Parallel Machine Scheduling

with a Single Resource per Job

T. Janssen[1] C. Swennenhuis[2] A. Bitar[3] T. Bosman[4]
S. Dauzère-Pérès[5] D. Gijswijt[1] L. van Iersel[1] C. Yugma[5]

[1]Delft Institute of Applied Mathematics, Delft University of Technology

[2]Department of Mathematics and Computer Science, Eindhoven University of Technology

[3]Decision Brain, Montpelleir

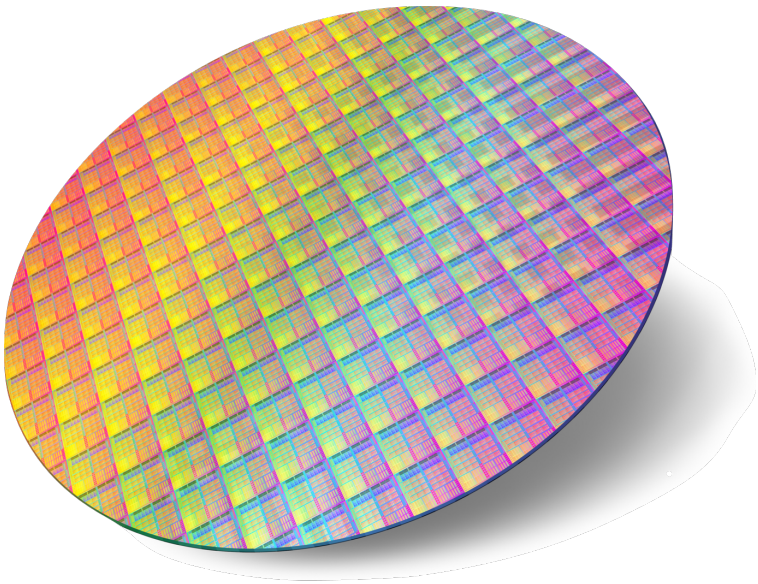
[4]Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam

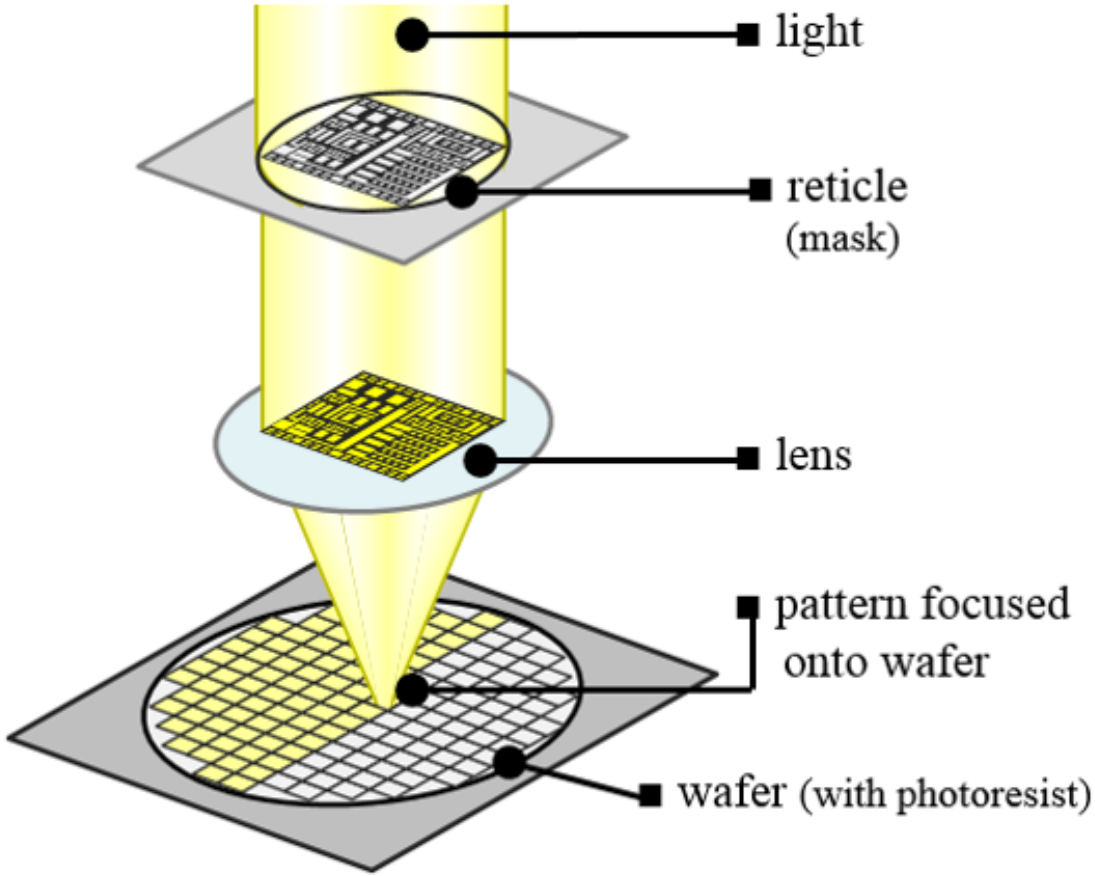
[5]Department of Manufacturing Sciences and Logistics, Center of Microelectronics in Provence

MAPSP, June 2019

Motivation: Semi-conductor Industry

Motivation: Semi-conductor Industry





Motivation: Semi-conductor Industry

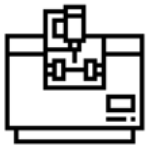
Table of Contents

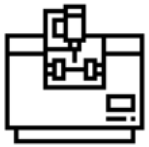
1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

