Approximation in Deterministic and Stochastic Machine Scheduling

Wissenschaftliche Aussprache

May 06, 2021

M.Sc. Sven Joachim Jäger

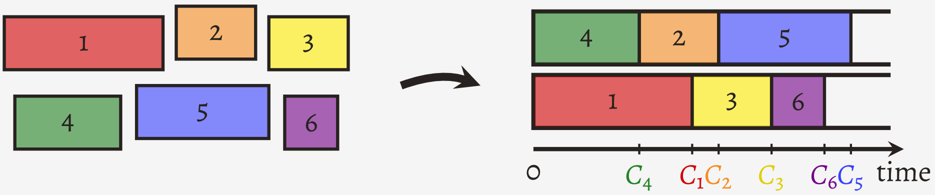
Fakultät II – Mathematik und Naturwissenschaften, Technische Universität Berlin

Vorsitzender: Prof. Dr. Yuri Suris
Gutachter: Prof. Dr. Martin Skutella
Prof. Dr. Marc Uetz

Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \ldots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

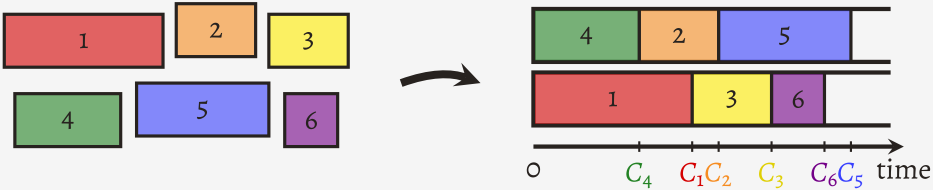
Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \ldots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



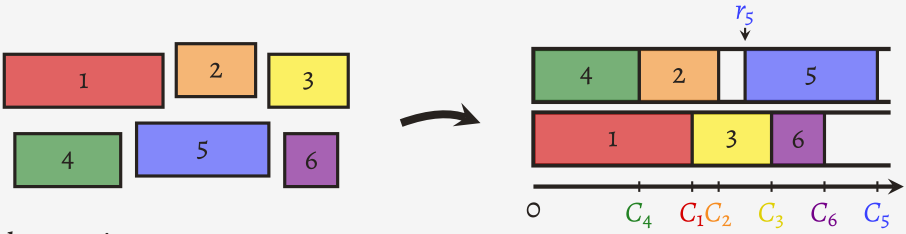
Complexity

- NP-hard for two machines (Bruno et al. 1974)
- strongly NP-hard for arbitrary number of machines (Lageweg, Lenstra 1977)

Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \ldots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



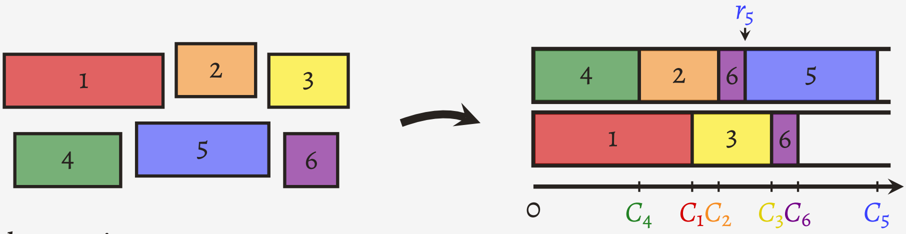
Problem variants

- release dates $r_j \geq 0$

Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \dots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



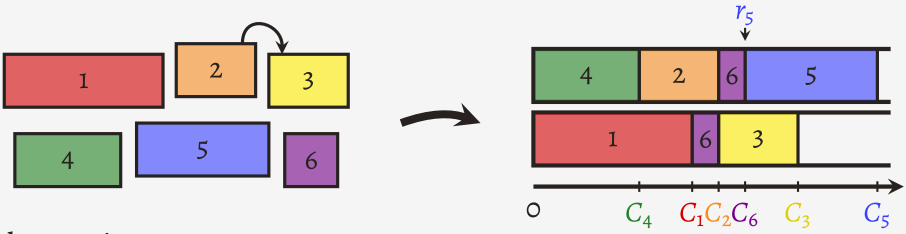
Problem variants

- release dates $r_j \geq 0$
- preemptive vs. non-preemptive

Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \ldots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



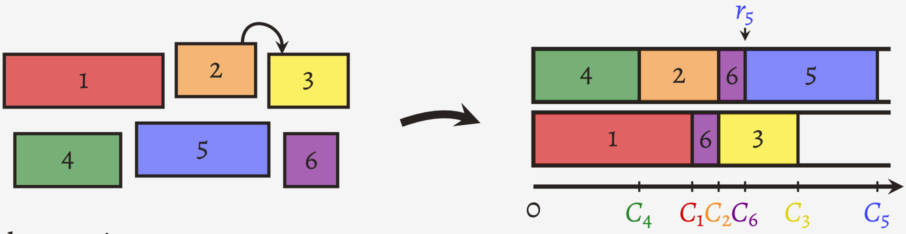
Problem variants

- release dates $r_j \geq 0$
- preemptive vs. non-preemptive
- precedence constraints/delays

Scheduling on Identical Parallel Machines

Given: jobs $j = 1, \ldots, n$ with processing times $p_j \geq 0$ and weights $w_j \geq 0$, number of machines m;

Task: schedule each job for p_j time units on some machine so as to minimize the sum of weighted completion times $\sum_{j=1}^{n} w_j \cdot C_j$.



Problem variants

- release dates $r_j \geq 0$
- preemptive vs. non-preemptive
- precedence constraints/delays
- heterogeneous machines

Scheduling under Uncertainty

I. Adversarial models
  - online job arrivals
  - unknown processing times (non-clairvoyant scheduling)

Competitive analysis: compare to offline optimum

Scheduling under Uncertainty

I. Adversarial models
- online job arrivals
- unknown processing times (non-clairvoyant scheduling)

Competitive analysis: compare to offline optimum

II. Stochastic models
- random processing times with known distributions

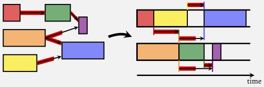
Minimize expected total weighted completion time

