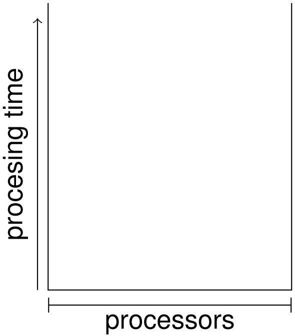
Approximation schemes for machine scheduling with resource (in-)dependent processing times

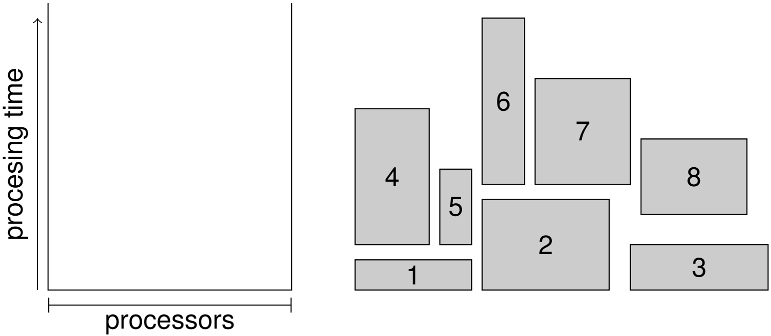
Klaus Jansen, Marten Maack, *Malin Rau*

April 4, 2016

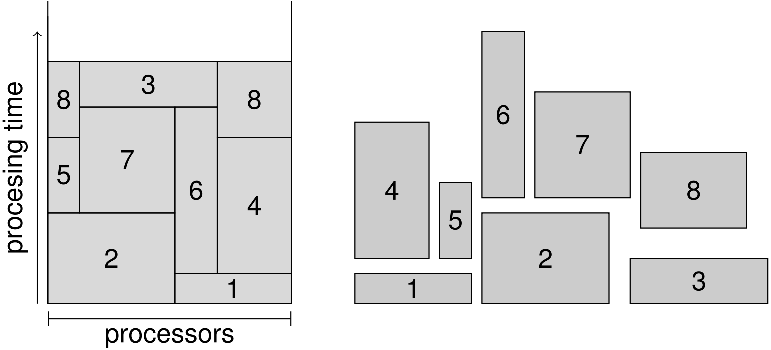
Scheduling parallel Tasks



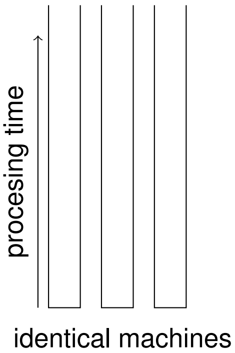
Scheduling parallel Tasks



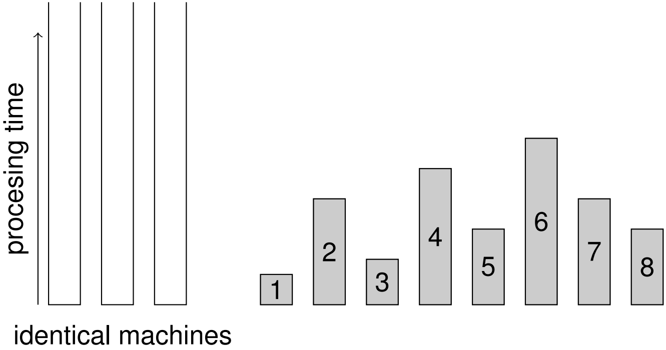
Scheduling parallel Tasks



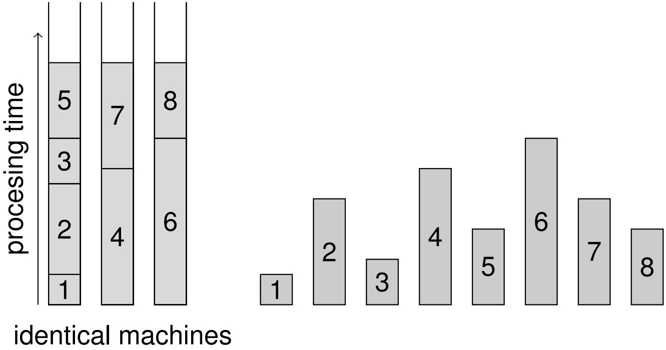
Scheduling on identical machines



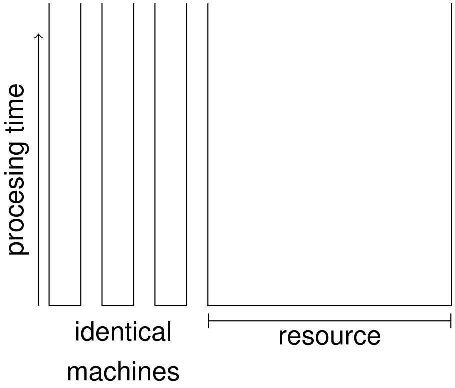
Scheduling on identical machines



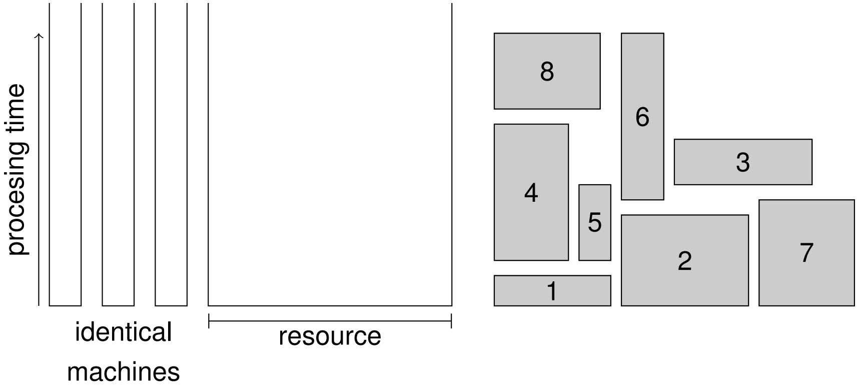
Scheduling on identical machines



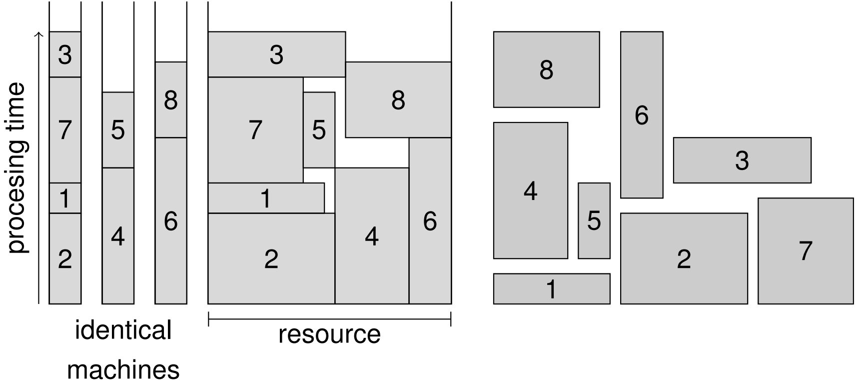