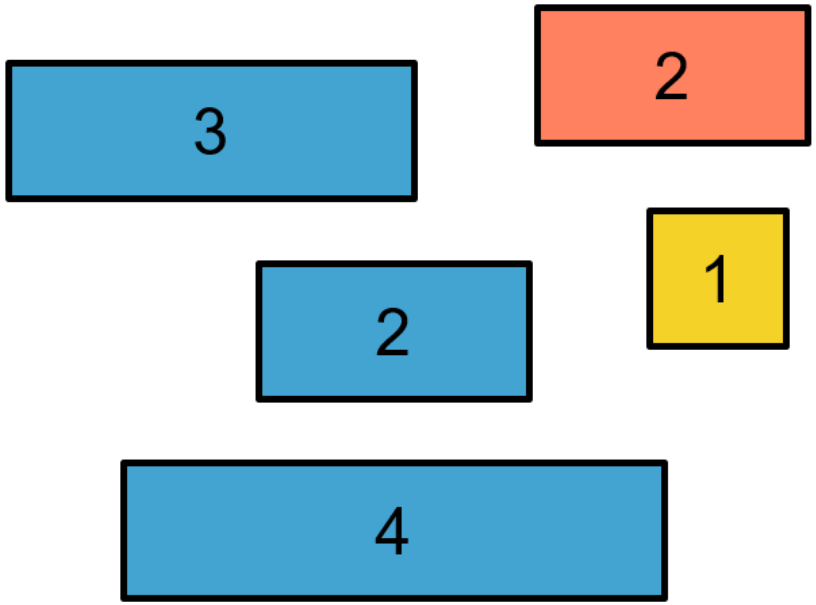
Table of Contents

1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

Problem Definition

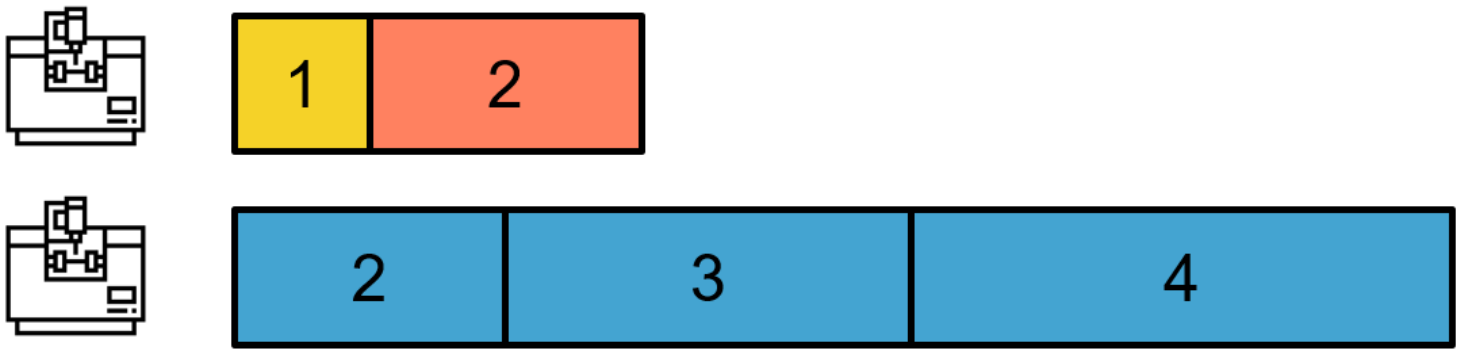




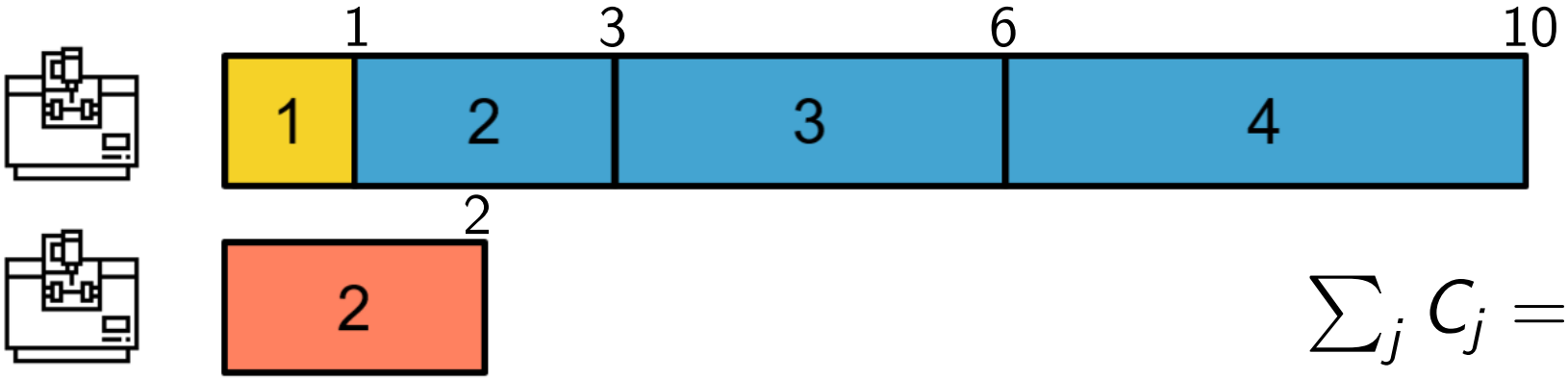


Problem Definition

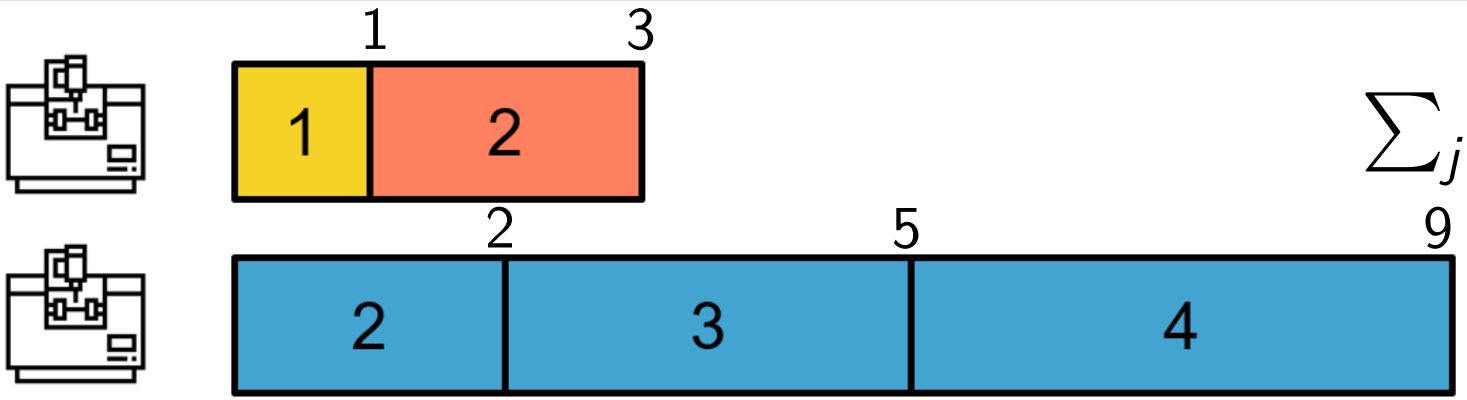
Problem Definition



Problem Definition



$$\sum_j C_j = 22$$



$$\sum_j C_j = 20$$

Problem Definition

- Input:
  - m machines
  - n jobs with processing times p_j
  - k resources
  - jobs are partitioned into which resource they use
- Schedule feasible
  - When each resource is used at most once at a time
- Objective:
  - Minimize $\sum_j C_j$

We denote[1] this problem by:

$$P|partition|\sum_j C_j$$

[1] Using the notation of [Graham et al., 1979]

Problem Definition

The complexity of $P|partition|\sum_j C_j$ remains open.