Overview

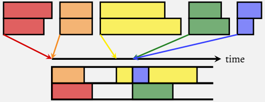
I. Deterministic Scheduling

WSPT Rule

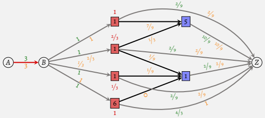
Scheduling with Precedence Delays



Online Non-Migratory Preemptive Scheduling on Unrelated Machines

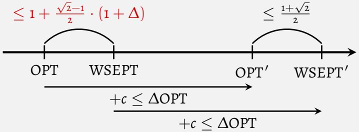


Non-Clairvoyant Online Scheduling with Precedence Delays

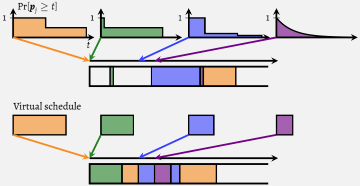


II. Stochastic Scheduling

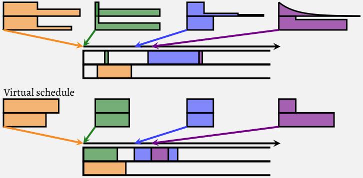
WSEPT Rule



Stochastic Scheduling on Single Machine with Release Dates



Stochastic Online Scheduling on Unrelated Machines



Part I

Deterministic Scheduling

Main Results

WSPT Rule

Scheduling with Precedence Delays

Online Non-Migratory Preemptive Scheduling on Unrelated Machines

Non-Clairvoyant Online Scheduling with Precedence Delays

Weighted Shortest Processing Time First (WSPT) rule

Whenever a machine is free, start available job with maximum ratio w_j/p_j on it.

For a fixed number m of machines, what is the maximum possible approximation ratio of the WSPT rule?

WSPT Rule – Previous Work

Theorem (Kawaguchi, Kyan 1986)

For an arbitrary number of machines WSPT rule has tight approximation ratio $\frac{1+\sqrt{2}}{2} \approx 1.207$.

WSPT Rule – Previous Work

Theorem (Kawaguchi, Kyan 1986)

For an arbitrary number of machines WSPT rule has tight approximation ratio $\frac{1+\sqrt{2}}{2} \approx 1.207$.

- Worst case is attained if all jobs have $w_j = p_j$.
- In this case WSPT is list scheduling according to the input order.
- Worst case is attained when $n, m \to \infty$.

WSPT Rule – Previous Work

Theorem (Kawaguchi, Kyan 1986)

For an arbitrary number of machines WSPT rule has tight approximation ratio $\frac{1+\sqrt{2}}{2} \approx 1.207$.

- Worst case is attained if all jobs have $w_j = p_j$.
- In this case WSPT is list scheduling according to the input order.
- Worst case is attained when $n, m \to \infty$.

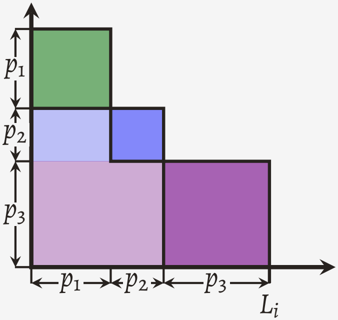
- There is a PTAS for scheduling jobs on an arbitrary number of identical machines. (Skutella, Woeginger 2000)
- For any number m of machines there is an FPTAS for scheduling jobs on m identical machines. (Sahni 1976)

Reduction to Unit Smith Ratio

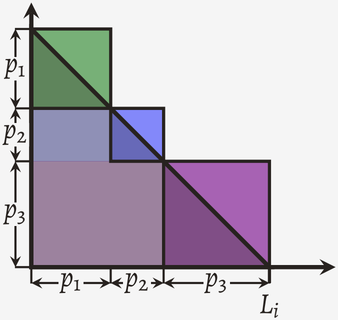
Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ *for all* j.

Objective Function in Terms of Machine Loads (for $w_j = p_j$)



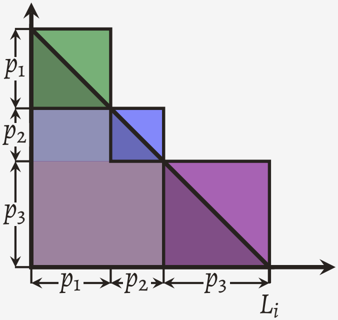
Objective Function in Terms of Machine Loads (for $w_j = p_j$)



one machine i:

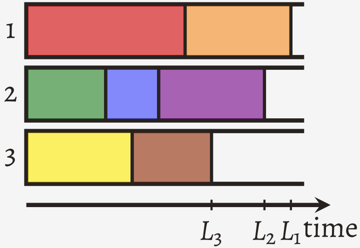
$$\sum_{j \to i} p_j \cdot C_j = \frac{1}{2} \Big(\underbrace{\sum_{j \to i} p_j}_{=:L_i} \Big)^2 + \frac{1}{2} \sum_{j \to i} p_j^2$$

Objective Function in Terms of Machine Loads (for $w_j = p_j$)



one machine i:

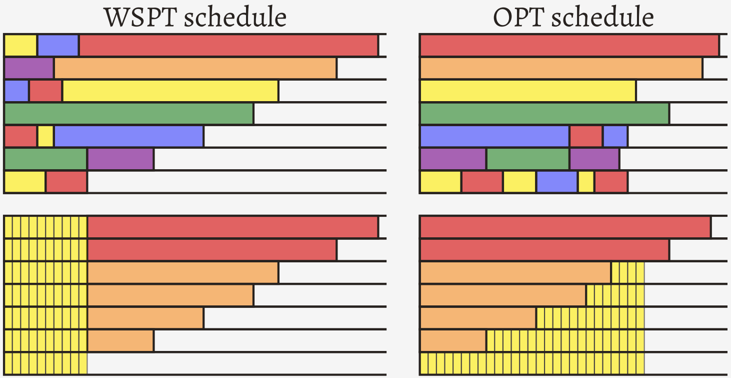
$$\sum_{j \to i} p_j \cdot C_j = \frac{1}{2} \Bigg(\underbrace{\sum_{j \to i} p_j}_{=:L_i} \Bigg)^2 + \frac{1}{2} \sum_{j \to i} p_j^2$$



m-machine schedule:

$$\sum_{j=1}^{n} p_j \cdot C_j = \frac{1}{2} \sum_{i=1}^{m} L_i^2 + \frac{1}{2} \sum_{j=1}^{n} p_j^2$$

