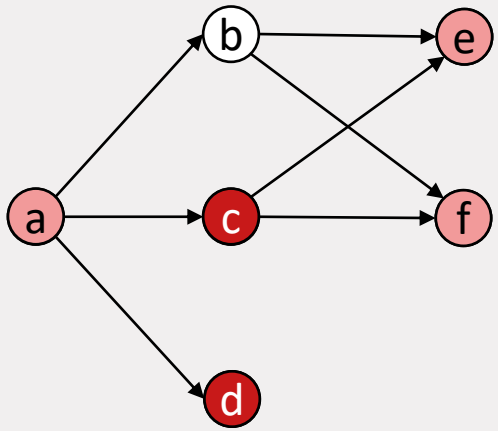
	1	2	3	4
1	a	b	d	f
2		c	e	

$P|prec, p_j = 1|C_{\max}$

- NP-complete
- For $m = 2$: polynomial time algorithms [FKN 1969, CG 1972, G 1982, GT 1985]
- For $m \geq 3$: complexity unknown! [Garey and Johnson 1961]

- XNLP-hard when parameterized by m $(+width(G))$ (i.e. W[t]-hard for all t) [BGNS 2021]

Definitions



Precendence Constraints Graph G

$G \Rightarrow$ partial order:

- $i \prec j$ if $(i,j) \in G$

Definition: Let A be a set of jobs.

$pred[A] = \{x \mid \exists\, a \in A \; s.t. \; x \preccurlyeq a\}$

$succ[A] = \{x \mid \exists\, a \in A \; s.t. \; x \succcurlyeq a\}$

$comp(A) = \{x \mid \exists\, a \in A \; s.t. \; x \preccurlyeq a \text{ or } x \succcurlyeq a\}$

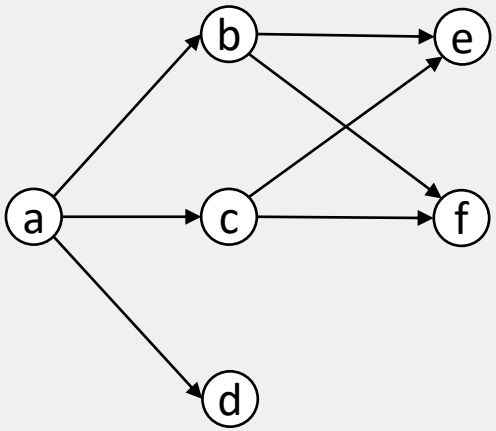
$sinks(A) = \max\{A\}$

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

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

Definitions

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



Precendence Constraints Graph G

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

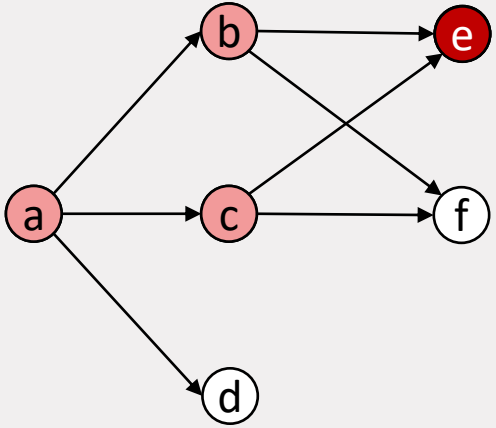
No antichain:

× $\{b, f\}$

× $\{a, e\}$

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:

A = antichain of graph G

$pred[A]$

$sinks(X)$

X = possible set of jobs to schedule

DP Algorithm:

$$T(A,t) = \begin{cases} \text{true}, & \text{if } pred[A] \text{ can be processed in } t \text{ time,} \\ \text{false}, & \text{otherwise.} \end{cases}$$

Antichain time

To compute $T(A,t)$:
try all $\binom{n}{m}$ possible jobs at time t

Runs in $O^*\left(\#\text{AC} \cdot \binom{\text{n}}{\text{m}}\right)$ time

$\#AC \leq 2^n$

$\Rightarrow O^*\left(2^n \cdot \binom{n}{m}\right)$ time algorithm

No $O^*(2^n)$ time algorithm known!

$P|prec, p_j = 1|C_{\max}$

Our result:

$P|prec, p_j = 1|C_{\max}$ can be solved in $O(1.995^n)$ time.

Proof: Combination of three algorithms

$P|prec, p_j = 1|C_{\max}$

- DP Algorithm
  - Dynamic Programming
  - $O^*\left(\#AC \cdot \binom{n}{m}\right)$ time

- Fast Subset Convolution Algorithm
  - Non-trivial application of Björklund, Husfeldt and Koivisto (SICOMP 2009)
  - $O^*(\#AC + 2^{n-m})$ time

Fast enough when

$|C| \geq n/7.5$

$\Rightarrow$ largest antichain

$< \left(1 - \frac{1}{7.5}\right) n$

$\xRightarrow{\text{Dilworth's Theorem}}$

$\#AC \leq 1.94^n$

- FPT Algorithm
  - Parameterized by vertex cover of *comparability graph*
  - $O^*\left(169^{|C|}\right)$ time

Fast enough when

$|C| < n/7.5$

What is vertex cover of comparability graph?