Table of Contents

1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

SPT-*available* Rule

Definition (Shortest Processing Time Rule, SPT)

The next job to be scheduled is the job with the shortest processing time.

SPT is exact for $P||\sum_j C_j$ [Smith, 1956]. However, how to deal with the resources?

SPT-*available* Rule

Definition (Shortest Processing Time Rule, SPT)

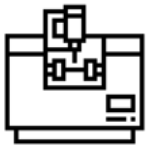
The next job to be scheduled is the job with the shortest processing time.

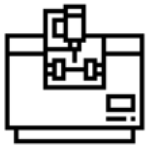
SPT is exact for $P||\sum_j C_j$ [Smith, 1956]. However, how to deal with the resources?

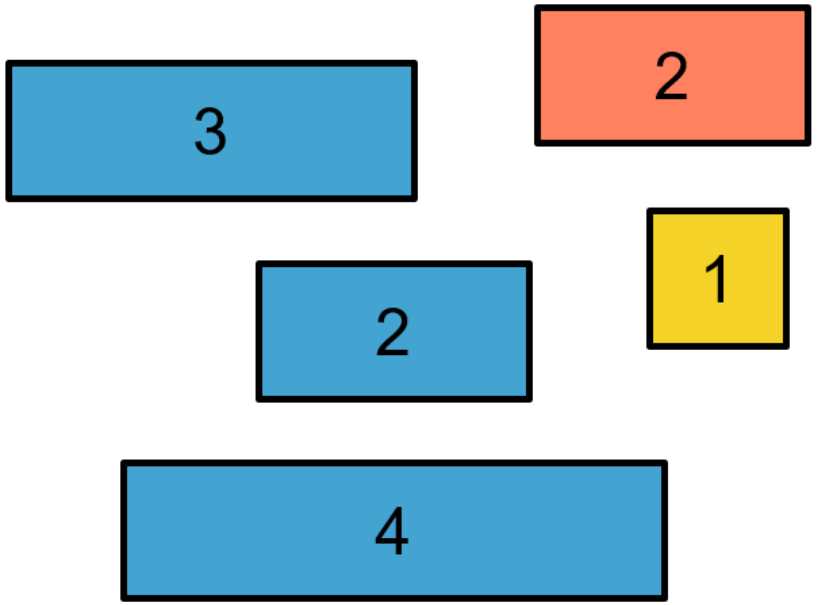
Definition (SPT-*available* Rule)

The next job to be scheduled is the job with the shortest processing time. If its resources is not available, schedule it at the earliest opportunity.