Reduction to Worst Case



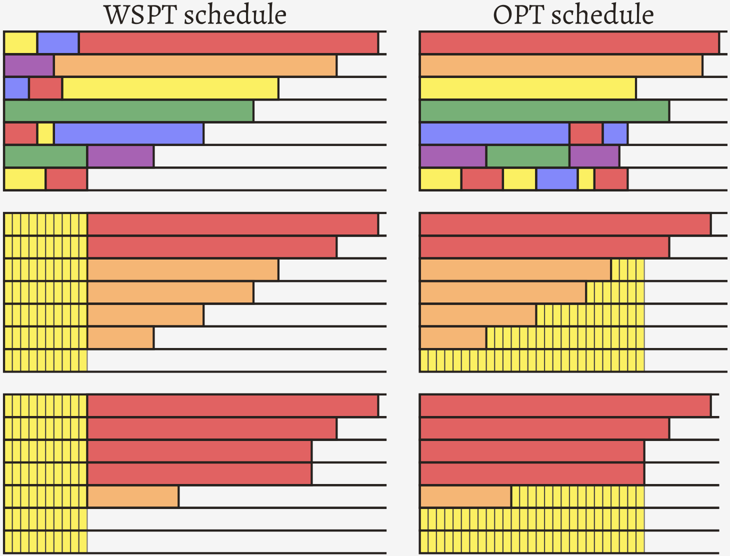
$\sum_i L_i^2$ unchanged

$\sum_j p_j^2$ decreases by δ

$\sum_i L_i^2$ not increased

$\sum_j p_j^2$ decreases by δ

Reduction to Worst Case



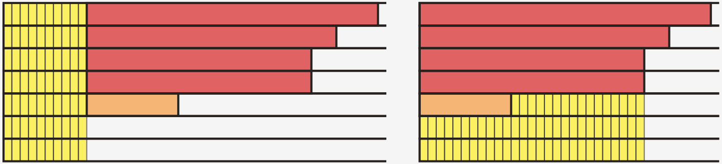
$\sum_i L_i^2$ unchanged
$\sum_j p_j^2$ decreases by δ

$\sum_i L_i^2$ not increased
$\sum_j p_j^2$ decreases by δ

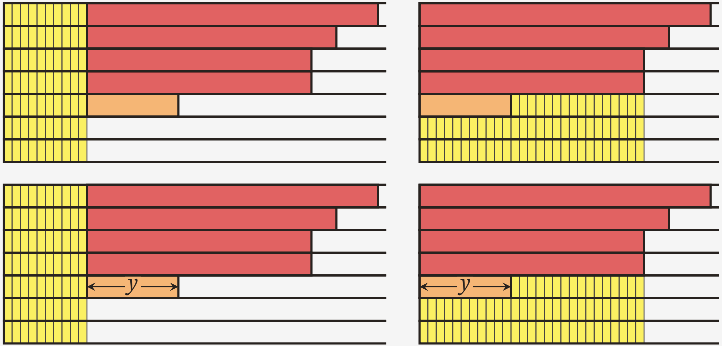
$\sum_i L_i^2$ increases by δ
$\sum_j p_j^2$ increases by δ

$\sum_i L_i^2$ unchanged
$\sum_j p_j^2$ increases by δ

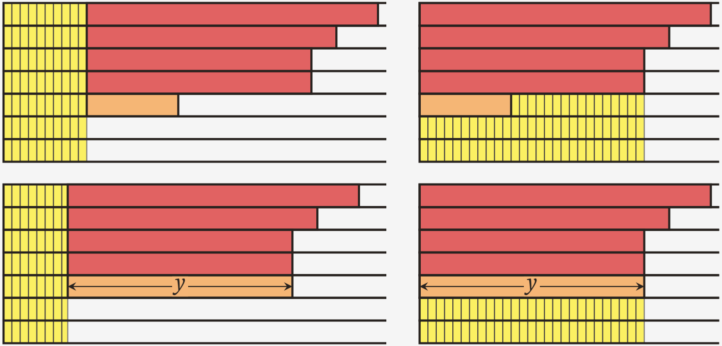
Reduction to Worst Case (Cont.)



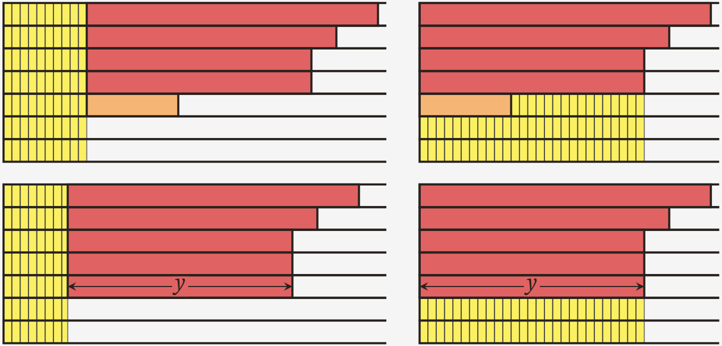
Reduction to Worst Case (Cont.)



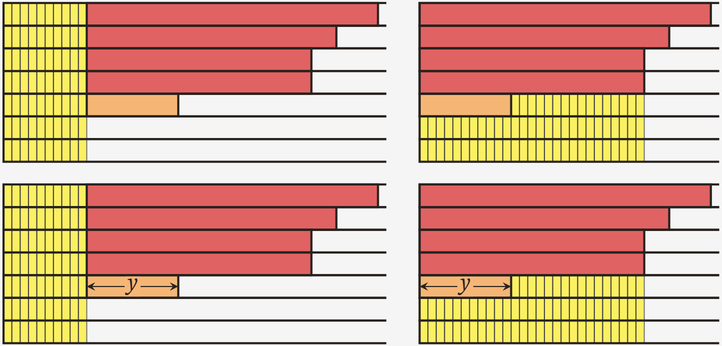
Reduction to Worst Case (Cont.)



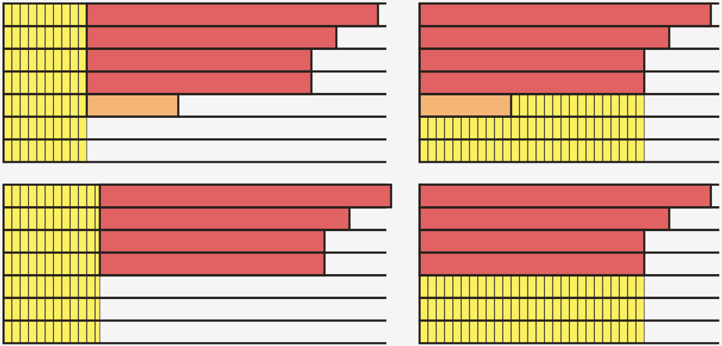
Reduction to Worst Case (Cont.)