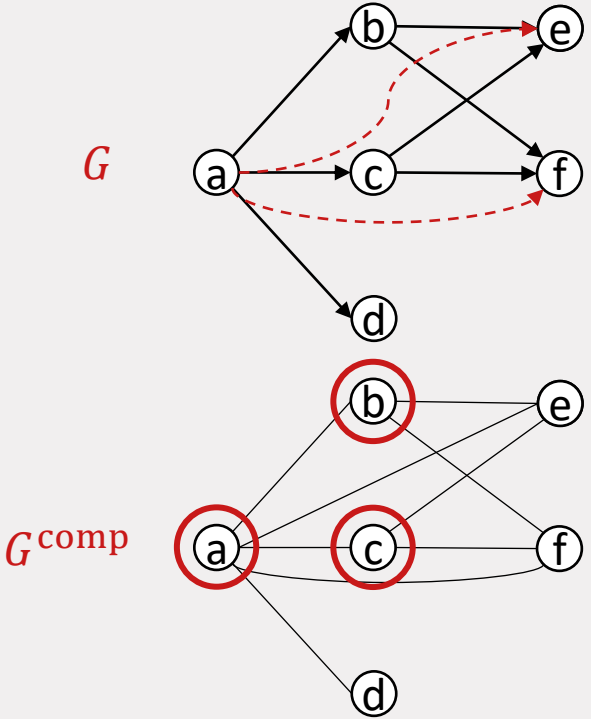
Let $G = (V, A)$ be input. Then $G^{\text{comp}} := (V, E)$

where $(v, w) \in E$ if v and w are *comparable*.

- G^{comp} is undirected transitive closure of G.

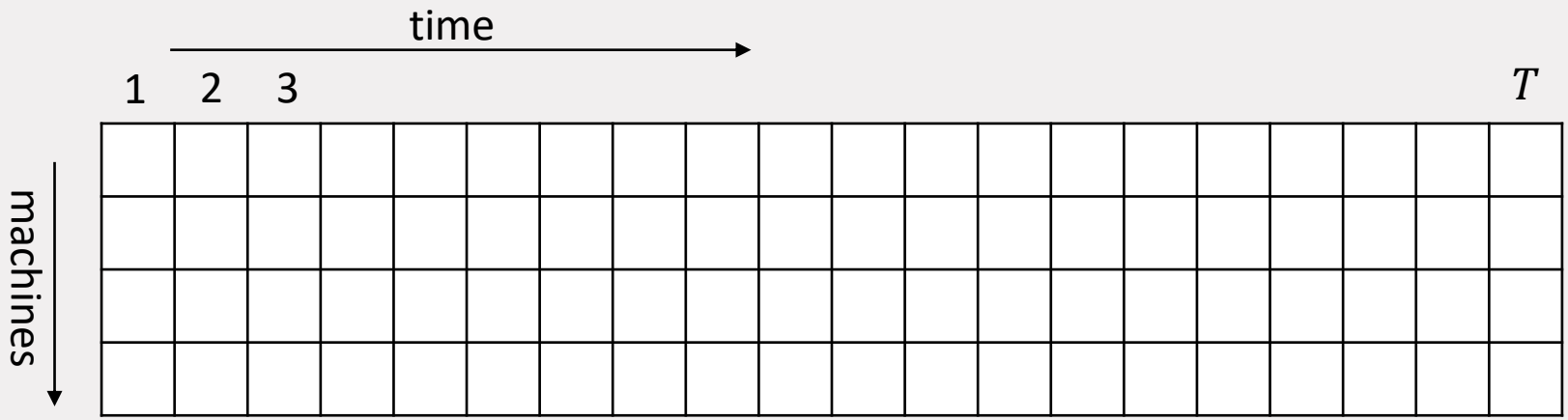
$C :=$ smallest vertex cover of G^{comp}

Claim: $V \setminus C$ is an antichain.



Zero-Adjusted Schedule (D&W)

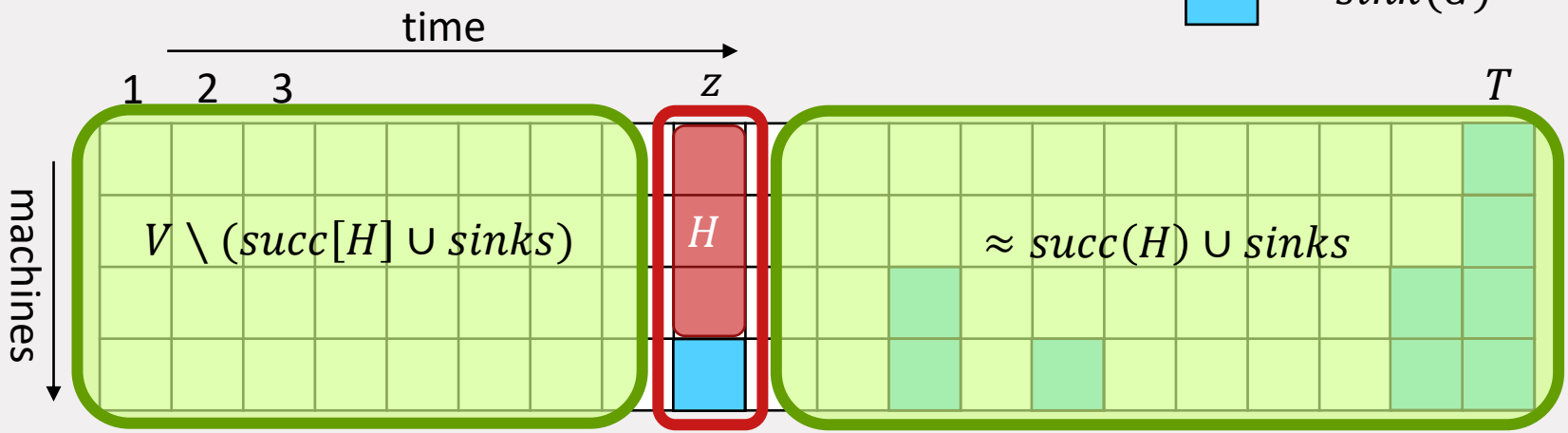
Assumption: $n = m \cdot T$



Zero-Adjusted Schedule (D&W)