Scheduling on identical machines with one extra resource

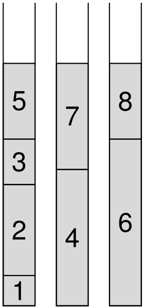


Scheduling on identical machines with one extra resource



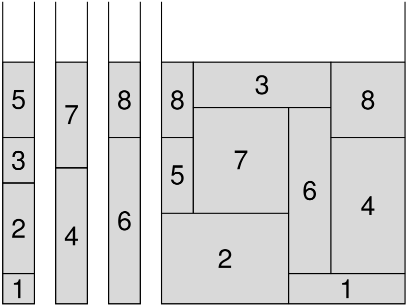
Scheduling on identical machines with one extra resource





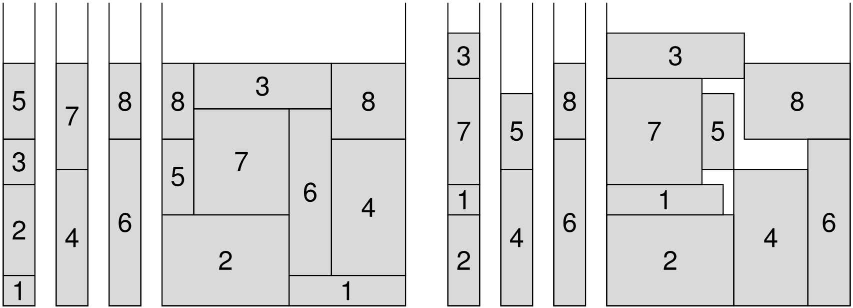
identical

machines



identical machines

parallel tasks



identical machines

parallel tasks

additional resource

Problem Definition

Given:

- m identical machines,
- one resource of size $R \in \mathbb{N}$,
- a set of jobs $\mathcal{J} = \{1, \ldots, n\}$. Each job has
  - a processing time $p_j \in \mathbb{Q}$ and
  - a resource amount $r_j \in \mathbb{N}$ with $r_j \leq R$.