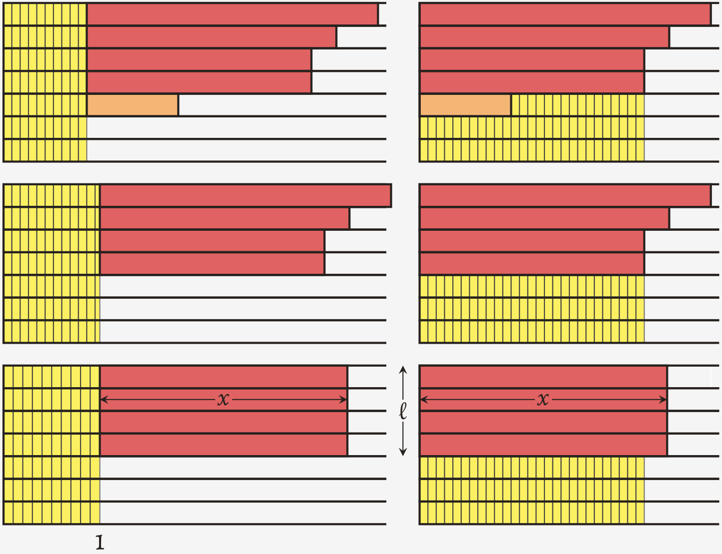
Reduction to Worst Case (Cont.)



Reduction to Worst Case (Cont.)



Reduction to Worst Case (Cont.)



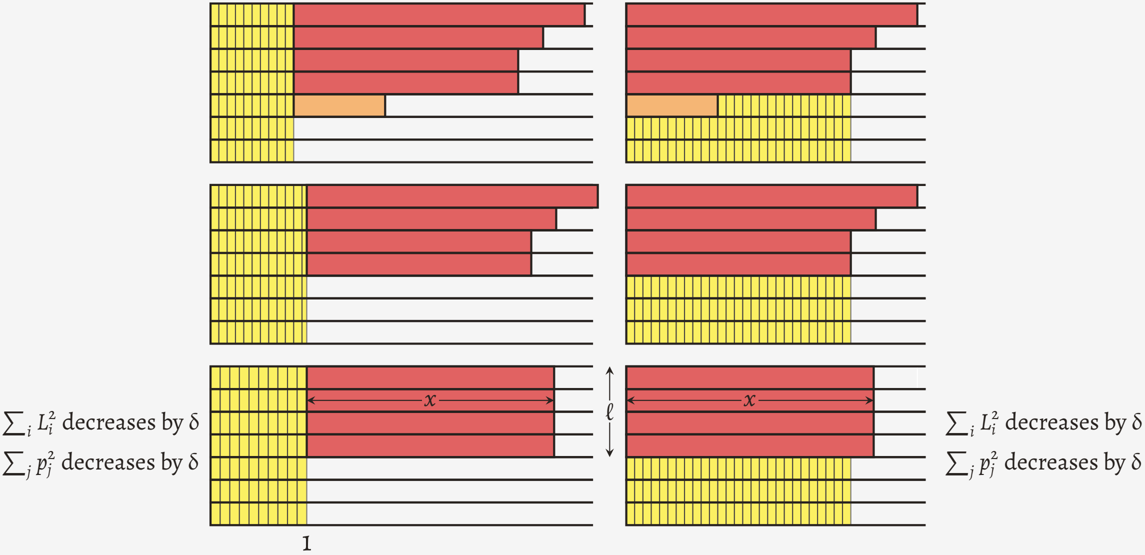
$\sum_i L_i^2$ decreases by δ

$\sum_j p_j^2$ decreases by δ

$\sum_i L_i^2$ decreases by δ

$\sum_j p_j^2$ decreases by δ

Reduction to Worst Case (Cont.)



Find worst ℓ, x by calculation.

Approximation Ratio of the WSPT Rule on Fixed Number of Machines

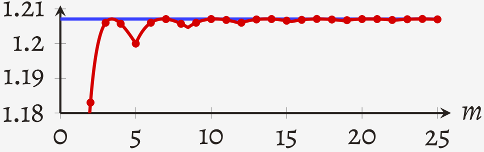
Theorem

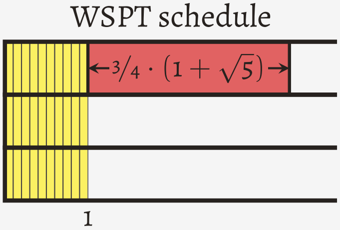
On $m \geq 2$ machines the worst case for the WSPT rule occurs for

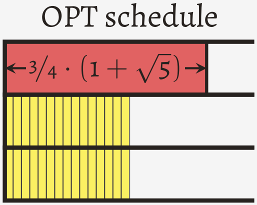
1. m/ε *small jobs with* $w_j = p_j = \varepsilon \quad (\varepsilon \to 0)$,
2. $\ell := \left\lfloor \frac{2m - \sqrt{2m^2-1}}{2} \right\rceil$ *jobs with* $w_j = p_j = \frac{m}{\sqrt{\ell \cdot (2m-\ell)} - \ell}$.

In this case the approximation ratio is

$$1 + \frac{\sqrt{\ell \cdot (2m - \ell)} - \ell}{2m}.$$







Main Results

WSPT Rule

Scheduling with Precedence Delays

Online Non-Migratory Preemptive Scheduling on Unrelated Machines

Non-Clairvoyant Online Scheduling with Precedence Delays

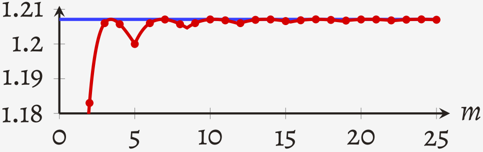
Theorem

On $m \geq 2$ machines the worst case for the WSPT rule occurs for

1. m/ε *small jobs with* $w_j = p_j = \varepsilon \quad (\varepsilon \to 0)$,
2. $\ell := \left\lfloor \frac{2m - \sqrt{2m^2 - 1}}{2} \right\rceil$ *jobs with* $w_j = p_j = \frac{m}{\sqrt{\ell \cdot (2m - \ell)} - \ell}$.