Assumption: $n = m \cdot T$

$= sink(G)$



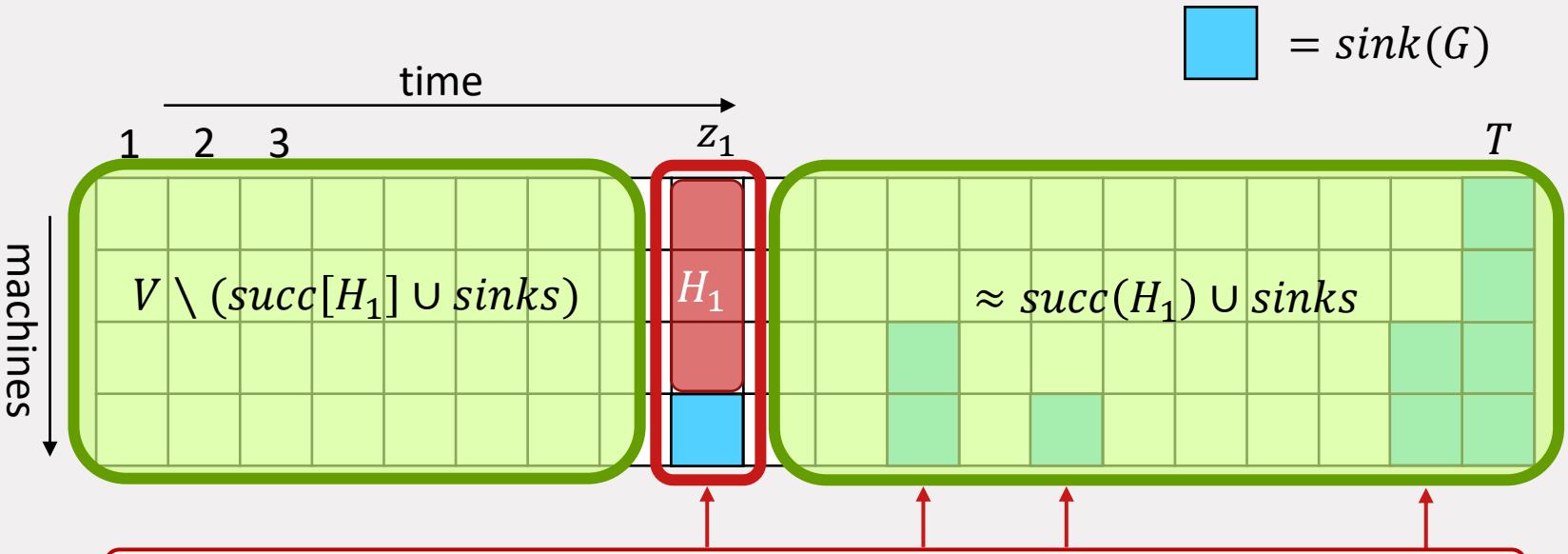
Let $z \in [1, T]$ be the first moment with a sink.

D&W: W.m.a. Each job x **after** z is a sink or a successor of a job at time z.