Problem Definition

Given:

- m identical machines,
- one resource of size $R \in \mathbb{N}$,
- a set of jobs $\mathcal{J} = \{1, \ldots, n\}$. Each job has
  - a processing time $p_j \in \mathbb{Q}$ and
  - a resource amount $r_j \in \mathbb{N}$ with $r_j \leq R$.

Goal: Find a schedule $\tau : \mathcal{J} \to \mathbb{Q}_{\geq 0}$ with minimum makespan, such that:

$$\forall t \geq 0 \quad \sum_{j : t \in [\tau(j), \tau(j)+p_j)} r_j \leq R,$$

$$\forall t \geq 0 \quad \sum_{j : t \in [\tau(j), \tau(j)+p_j)} 1 \leq m.$$

Known Results

- There is no approximation algorithm with absolute ratio < 1.5, unless $\mathcal{P} = \mathcal{NP}$ (Drozdowski 1995)
- The list scheduling algorithm has absolute approximation ratio $3 - 3/m$ (Garey and Graham 1975)
- There is a polynomial time approximation algorithm with absolute ratio $2 + \varepsilon$ (Niemeier, Wiese 2013)
- For the case of unit processing times there is an AFPTAS with $A(I) \leq (1+\varepsilon)OPT + \mathcal{O}(\min\{(1/\varepsilon^2)^{(1/\varepsilon)}, n\log(R)/\varepsilon + 1/\varepsilon^3\})$ (Epstein and Levin 2010)

New Result

Theorem

There is an asymptotic FPTAS for the resource constrained scheduling problem with

$$A(I) \leq (1+\varepsilon)\mathrm{OPT}(I) + \mathcal{O}(1/\varepsilon^2)p_{\max},$$

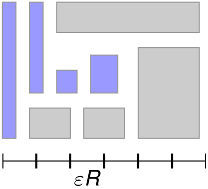
where $p_{\max}$ is the maximal processing time in the set of jobs.

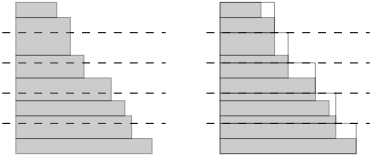
Algorithm Overview

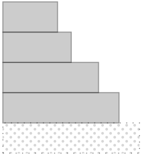
1. Simplify the instance and find an approximative preemptive schedule via a configuration LP.
2. Generalize the configurations by considering windows for narrow jobs.
3. Transform the preemptive schedule into a solution of a LP with generalized configurations.
4. Reduce the number of generalized configurations and windows in the LP solution.
5. Generate an integral solution.

Simplifying

- Partition $\mathcal{J}$ into wide jobs $\mathcal{J}_W$ and narrow jobs $\mathcal{J}_N$.
- Reduce the number of different wide resource amounts to $1/\varepsilon^2$.
- Glue together wide jobs with same resource amount.
- Discard widest group
- Simplified instance: I^{sup}.







Definition Configuration

A configuration C is a multiset of jobs with:

- $\sum_j C(j) r_j \leq R$,
- $\sum_j C(j) \leq m$.

$C(j) \in \mathbb{N}$ says how often the job j is contained in C.

Let $\mathcal{C}$ be the set of all configurations.

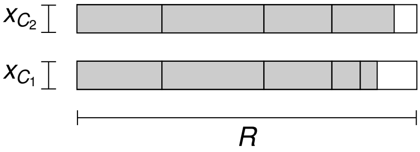
Definition Configuration

A configuration C is a multiset of jobs with:

- $\sum_j C(j) r_j \leq R$,
- $\sum_j C(j) \leq m$.

$C(j) \in \mathbb{N}$ says how often the job j is contained in C.

Let $\mathcal{C}$ be the set of all configurations.



Definition Configuration

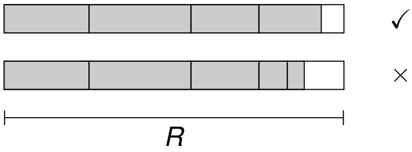
A configuration C is a multiset of jobs with:

- $\sum_j C(j) r_j \leq R$,
- $\sum_j C(j) \leq m$.

$C(j) \in \mathbb{N}$ says how often the job j is contained in C.

Let $\mathcal{C}$ be the set of all configurations.