In this case the approximation ratio is

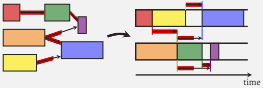
$$1 + \frac{\sqrt{\ell \cdot (2m - \ell)} - \ell}{2m}.$$



Main Results

WSPT Rule

Scheduling with Precedence Delays



Online Non-Migratory Preemptive Scheduling on Unrelated Machines

Non-Clairvoyant Online Scheduling with Precedence Delays

Theorem (Li 2020)

There is a 3.386-approximation algorithm for scheduling jobs with precedence constraints.

Theorem

There is a 3.386-approximation algorithm for scheduling jobs with precedence delays.

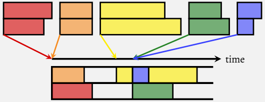
- includes the setting with precedence constraints and release dates.

Main Results

WSPT Rule

Scheduling with Precedence Delays

Online Non-Migratory Preemptive Scheduling on Unrelated Machines



Non-Clairvoyant Online Scheduling with Precedence Delays

Theorem

The Greedy-Assignment WSRPT algorithm is a 4-competitive deterministic online algorithm for non-migratory preemptive scheduling on unrelated machines.

Previous results

- 7.216-competitive deterministic online algorithm (Gupta et al. 2021+)
- 5.771-competitive randomized online algorithm (Hall et al. 1997; Chakrabarti et al. 1996)
- With $1+\varepsilon$ speed augmentation, Greedy-Assignment WSRPT algorithm is $\left(2+\frac{2}{\varepsilon}\right)$-competitive for $\sum w_j(C_j - r_j)$. (Anand et al. 2012)
  $\rightsquigarrow$ 8-competitive alg. for $\sum w_j C_j$. (Bansal, Pruhs 2004)

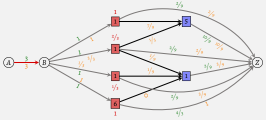
Main Results

WSPT Rule

Scheduling with Precedence Delays

Online Non-Migratory Preemptive Scheduling on Unrelated Machines

Non-Clairvoyant Online Scheduling with Precedence Delays



Theorem (Garg et al. 2019)

There is a 10-competitive deterministic non-clairvoyant online algorithm for jobs with precedence constraints.

Theorem

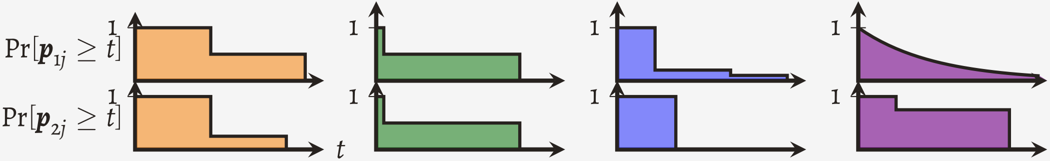
A simplification of the algorithm of Garg et al. is 8-competitive for jobs with precedence delays.

Part II

Stochastic Scheduling

Stochastic Scheduling on Unrelated Parallel Machines

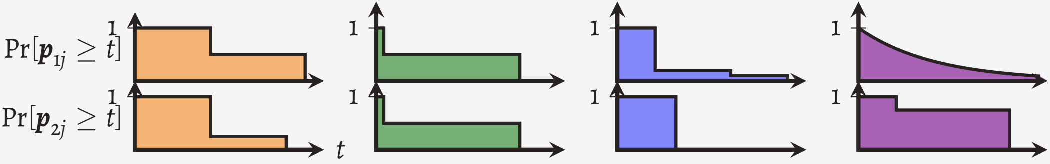
Given: weights $w_j \geq 0$ and distributions of independent random processing times $\boldsymbol{p}_{ij} > 0$ of jobs $j = 1, \dots, n$ on machines $i = 1, \dots, m$



Task: find non-preemptive scheduling policy Π minimizing the expected sum of weighted completion times.

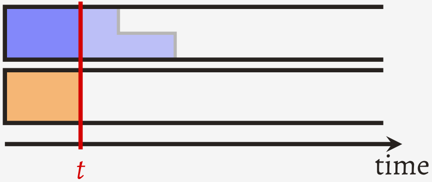
Stochastic Scheduling on Unrelated Parallel Machines

Given: weights $w_j \geq 0$ and distributions of independent random processing times $\boldsymbol{p}_{ij} > 0$ of jobs $j = 1, \ldots, n$ on machines $i = 1, \ldots, m$



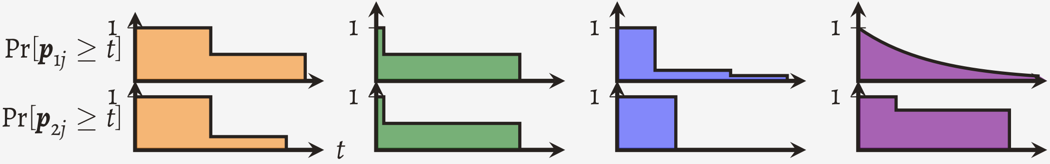
Task: find non-preemptive scheduling policy Π minimizing the expected sum of weighted completion times.

- must be non-anticipative, i.e., a decision made at time t may only depend on the information known at time t



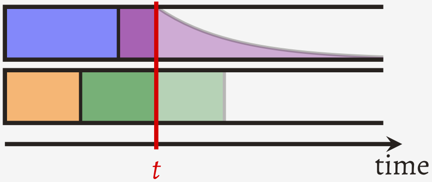
Stochastic Scheduling on Unrelated Parallel Machines

Given: weights $w_j \geq 0$ and distributions of independent random processing times $\boldsymbol{p}_{ij} > 0$ of jobs $j = 1, \ldots, n$ on machines $i = 1, \ldots, m$



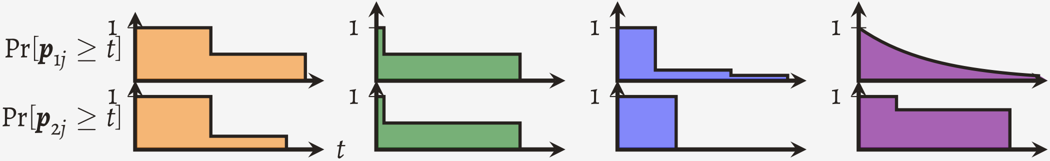
Task: find non-preemptive scheduling policy Π minimizing the expected sum of weighted completion times.