$\Rightarrow n^{O(m \cdot h)}$ time algorithm, $h = |\text{longest chain}|$

Sink-Adjusted Schedule

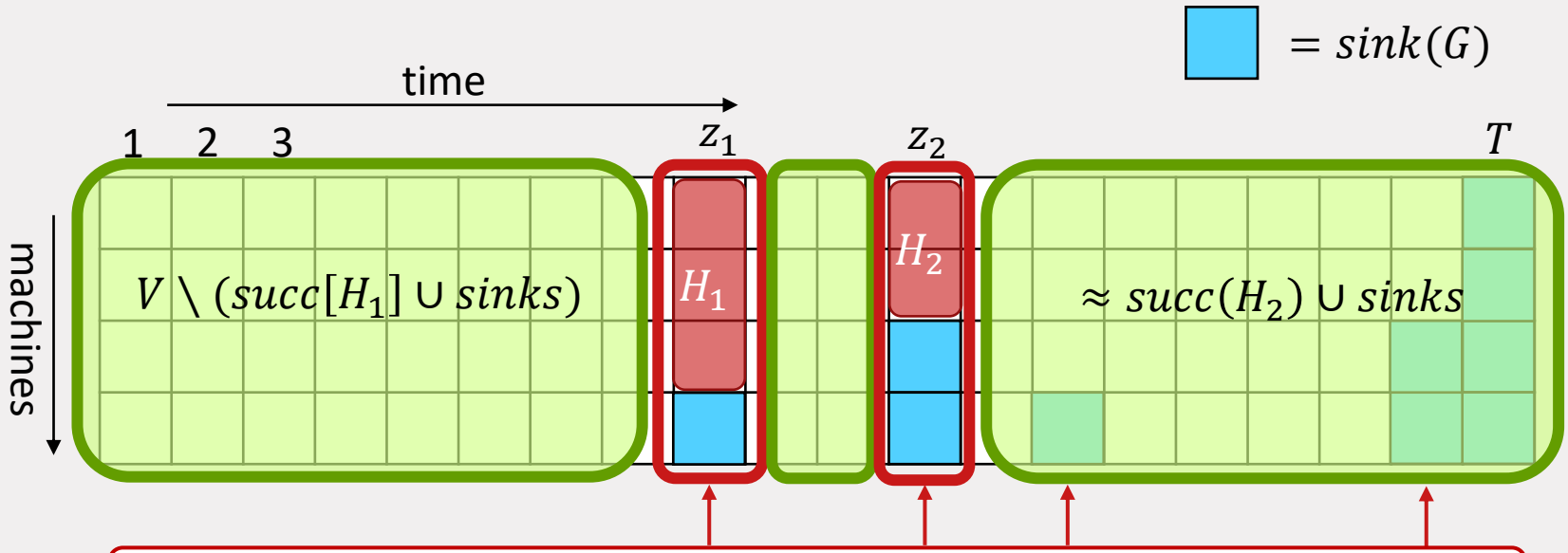
Assumption: $n = m \cdot T$



$z \in [1, T]$ is a ***sink moment*** if there are both sinks and non-sinks at time z.

Sink-Adjusted Schedule

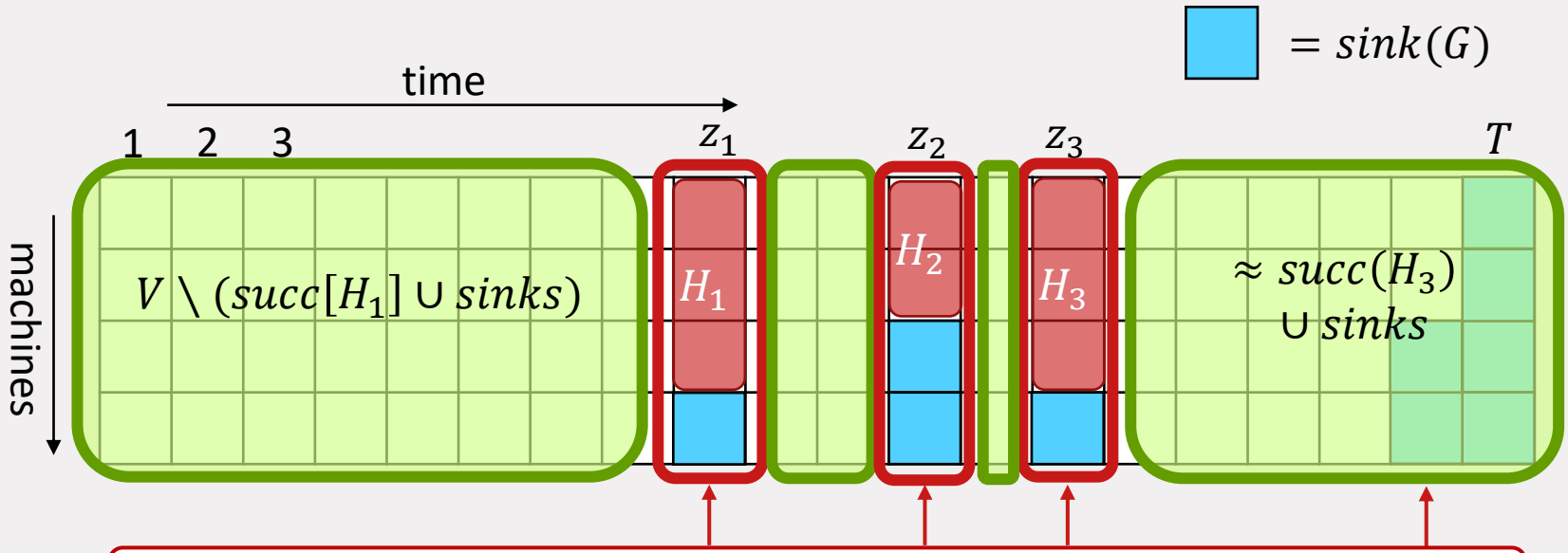
Assumption: $n = m \cdot T$



$z \in [1, T]$ is a ***sink moment*** if there are both sinks and non-sinks at time z.