$m = 4$



Preemptive Schedule

$$\min \sum_{C \in \mathcal{C}} x_C \qquad (1)$$

$$\sum_{C \in \mathcal{C}} C(j) x_C \geq p_j \quad \forall j \in \mathcal{J}^{\text{sup}} \qquad (2)$$

$$x_C \geq 0 \quad \forall C \in \mathcal{C} \qquad (3)$$

Preemptive Schedule

$$\min \sum_{C \in \mathcal{C}} x_C \tag{1}$$

$$\sum_{C \in \mathcal{C}} C(j) x_C \geq p_j \quad \forall j \in \mathcal{J}^{\text{sup}} \tag{2}$$

$$x_C \geq 0 \quad \forall C \in \mathcal{C} \tag{3}$$

- Can be solved approximately by max-min-resource-sharing.
- An approximate solution with at most $|\mathcal{J}^{\text{sup}}| + 1$ non zero components can be computed in polynomial time.

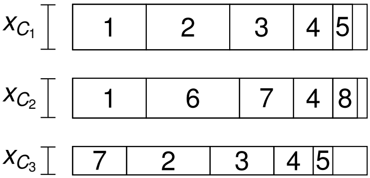
Generalized Configuration

- A *window* $w = (w_r, w_m)$ is a pair, where
  - w_r denotes its resource amount and
  - w_m its number of machines.

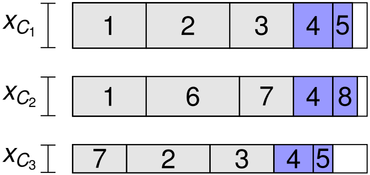
- A *generalized configuration* (C, w) is a pair consisting of
  - a configuration of wide jobs C and
  - a window w

  where
  - $\sum_{j \in \mathcal{J}_W^{\text{sup}}} C(j) r_j + w_r \leq R$ and
  - $\sum_{j \in \mathcal{J}_W^{\text{sup}}} C(j) + w_m \leq m$.

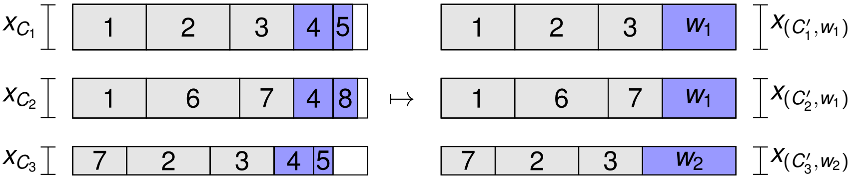
Generalized Preemptive Solution



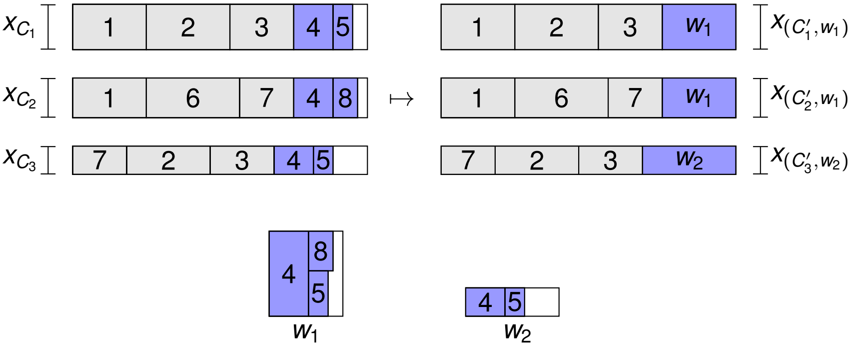
Generalized Preemptive Solution



Generalized Preemptive Solution



Generalized Preemptive Solution



$$y_{4,w_1} = x_{C_1} + x_{C_2}$$
$$y_{5,w_1} = x_{C_1}$$
$$y_{8,w_1} = x_{C_2}$$

$$y_{5,w_2} = x_{C_3}$$
$$y_{4,w_2} = x_{C_3}$$

Generalized Configuration LP

$$\sum_{C \in \mathcal{C}_W} \sum_{\substack{w \in \mathcal{W} \\ w \leq w(C)}} C(j) x_{(C,w)} \geq p_j, \qquad \forall j \in \mathcal{J}_W^{\text{sup}} \qquad (4)$$

$$\sum_{w \in \mathcal{W}} y_{j,w} \geq p_j, \qquad \forall j \in \mathcal{J}_N^{\text{sup}} \qquad (5)$$

$$w_m \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} y_{j,w}, \qquad \forall w \in \mathcal{W} \qquad (6)$$

$$w_r \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} r_j y_{j,w}, \qquad \forall w \in \mathcal{W} \qquad (7)$$

$$x_{(C,w)} \geq 0, \qquad \forall C \in \mathcal{C}_W, \forall w \in \mathcal{W} \qquad (8)$$

$$y_{j,w} \geq 0, \qquad \forall w \in \mathcal{W}, \forall j \in \mathcal{J}_N^{\text{sup}} \qquad (9)$$