- must be non-anticipative, i.e., a decision made at time t may only depend on the information known at time t



Stochastic Scheduling on Unrelated Parallel Machines

Given: weights $w_j \geq 0$ and distributions of independent random processing times $\boldsymbol{p}_{ij} > 0$ of jobs $j = 1, \ldots, n$ on machines $i = 1, \ldots, m$

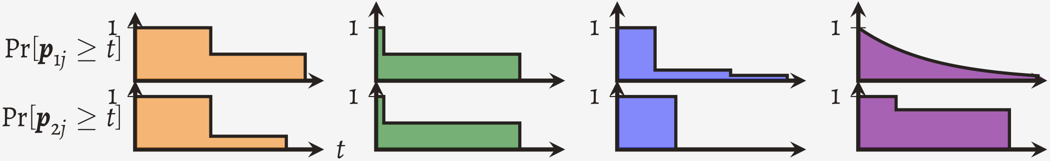


Task: find non-preemptive scheduling policy Π minimizing the expected sum of weighted completion times.

- must be non-anticipative, i.e., a decision made at time t may only depend on the information known at time t
- function $\mathbb{R}^{mn} \to \mathbb{R};\ (p_{ij}) \mapsto \sum_{j=1}^{n} w_j \cdot C_j^{\Pi}(p_{ij})$ must be measurable.

Stochastic Scheduling on Unrelated Parallel Machines

Given: weights $w_j \geq 0$ and distributions of independent random processing times $\boldsymbol{p}_{ij} > 0$ of jobs $j = 1, \ldots, n$ on machines $i = 1, \ldots, m$



Task: find non-preemptive scheduling policy Π minimizing the expected sum of weighted completion times.

- must be non-anticipative, i.e., a decision made at time t may only depend on the information known at time t
- function $\mathbb{R}^{mn} \to \mathbb{R};\ (p_{ij}) \mapsto \sum_{j=1}^{n} w_j \cdot C_j^{\Pi}(p_{ij})$ must be measurable.

Alternative definitions of scheduling policy (cp. Möhring, Radermacher 1985)

A) function $\mathbb{R}^{mn} \to \mathbb{R}^n_{\geq 0} \times [m]^n$;

B) function from state space to action space.

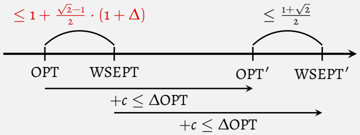
Approximative Policies

- We are interested in simple approximative policies.
- Some performance guarantees depend on an upper bound Δ on the squared coefficients of variation

$$\mathrm{CV}[\boldsymbol{p}_{ij}]^2 = \frac{\mathrm{Var}[\boldsymbol{p}_{ij}]}{\mathrm{E}[\boldsymbol{p}_{ij}]^2}.$$

Main Results

WSEPT Rule



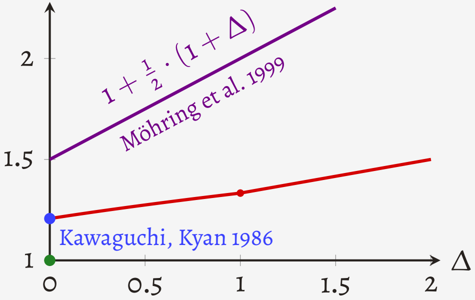
Stochastic Scheduling on Single Machine with Release Dates

Stochastic Online Scheduling on Unrelated Machines

Theorem

The WSEPT rule has performance guarantee

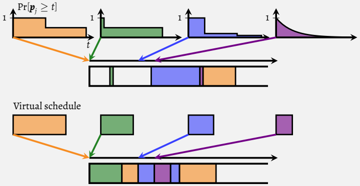
$$\begin{cases} 1 + \frac{1}{2\cdot(1+\sqrt{2(1+\Delta)})} \cdot (1+\Delta) & \text{if } \Delta \leq 1; \\ 1 + \frac{1}{6} \cdot (1+\Delta) & \text{if } \Delta \geq 1. \end{cases}$$



Main Results

WSEPT Rule

Stochastic Scheduling on Single Machine with Release Dates

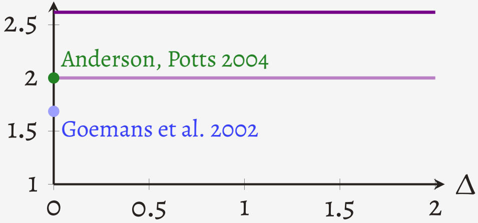


Stochastic Online Scheduling on Unrelated Machines

Theorem (Schulz 2008)

There are a $\left(1 + \max\left\{\phi, \frac{\phi+1}{2}(1+\Delta)\right\}\right)$*-competitive deterministic and a* $(2+\Delta)$*-competitive randomized online scheduling policy for identical machines.*

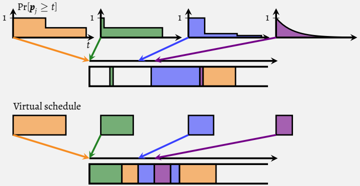
- on single machine $(\phi + 1)$-competitive (deterministic) and 2-competitive (randomized) $(\phi + 1 \approx 2.618)$



Main Results

WSEPT Rule

Stochastic Scheduling on Single Machine with Release Dates

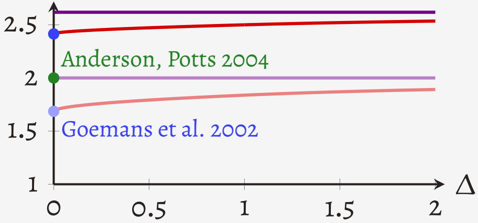


Stochastic Online Scheduling on Unrelated Machines

Theorem (Schulz 2008)

There are a $\left(1+\max\left\{\phi, \frac{\phi+1}{2}(1+\Delta)\right\}\right)$*-competitive deterministic and a* $(2+\Delta)$*-competitive randomized online scheduling policy for identical machines.*

- on single machine $(\phi+1)$-competitive (deterministic) and 2-competitive (randomized) $(\phi+1 \approx 2.618)$

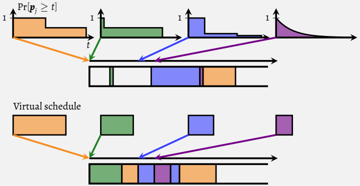


- If Δ is known in advance (*semi-online*), these can be improved.