Sink-Adjusted Schedule

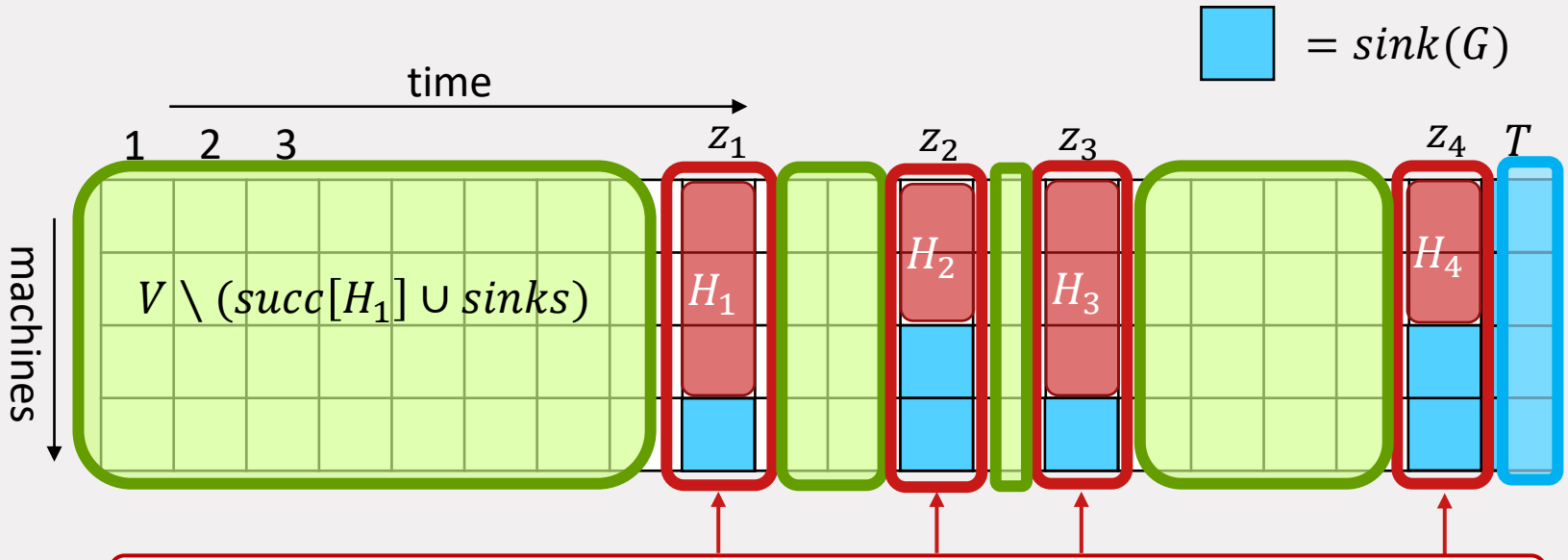
Assumption: $n = m \cdot T$



$z \in [1, T]$ is a ***sink moment*** if there are both sinks and non-sinks at time z.

Sink-Adjusted Schedule

Assumption: $n = m \cdot T$

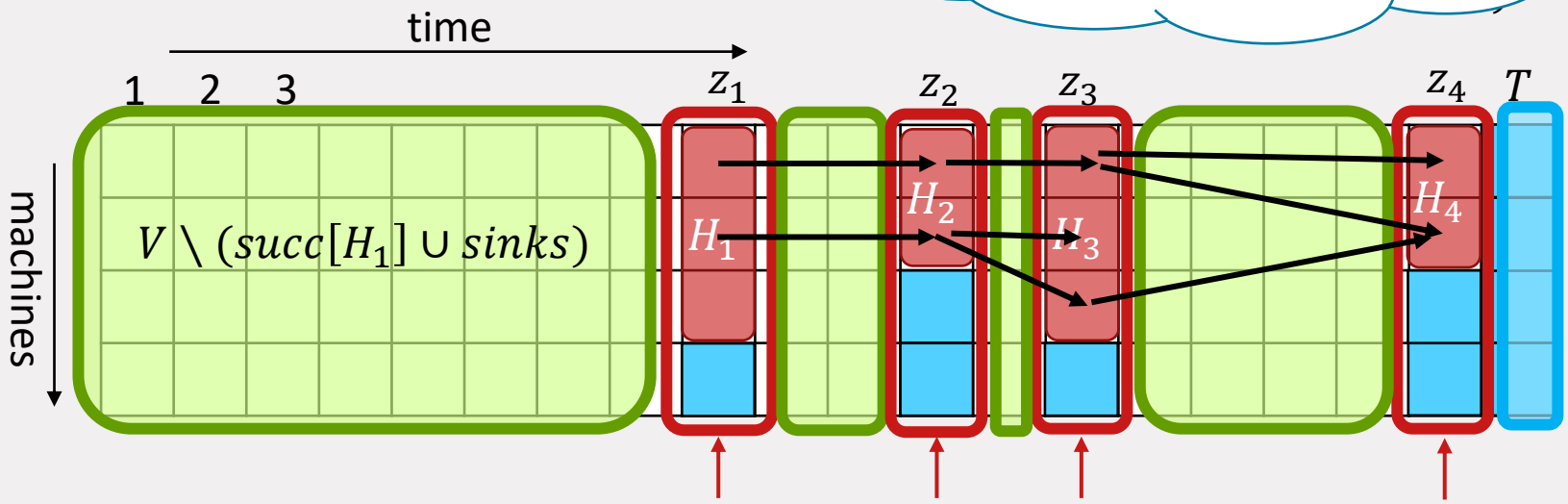


$z \in [1, T]$ is a ***sink moment*** if there are both sinks and non-sinks at time z.