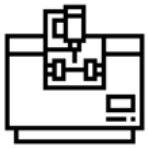
SPT-*available* Rule

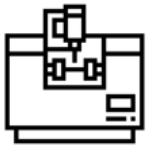




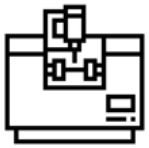


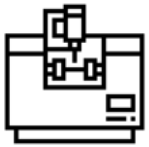
SPT-*available* Rule



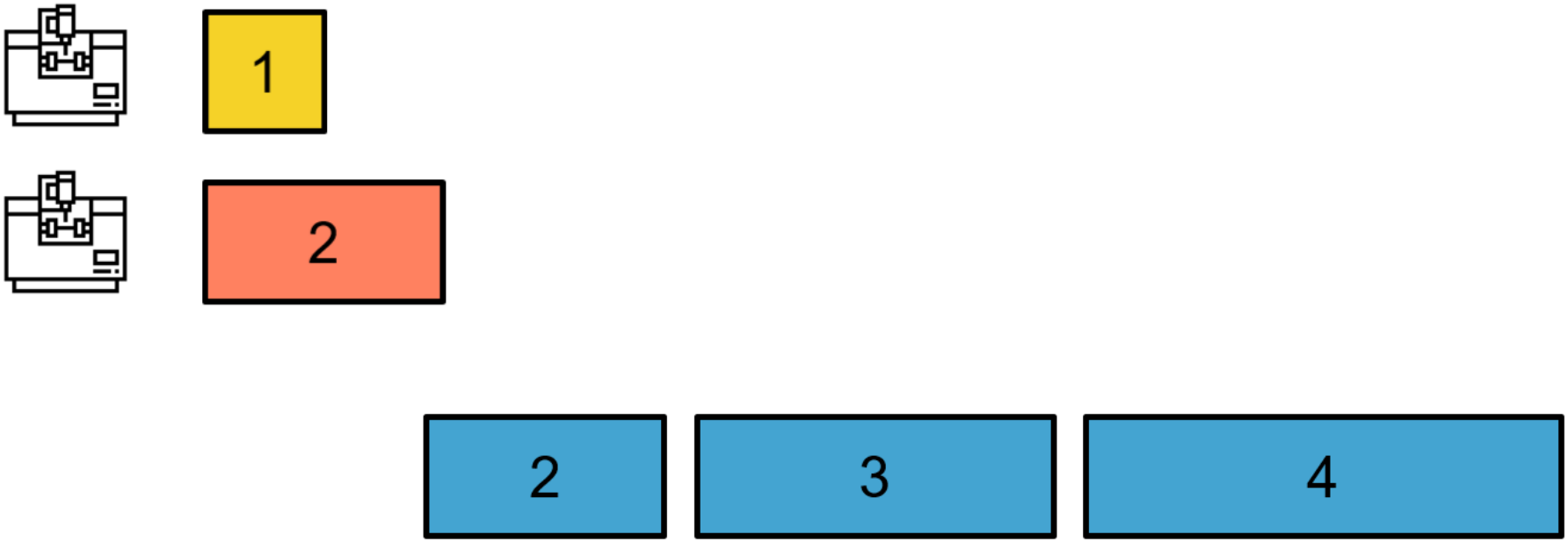


SPT-*available* Rule

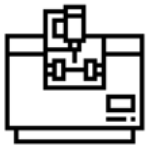


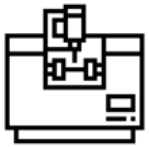


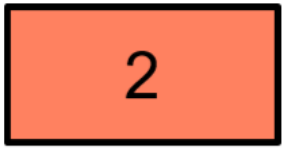
SPT-*available* Rule



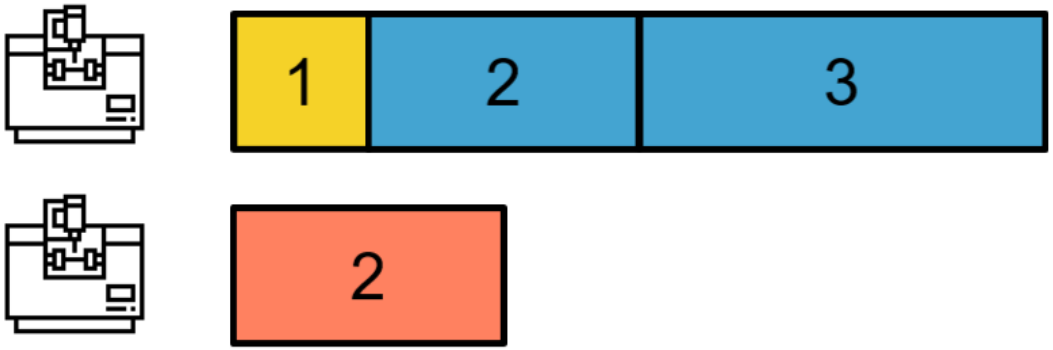
SPT-*available* Rule





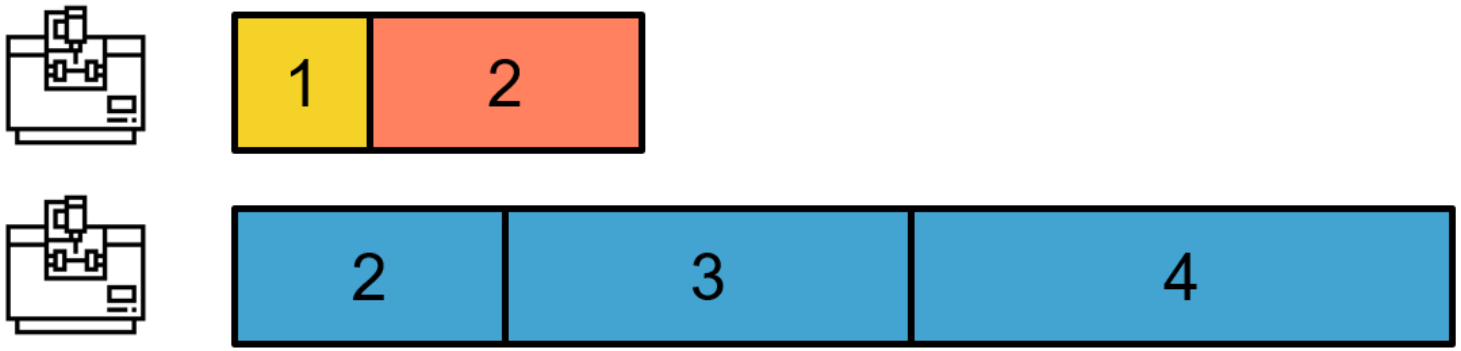


SPT-*available* Rule

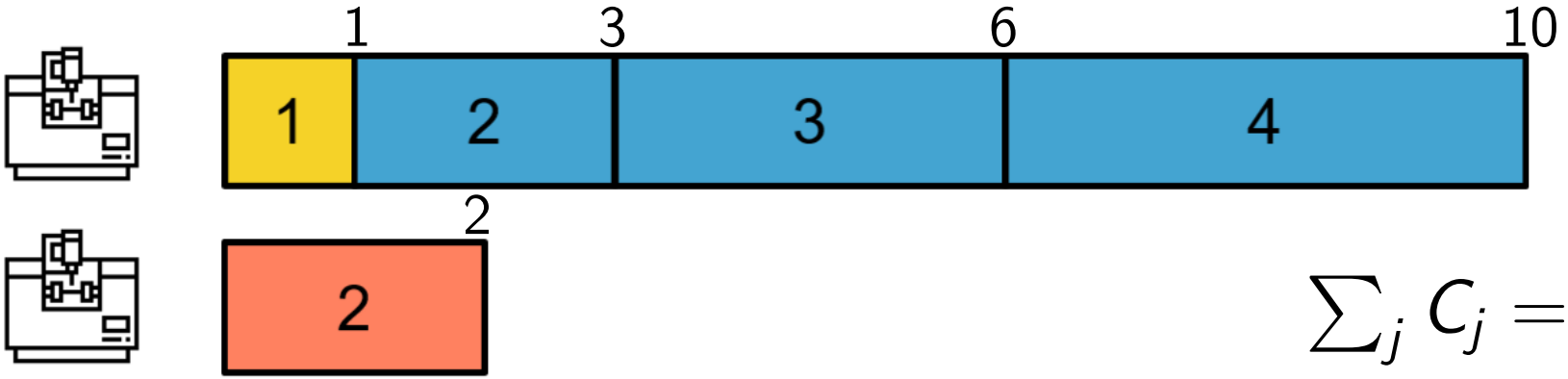


SPT-*available* Rule

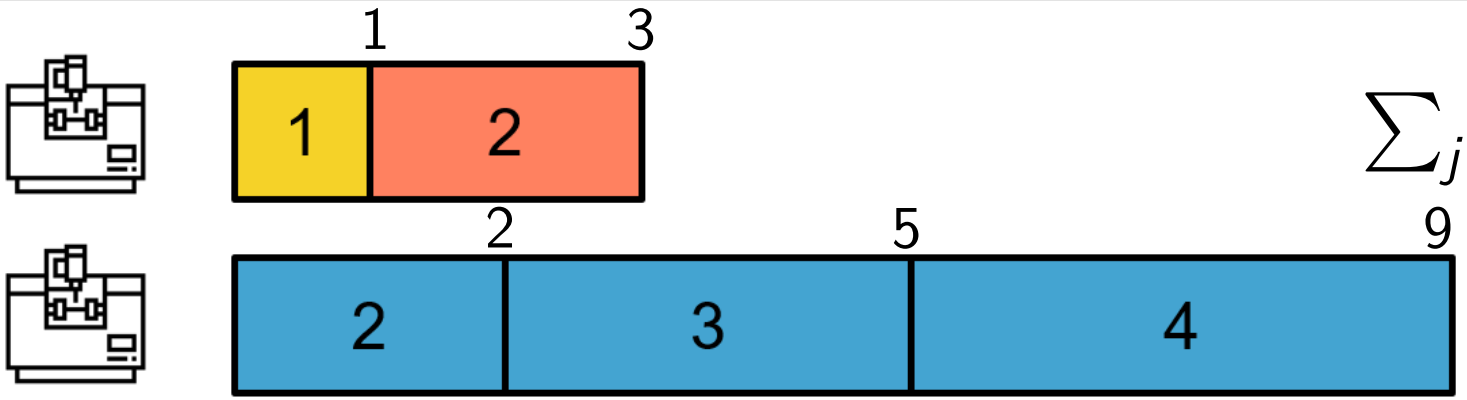
SPT-*available* Rule



SPT-*available* Rule



$$\sum_j C_j = 22$$