Generalized Configuration LP

$$\sum_{C\in\mathcal{C}_W}\sum_{\substack{w\in\mathcal{W}\\ w\leq w(C)}} C(j)x_{(C,w)} \geq p_j, \qquad \forall j\in\mathcal{J}_W^{\text{sup}} \tag{4}$$

$$\sum_{w\in\mathcal{W}} y_{j,w} \geq p_j, \qquad \forall j\in\mathcal{J}_N^{\text{sup}} \tag{5}$$

$$w_m \sum_{\substack{C\in\mathcal{C}_W\\ w(C)\geq w}} x_{(C,w)} \geq \sum_{j\in\mathcal{J}_N} y_{j,w}, \qquad \forall w\in\mathcal{W} \tag{6}$$

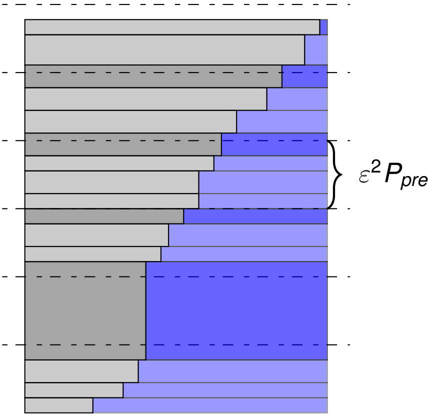
$$w_r \sum_{\substack{C\in\mathcal{C}_W\\ w(C)\geq w}} x_{(C,w)} \geq \sum_{j\in\mathcal{J}_N} r_j y_{j,w}, \qquad \forall w\in\mathcal{W} \tag{7}$$

$$x_{(C,w)} \geq 0, \qquad \forall C\in\mathcal{C}_W, \forall w\in\mathcal{W} \tag{8}$$

$$y_{j,w} \geq 0, \qquad \forall w\in\mathcal{W}, \forall j\in\mathcal{J}_N^{\text{sup}} \tag{9}$$

- Basic solution has $|\mathcal{J}_W^{\text{sup}}| + |\mathcal{J}_N^{\text{sup}}| + 2|\mathcal{W}|$ non zero components.
- At most $|\mathcal{J}_W^{\text{sup}}| + 2|\mathcal{W}|$ configurations and fractional jobs.
- $|\mathcal{J}_W^{\text{sup}}| \in \mathcal{O}(1/\varepsilon^2)$, $|\mathcal{W}| \in \mathcal{O}(\min\{|\mathcal{J}^{\text{sup}}|, |\mathcal{C}_W|\})$