Main Results

WSEPT Rule

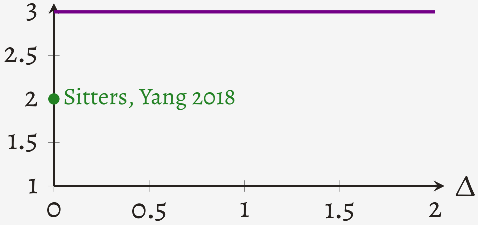
Stochastic Scheduling on Single Machine with Release Dates



Stochastic Online Scheduling on Unrelated Machines

Theorem (Möhring et al. 1999)

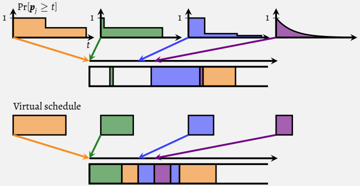
There is an efficient scheduling policy with performance guarantee 3 for scheduling jobs with precedence constraints and release dates on a single machine.



Main Results

WSEPT Rule

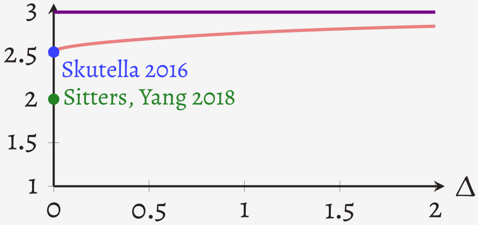
Stochastic Scheduling on Single Machine with Release Dates



Stochastic Online Scheduling on Unrelated Machines

Theorem (Möhring et al. 1999)

There is an efficient scheduling policy with performance guarantee 3 for scheduling jobs with precedence constraints and release dates on a single machine.

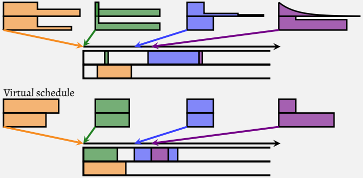


Main Results

WSEPT Rule

Stochastic Scheduling on Single Machine with Release Dates

Stochastic Online Scheduling on Unrelated Machines



Theorem (Gupta et al. 2021+)

There is a $3.608 \cdot h(\Delta) \cdot (2 + \Delta)$*-competitive deterministic online scheduling policy for unrelated machines, where*

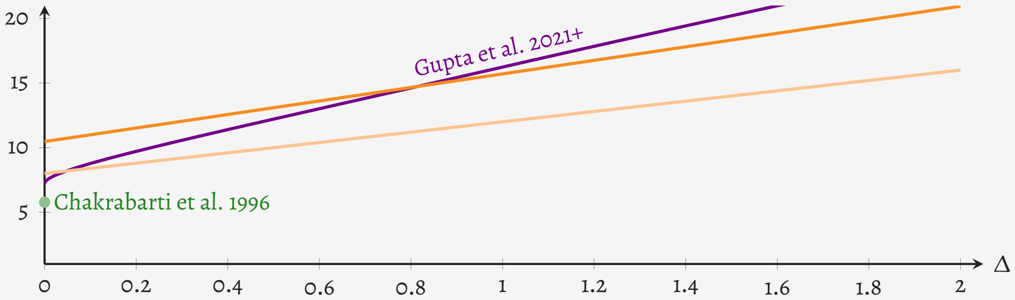
$$h(\Delta) := \begin{cases} 1 + \frac{\sqrt{\Delta}}{2} & \text{if } \Delta \leq 1; \\ 1 + \frac{\Delta}{\Delta+1} & \text{if } \Delta \geq 1. \end{cases}$$

Theorem

There is a $(3 + \sqrt{5}) \cdot (2 + \Delta)$*-competitive deterministic and an* $(8 + 4\Delta)$*-competitive randomized online scheduling policy for unrelated machines.*

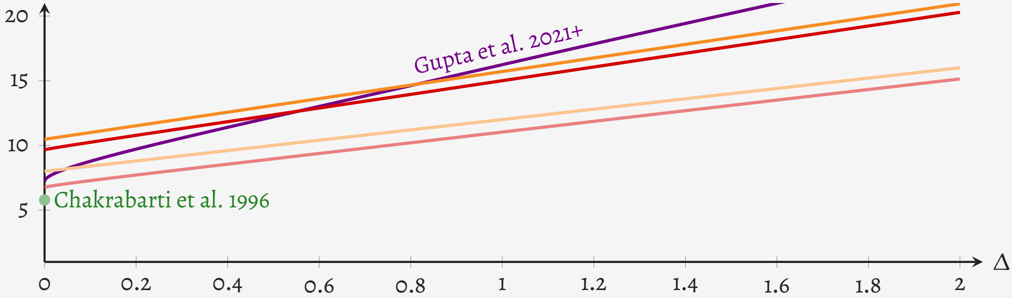
- If Δ is known in advance (semi-online), these can be improved.

Stochastic Online Scheduling on Unrelated Machines



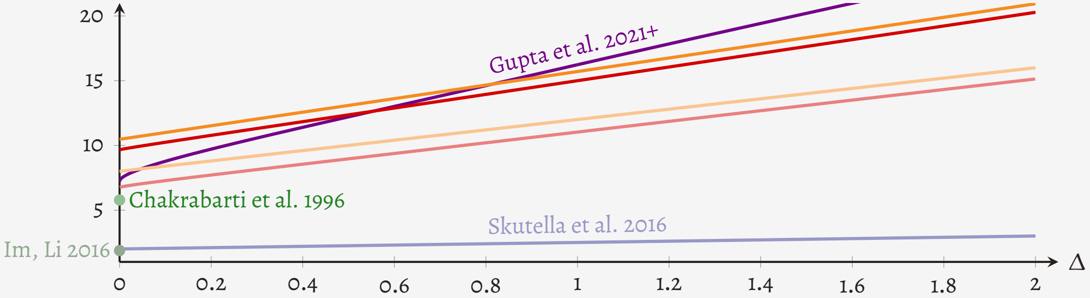
- Jobs j arrive over time;
- at the release date r_j the weight w_j and the distributions of all $\boldsymbol{p}_{ij}$, $i \in [m]$, are given;
- if j is scheduled on machine i, the outcome of $\boldsymbol{p}_{ij}$ becomes known when j completes.
- *Competitive analysis*: compare to optimal scheduling policy.

Stochastic Online Scheduling on Unrelated Machines



- Jobs j arrive over time;
- at the release date r_j the weight w_j and the distributions of all $\boldsymbol{p}_{ij}$, $i \in [m]$, are given;
- if j is scheduled on machine i, the outcome of $\boldsymbol{p}_{ij}$ becomes known when j completes.
- *Competitive analysis*: compare to optimal scheduling policy.