$$\sum_j C_j = 20$$

SPT-*available* Rule

Theorem 2

The SPT-*available* rule gives a $(2 - \frac{1}{m})$-approximation for $P|partition|\sum_j C_j$.

SPT-*available* Rule

Theorem 2

The SPT-*available* rule gives a $(2 - \frac{1}{m})$-approximation for $P|partition|\sum_j C_j$.

This bound is not tight. We can construct instances where it gives an α-approximation for $\alpha < \frac{4}{3}$.

Table of Contents

1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:

$P|partition|\sum_j C_j$

Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:

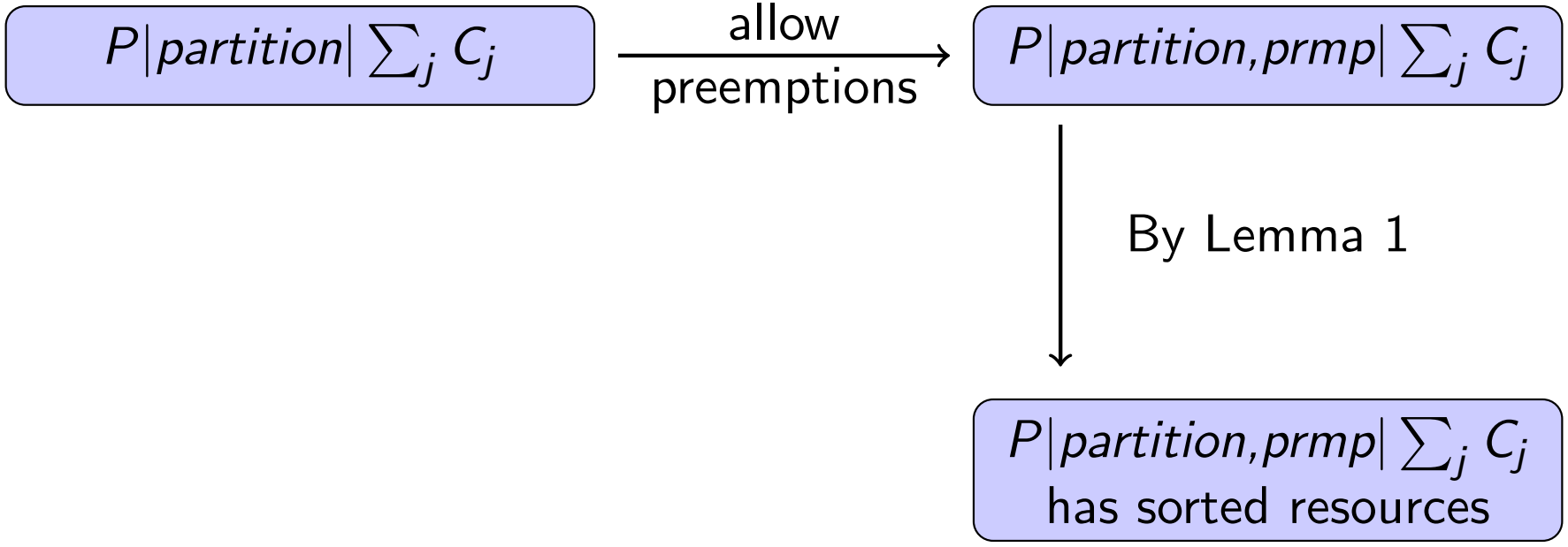
$$P|partition|\sum_j C_j \xrightarrow[\text{preemptions}]{\text{allow}} P|partition,prmp|\sum_j C_j$$

Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:

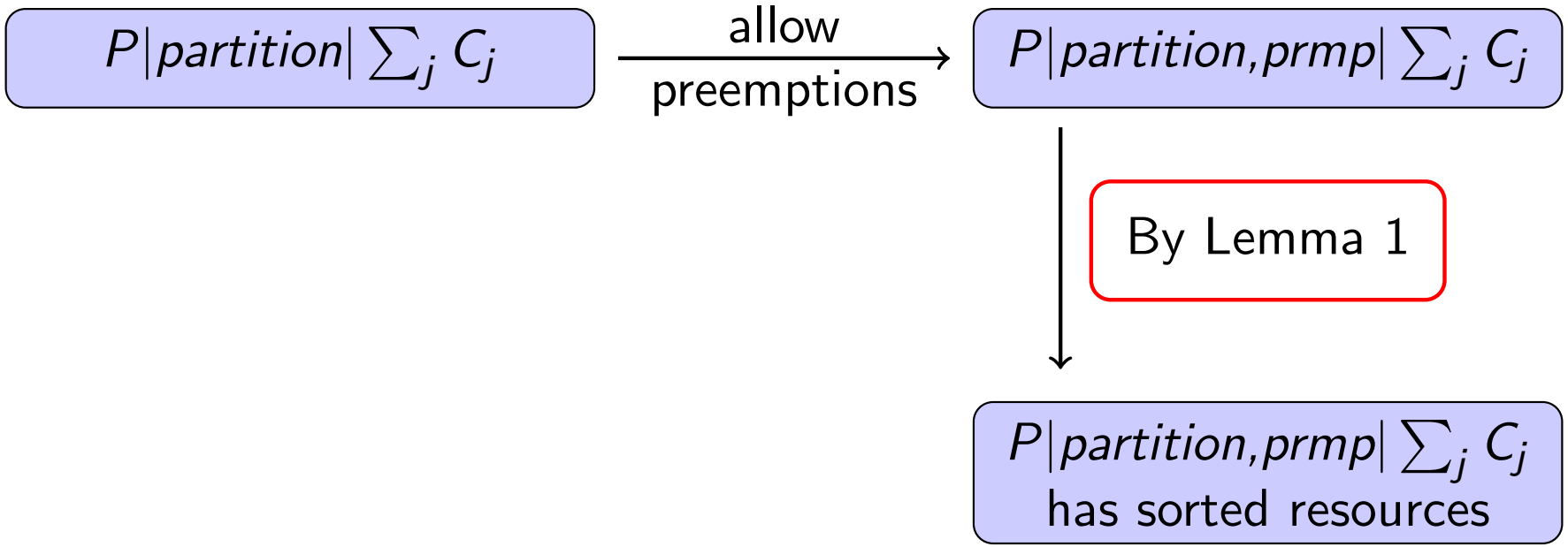


Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:



Ordered Resources Property

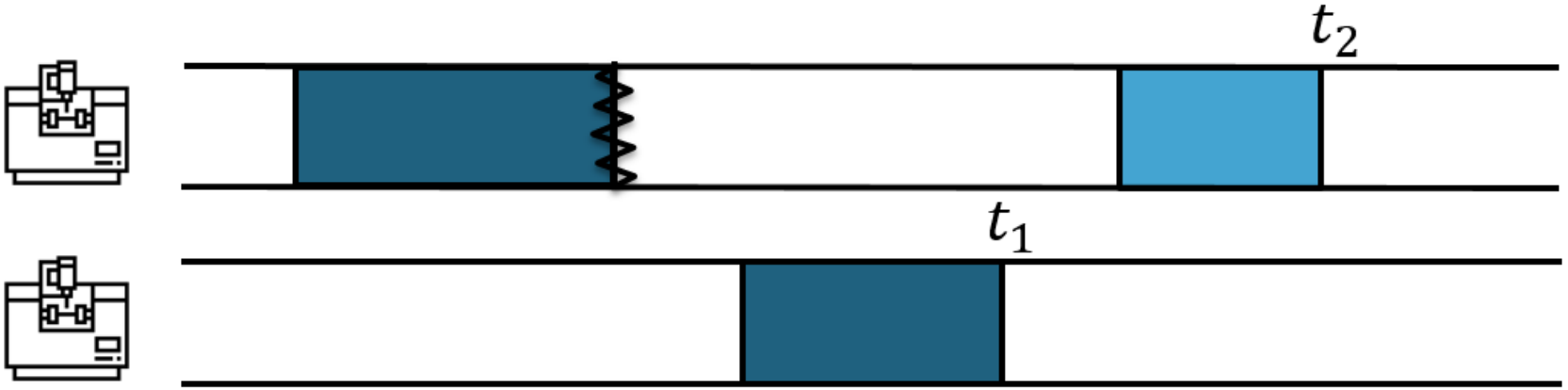
Lemma 1

In any optimal schedule for $P|partition, \text{prmp}|\sum_j C_j$, all jobs sharing the same resource must be processed in order of non-decreasing processing times.

Ordered Resources Property

Lemma 1

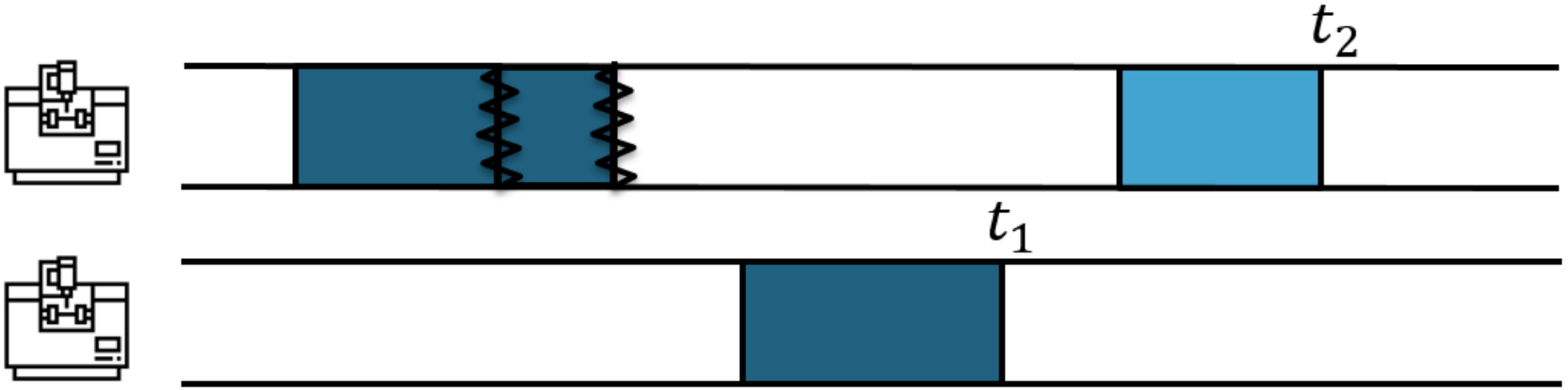
In any optimal schedule for $P|partition, \text{prmp}|\sum_j C_j$, all jobs sharing the same resource must be processed in order of non-decreasing processing times.