No sinks in

Sink-Adjusted Schedule

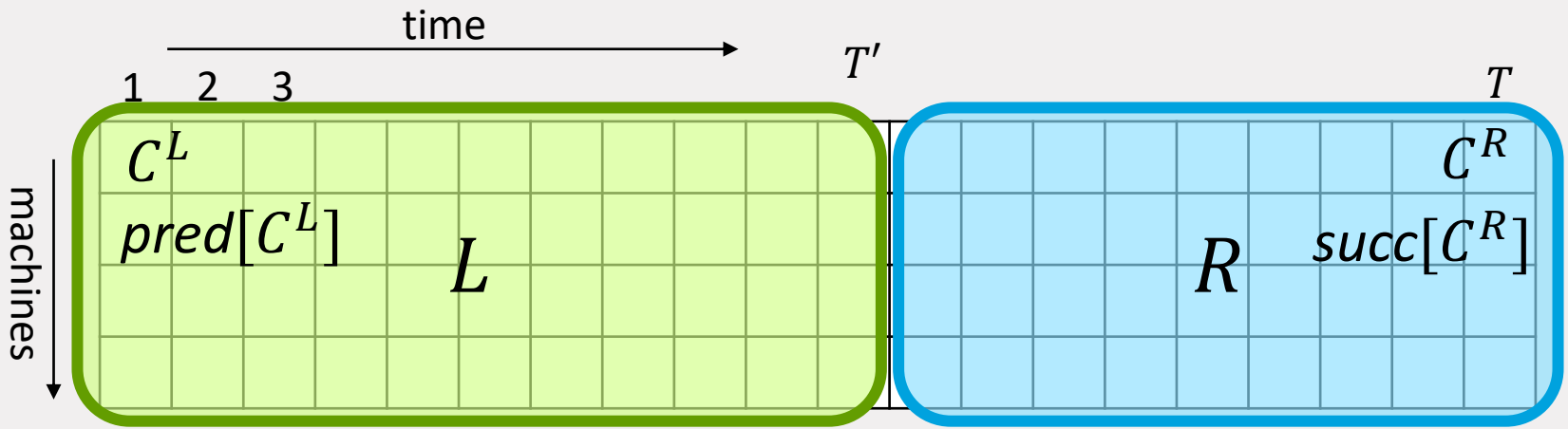
How to use this…?