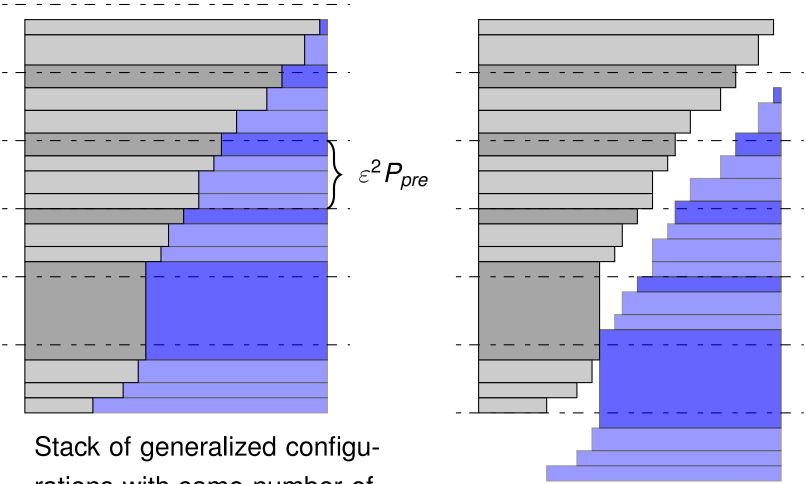
Reduce Number Of Windows



Stack of generalized configurations with same number of wide jobs

P_{pre}: length of preemptive schedule

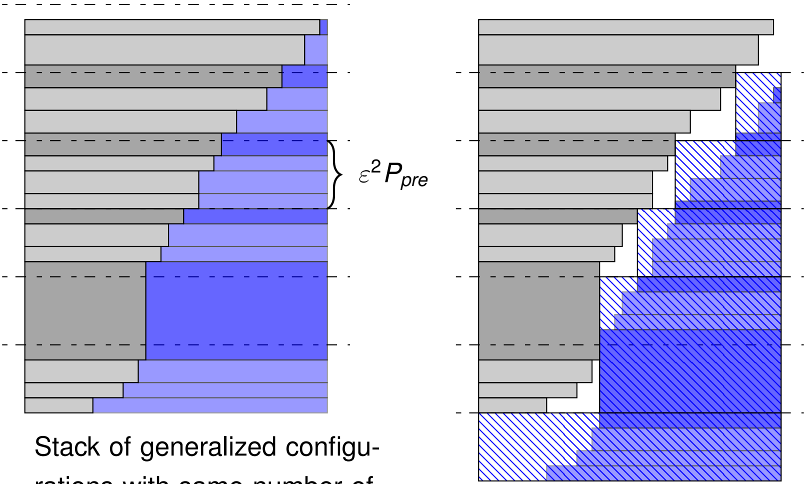
Reduce Number Of Windows



Stack of generalized configurations with same number of wide jobs

P_{pre}: length of preemptive schedule

Reduce Number Of Windows



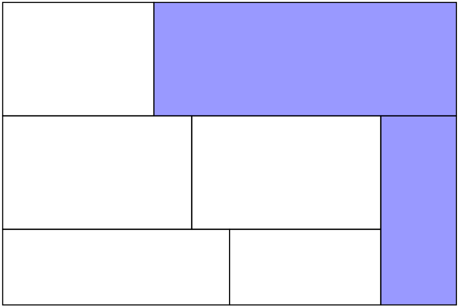
Stack of generalized configurations with same number of wide jobs

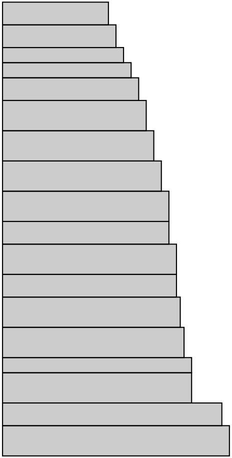
P_{pre}: length of preemptive schedule

Properties

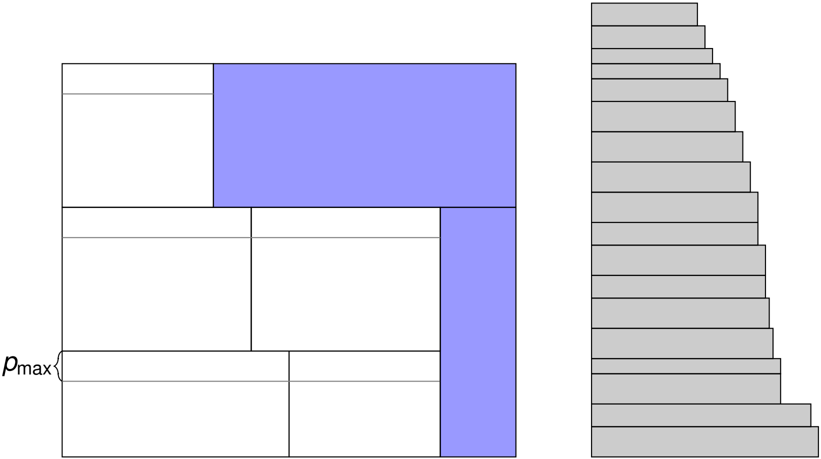
- $1/\varepsilon$ different stacks of generalized configurations.
- $\varepsilon^2 P_{pre}$ additional processing time per stack
- εP_{pre} additional total processing time
- Number of windows $\leq 1/\varepsilon^2 + 2$.
- Basic solution has $\mathcal{O}(1/\varepsilon^2)$ configurations and $\mathcal{O}(1/\varepsilon^2)$ fractional small jobs