Ordered Resources Property

Lemma 1

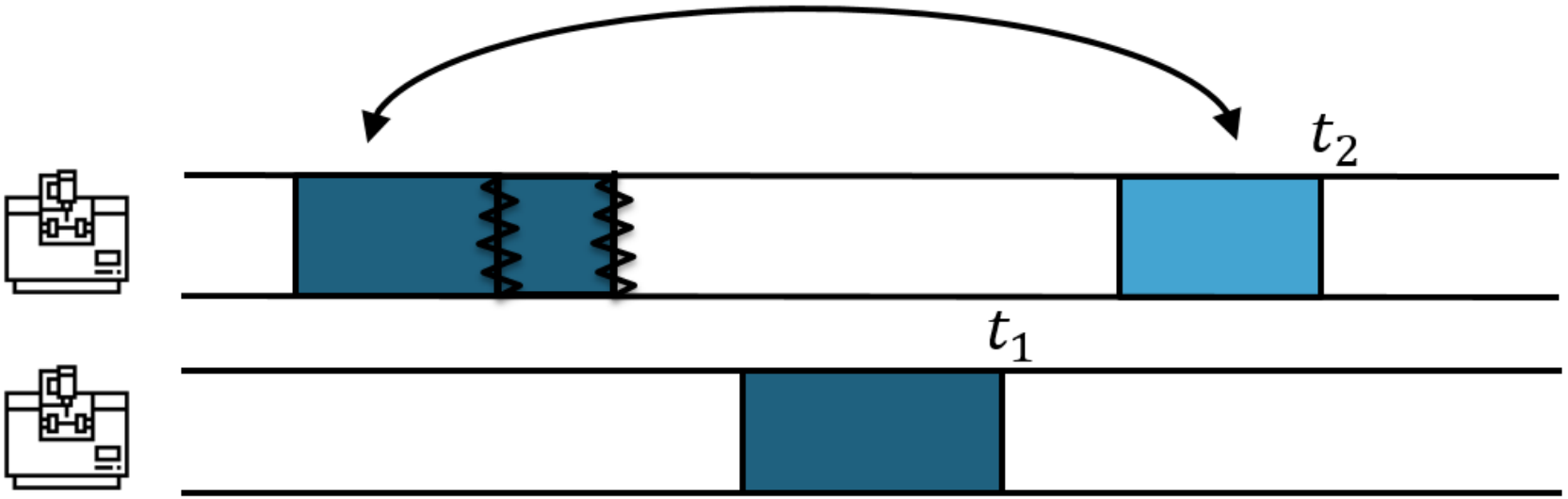
In any optimal schedule for $P|partition, \text{prmp}|\sum_j C_j$, all jobs sharing the same resource must be processed in order of non-decreasing processing times.



Ordered Resources Property

Lemma 1

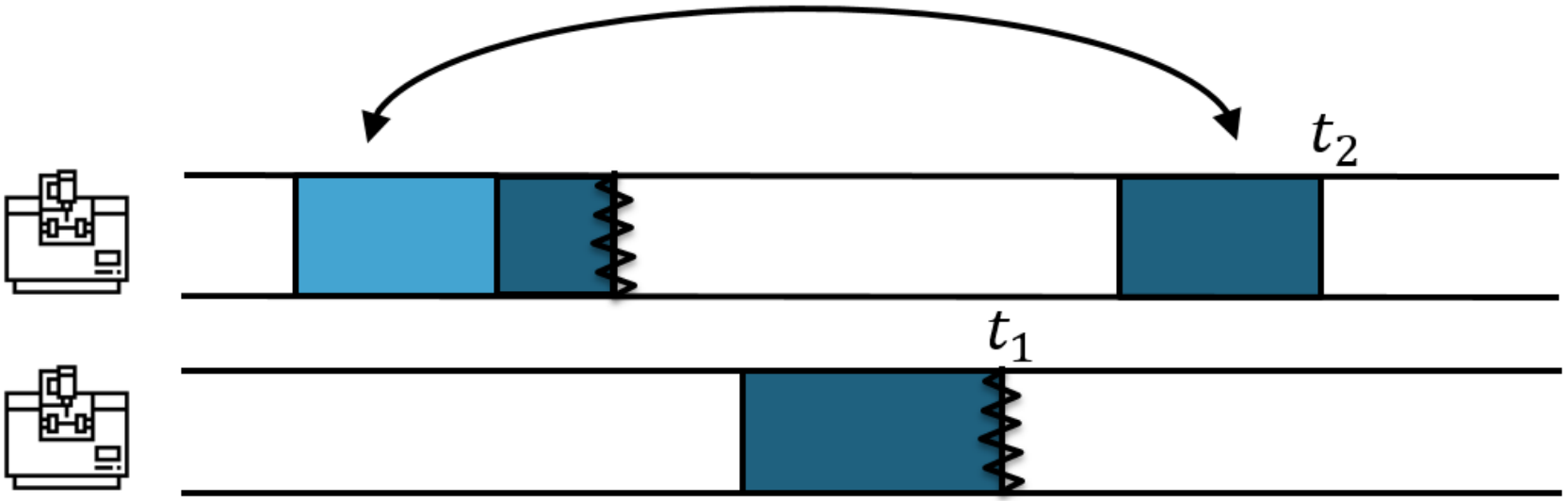
In any optimal schedule for $P|partition, \text{prmp}|\sum_j C_j$, all jobs sharing the same resource must be processed in order of non-decreasing processing times.



Ordered Resources Property

Lemma 1

In any optimal schedule for $P|partition, \text{prmp}|\sum_j C_j$, all jobs sharing the same resource must be processed in order of non-decreasing processing times.