Claim: if we know $\cup_i H_i$, we can derive $H_1, H_2, \ldots, H_l$

Middle-Adjusted Schedule

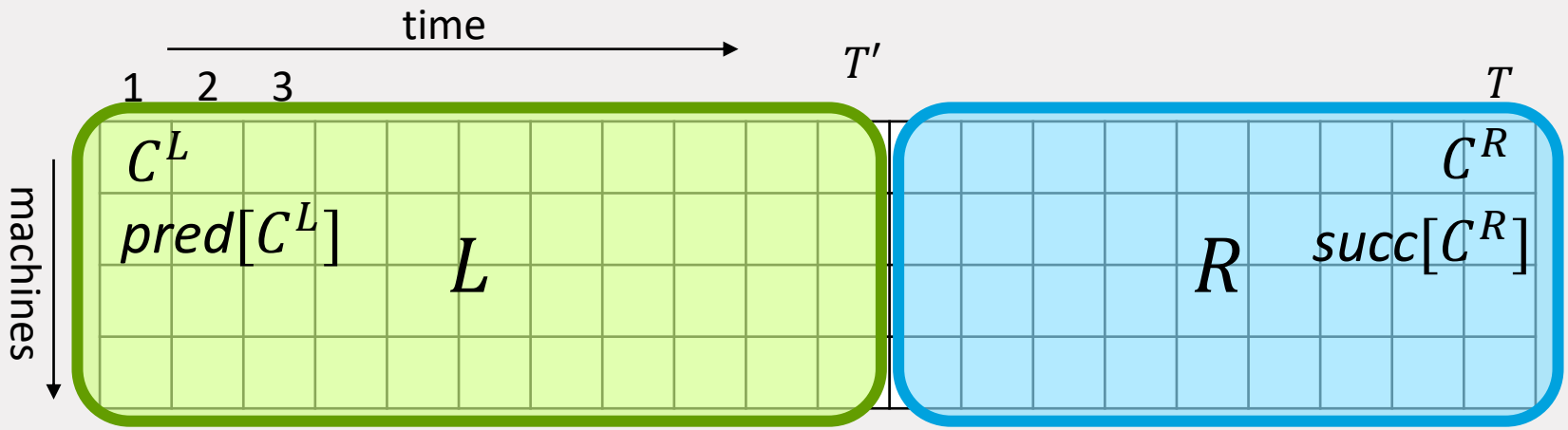
Assumption: $n = m \cdot T$



$$|C^L| \approx \frac{|C|}{2} \qquad |C^R| \approx \frac{|C|}{2}$$

Middle-Adjusted Schedule

Assumption: $n = m \cdot T$

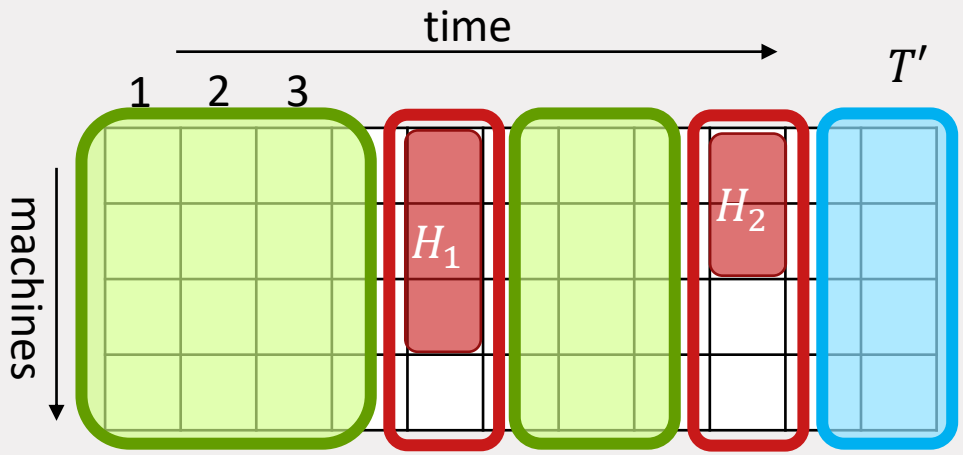


Undediced $U := V \setminus (pred[C^L] \cup succ[C^R])$

- U is antichain
- U are sinks in L
- U are sources in R

Middle-Adjusted Schedule

Assumption: $n = m \cdot T$



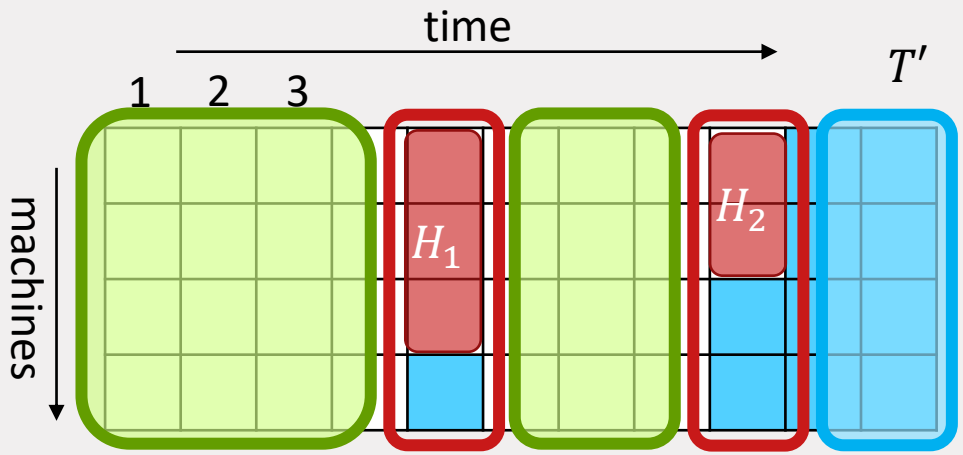
- Assume sink-adjusted

Undediced $U := V \setminus (pred[C^L] \cup succ[C^R])$

- U is antichain
- U are sinks in L

Middle-Adjusted Schedule

Assumption: $n = m \cdot T$



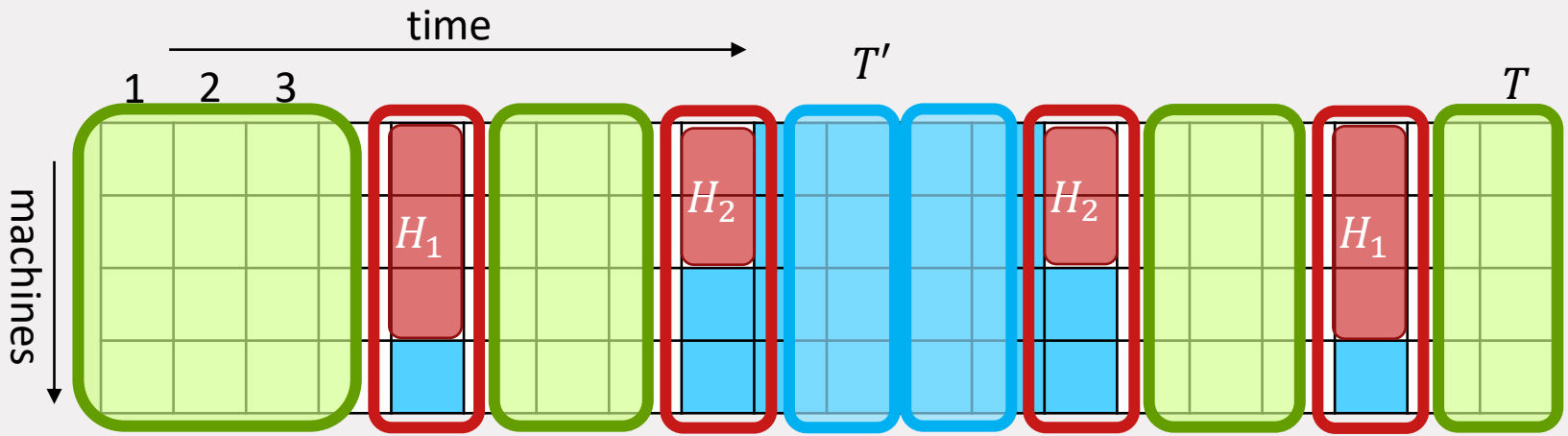
Undediced $U \coloneqq V \setminus (pred[C^L] \cup succ[C^R])$

- U is antichain
- U are sinks in L

- Assume sink-adjusted
- Guess jobs in H from C^L
- Reconstruct $H_1, H_2, \ldots$
- Find out which jobs from U can be processed at which slots