Integral Schedule

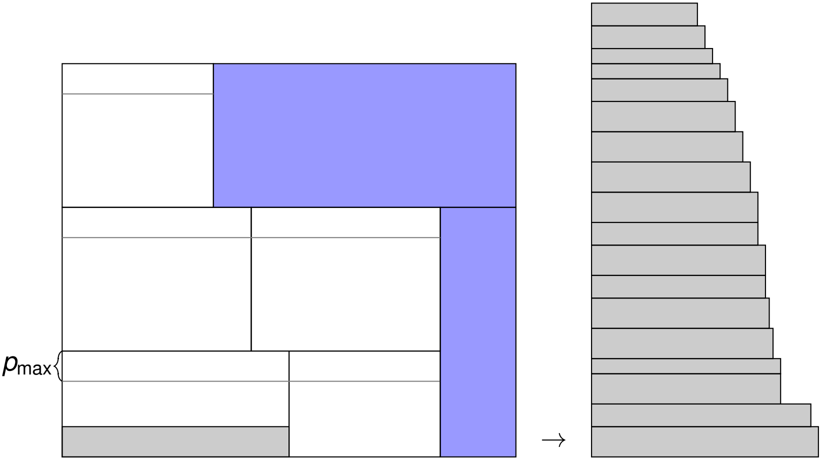




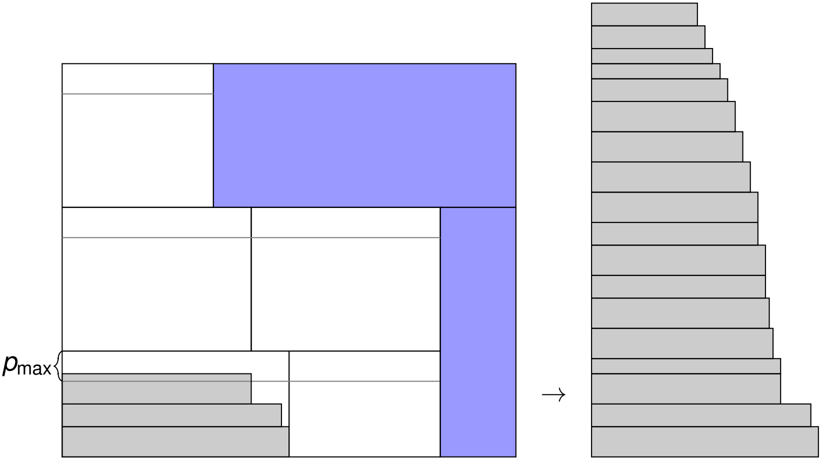
Integral Schedule



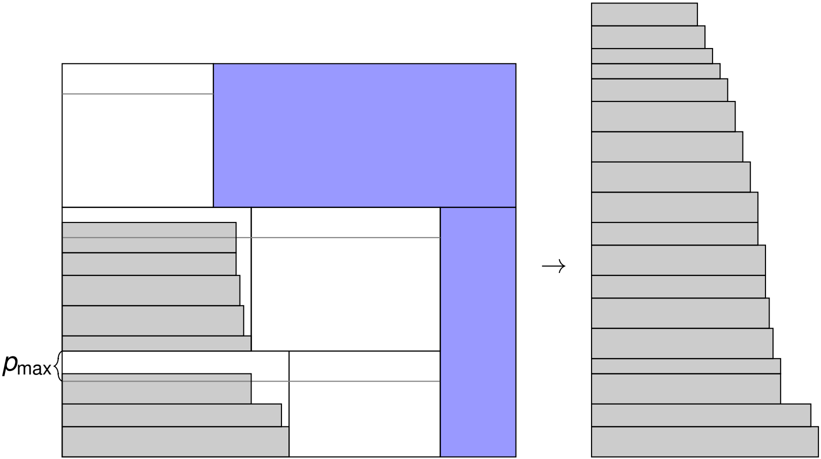
Integral Schedule



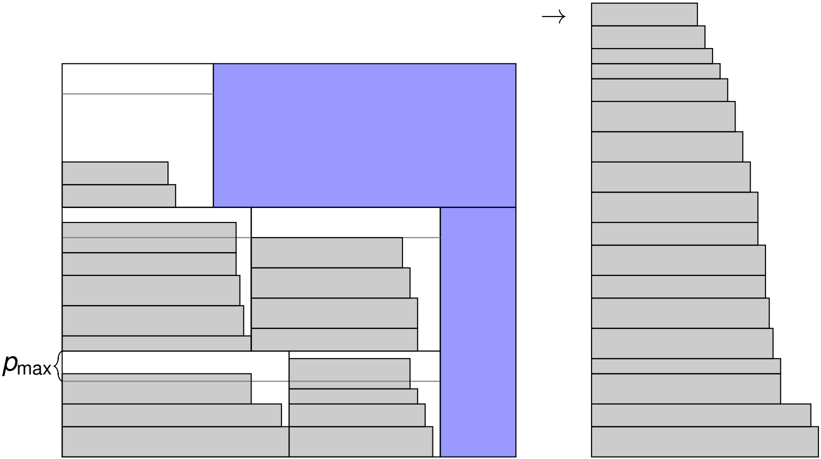
Integral Schedule



Integral Schedule

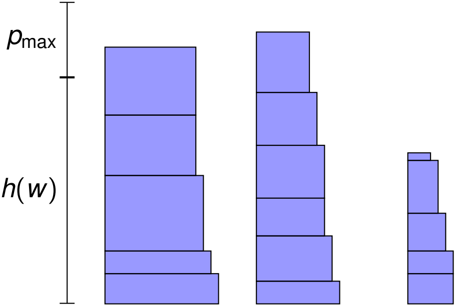


Integral Schedule



Integral Schedule

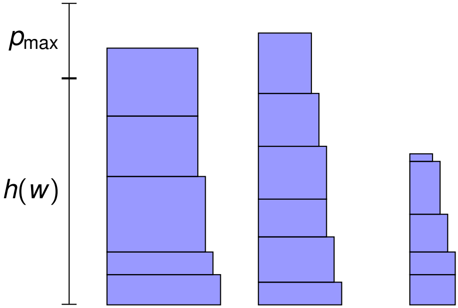
$$w_m \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} y_{j,w}$$



Integral Schedule

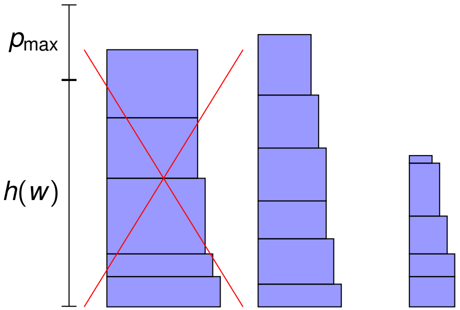
$$w_m \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} y_{j,w}$$

$$w_r \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} r_j y_{j,w}$$



Integral Schedule

$$w_m \sum_{\substack{C \in \mathcal{C}_W \\ w(C) \geq w}} x_{(C,w)} \geq \sum_{j \in \mathcal{J}_N} y_{j,w}$$



Scheduling With Resource Dependent Processing Times

Given:

- m identical machines,
- one resource of size $R \in \mathbb{N}$,
- a set of jobs $\mathcal{J} = \{1, \ldots, n\}$. Each job has a processing time function $\pi_j : \{0, \ldots, R\} \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$.

Goal: Find a resource allocation $\rho : \mathcal{J} \to \{0, \ldots, R\}$ and schedule $\tau : \mathcal{J} \to \mathbb{Q}_{\geq 0}$ with minimal makespan, such that:

$$\forall t \geq 0 \sum_{j : t \in [\tau(j), \tau(j) + \pi_j(\rho(j)))} \rho(j) \leq R,$$

$$\forall t \geq 0 \sum_{j : t \in [\tau(j), \tau(j) + \pi_j(\rho(j)))} 1 \leq m.$$

Known Results

- There is a $3.5 + \epsilon$ approximation algorithm for the scheduling problem on identical machines (Kellerer 2008).