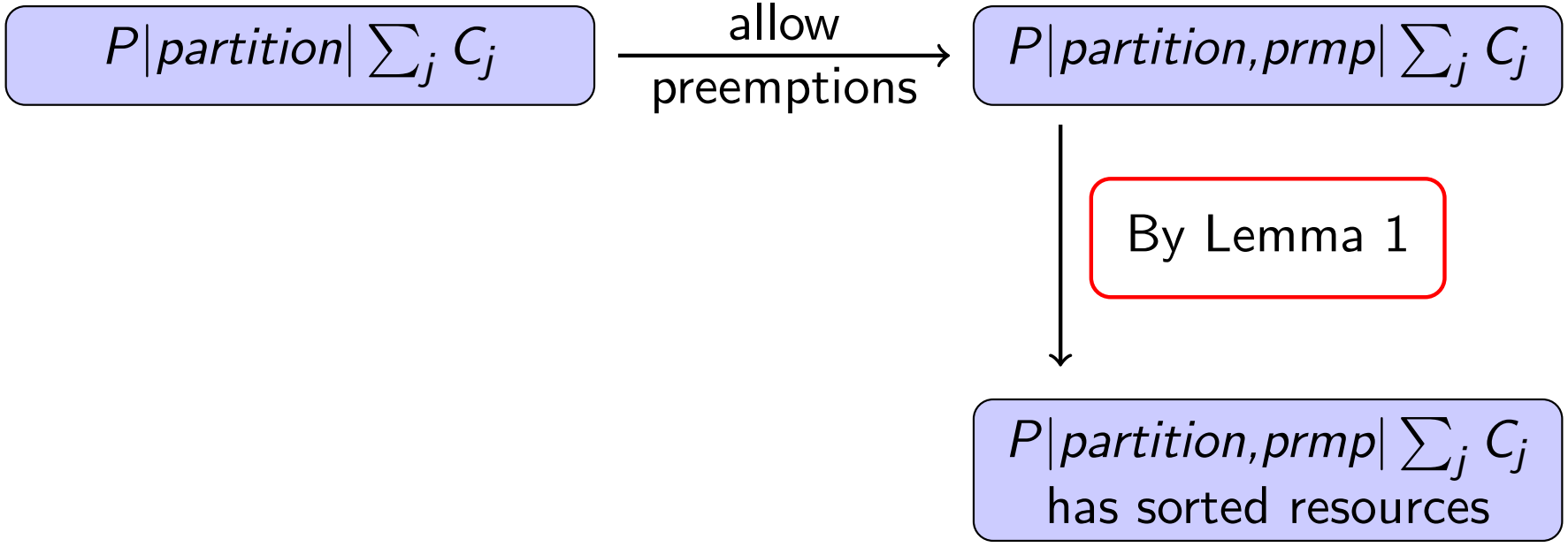


Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:



TU Delft TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Ordered Resources Property

Theorem 1

In any optimal schedule for $P|partition|\sum_j C_j$, all jobs sharing the same resource are processed in order of non-decreasing processing times.

Outline of proof:

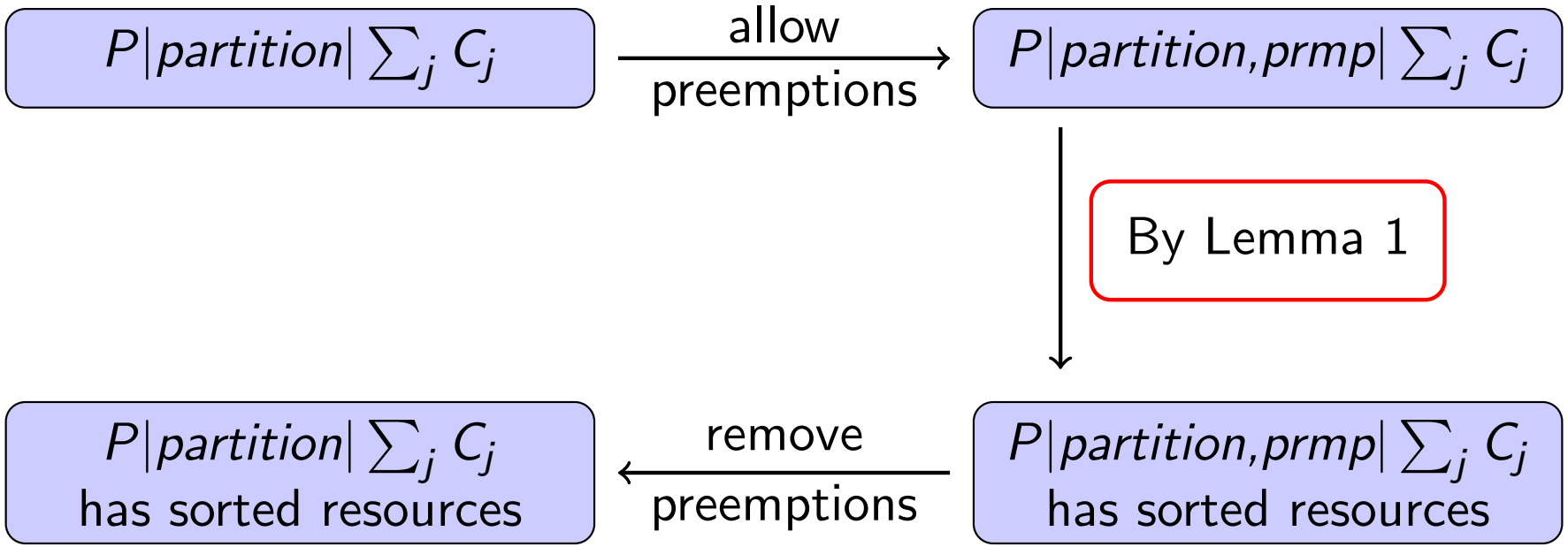


Table of Contents

1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

Practical Solution

In practice, machines are not identical (but unrelated). We proved this problem to be *NP*-hard.

- In practice, every resource (reticle) is used at most 30 times a month
  - Solve easier problem without resources
  - Make schedule feasible
- Results:
  - Gives 6.02% reduction in total completion time
  - Gives 1.97% reduction in machine usage

Table of Contents

1. Problem Definition
2. SPT-available Rule
3. Ordered Resources Property
4. Practical Solution
5. Future Work

Future Work

- Establish the complexity of $P|partition|\sum_j C_j$
- Establish worst-case approximation ratio for SPT-*available* rule.
- Is $P|partition|\sum_j C_j$ fixed-parameter tractable in k (the number of resources)?

References

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979).
Optimization and approximation in deterministic sequencing and scheduling.
A Survey, Annals of Discrete Mathematics, 5:287–326.

Smith, W. E. (1956).
Various optimizers for single-stage production.
Naval Research Logistics Quarterly, 3(1-2):59–66.

Parallel Machine Scheduling

with a Single Resource per Job

T. Janssen[1] C. Swennenhuis[2] A. Bitar[3] T. Bosman[4]
S. Dauzère-Pérès[5] D. Gijswijt[1] L. van Iersel[1] C. Yugma[5]

[1]Delft Institute of Applied Mathematics, Delft University of Technology

[2]Department of Mathematics and Computer Science, Eindhoven University of Technology

[3]Decision Brain, Montpelleir

[4]Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam

[5]Department of Manufacturing Sciences and Logistics, Center of Microelectronics in Provence

MAPSP, June 2019