Middle-Adjusted Schedule

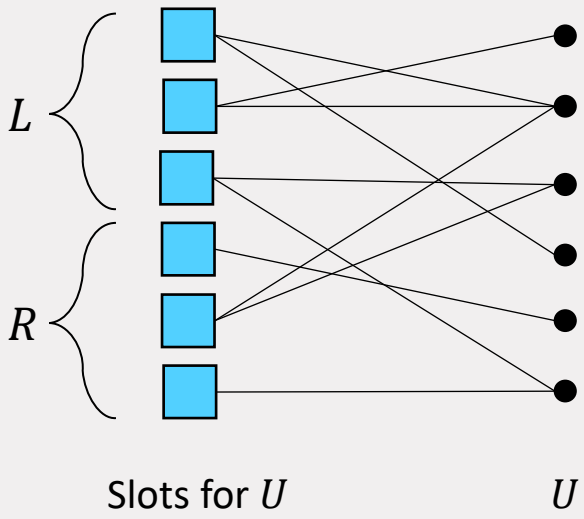
Assumption: $n = m \cdot T$



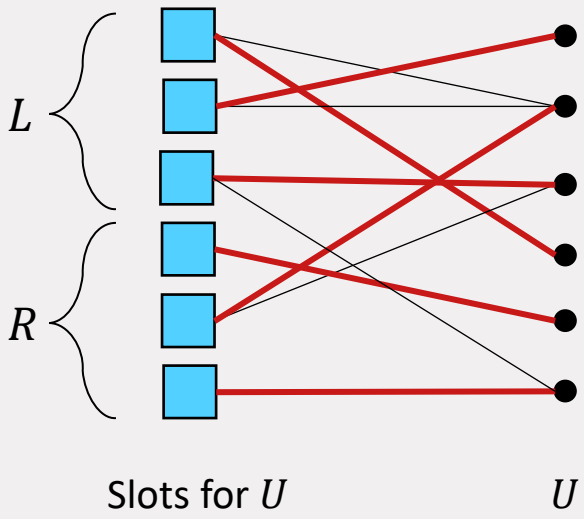
Undediced $U := V \setminus (pred[C^L] \cup succ[C^R])$

- U is antichain
- U are sinks in L
- U are sources in R

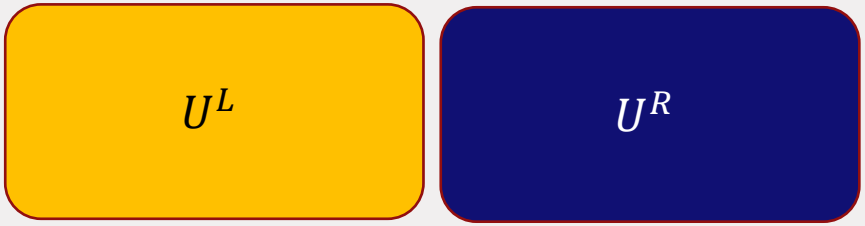
Sink-Adjusted Schedule (D&W)



Sink-Adjusted Schedule (D&W)

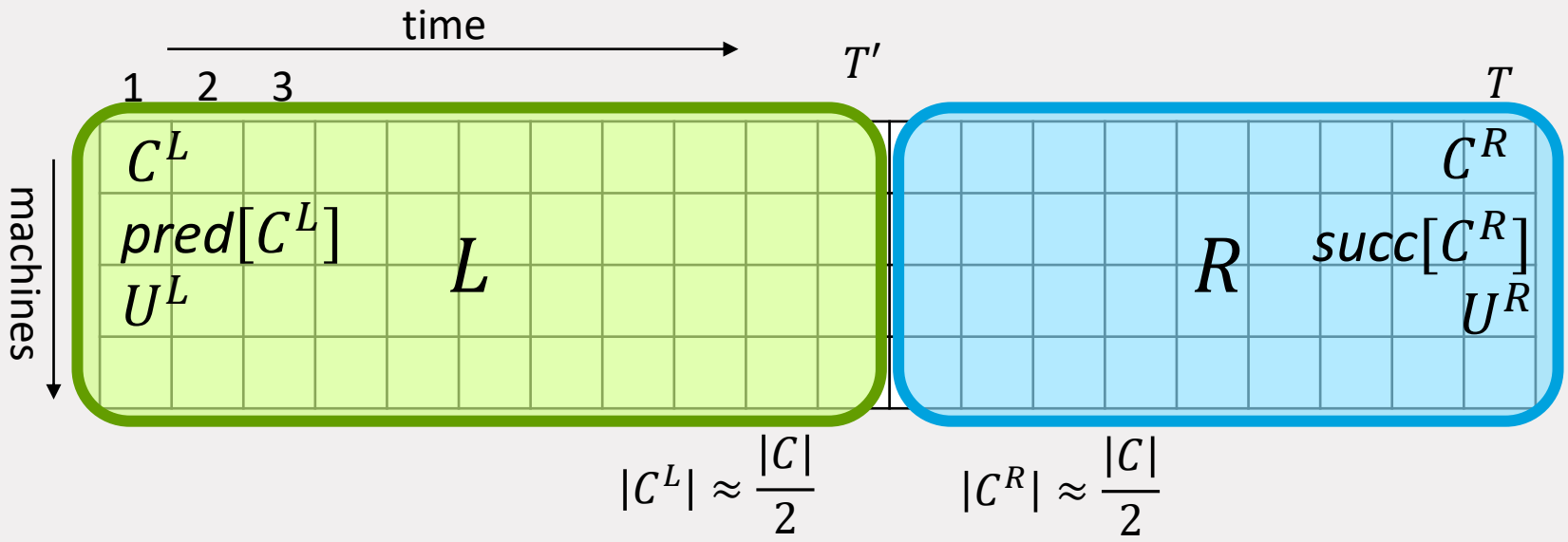


Find partition of U:



Middle-Adjusted Schedule

Assumption: $n = m \cdot T$



Guesses: $2^{|C|} \cdot 2^{|C|} \Rightarrow O^*(16^{|C|})$ time. **Not the full story.**

Conclusion

Main result:

$P|prec, p_j = 1|C_{\max}$ in $O(1.995^n)$ time.

Key idea's:

- Tradeoff between $\#AC's$ and $|C|$.
- $O^*(169^{|C|})$ time algorithm

Future Research:

$P3|prec, p_j = 1|C_{\max}$ in $2^{o(n)}$ time?

Thanks for your attention!

Extra ideas for the general case