- There is a $3 + \epsilon$ approximation algorithm for the scheduling problem where each job is pre-assigned to one machine (Grigoriev, Uetz 2009).
- There is a $3.75 + \epsilon$ approximation algorithm for the scheduling problem on unrelated machines (Grigoriev et al. 2007).

New Result

Theorem

There is an asymptotic FPTAS for the scheduling problem with resource dependent processing times with

$$A(I) \leq (1 + \varepsilon)\mathrm{OPT}(I) + O(1/\varepsilon^2)\pi_{\max}.$$

The running time of the algorithm is polynomially bounded in n,R, and $1/\varepsilon$*.*

Overview

- Compute preemptive schedule, where jobs are allowed to have several resource amounts
- Use the preemptive schedule to define an instance for the fixed resource variant
- Apply the steps of the fixed resource algorithm until a solution for the generalised LP is found.
- Use this solution to define unique resource allotment for almost all the original jobs
- Define a new solution to the LP using the unique resource allotment
- Define the integral schedule
- Schedule the jobs with no unique resource allotment on top

Future Work

Resource independent processing times:

- Is there an approximation algorithm A with $A(I) < (2 + \varepsilon)\mathrm{OPT}(I)$?
- Can we reduce the additive term to p_{max} and how far would this increase the running time of the algorithm?

Resource dependent processing times:

- Can the additional factor π_{max} be reduced?
- Can we improve on the result for unrelated machines?