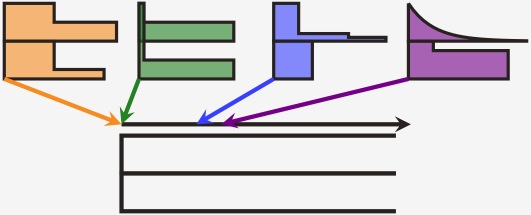
Stochastic Online Scheduling on Unrelated Machines



- Jobs j arrive over time;
- at the release date r_j the weight w_j and the distributions of all $\boldsymbol{p}_{ij}$, $i \in [m]$, are given;
- if j is scheduled on machine i, the outcome of $\boldsymbol{p}_{ij}$ becomes known when j completes.
- *Competitive analysis*: compare to optimal scheduling policy.

Online Scheduling Policies

- The *mean busy time* M_j of job j is the average of all times when it is being processed.

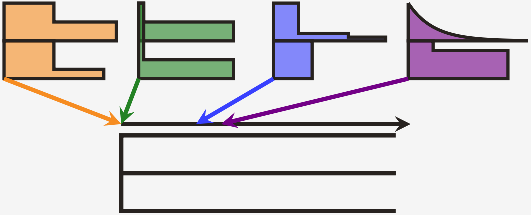


Online Scheduling Policies

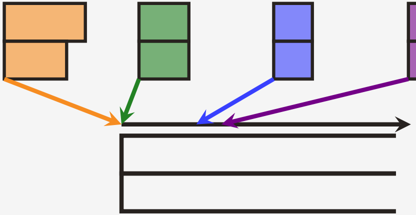
- The *mean busy time* M_j of job j is the average of all times when it is being processed.
- When job j is released, assign it to a machine i with minimum increase of

$$\sum_{j=1}^{n} w_j \cdot \left(M_j + \frac{p_{ij}}{2}\right)$$

in the virtual preemptive WSPT schedule of deterministic counterparts with $p_{ij} := \mathrm{E}[\boldsymbol{p}_{ij}]$.



Virtual schedule

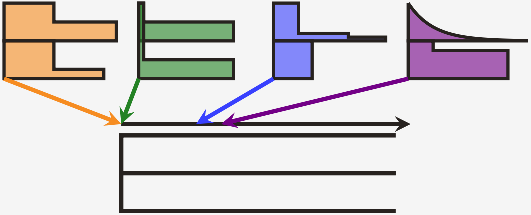


Online Scheduling Policies

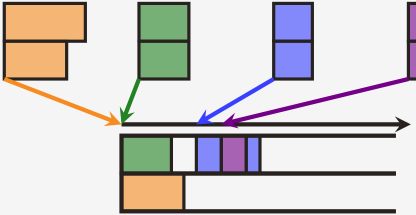
- The *mean busy time* M_j of job j is the average of all times when it is being processed.
- When job j is released, assign it to a machine i with minimum increase of

$$\sum_{j=1}^{n} w_j \cdot \left(M_j + \frac{p_{ij}}{2}\right)$$

in the virtual preemptive WSPT schedule of deterministic counterparts with $p_{ij} := \mathrm{E}[\boldsymbol{p}_{ij}]$.



Virtual schedule

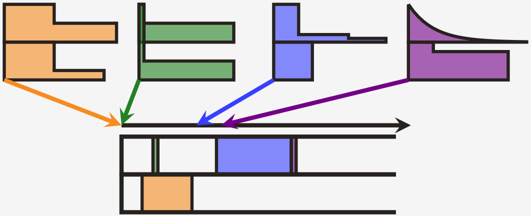


Online Scheduling Policies

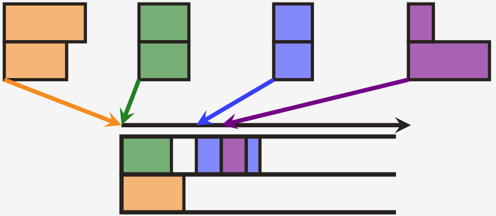
- The *mean busy time* M_j of job j is the average of all times when it is being processed.
- When job j is released, assign it to a machine i with minimum increase of

$$\sum_{j=1}^{n} w_j \cdot \left(M_j + \frac{p_{ij}}{2}\right)$$

  in the virtual preemptive WSPT schedule of deterministic counterparts with $p_{ij} := \mathrm{E}[\boldsymbol{p}_{ij}]$.
- On each machine, schedule the jobs by single machine (semi-)online policy from previous chapter.



Virtual schedule



Overview

I. Deterministic Scheduling

WSPT Rule

Scheduling with Precedence Delays

Online Non-Migratory Preemptive Scheduling on Unrelated Machines

Non-Clairvoyant Online Scheduling with Precedence Delays

Thank you!

II. Stochastic Scheduling

WSEPT Rule

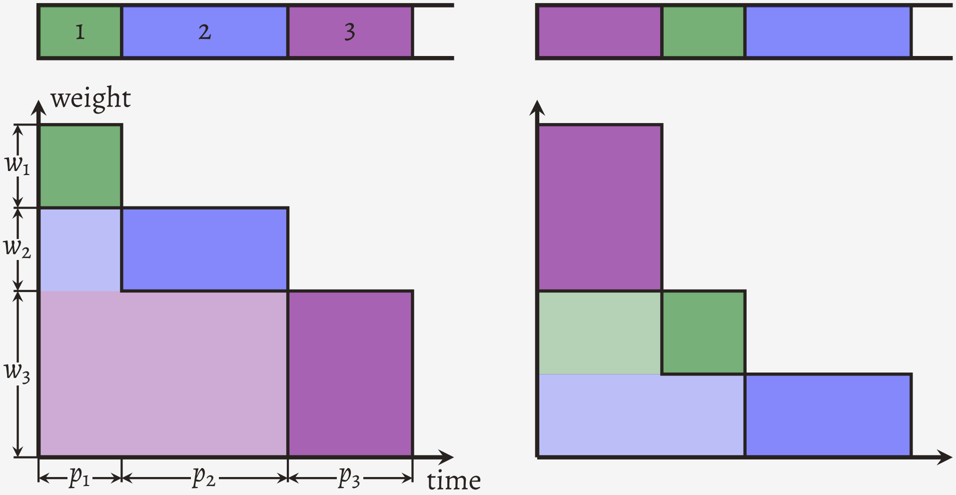
Stochastic Scheduling on Single Machine with Release Dates

Stochastic Online Scheduling on Unrelated Machines

Additional Slides

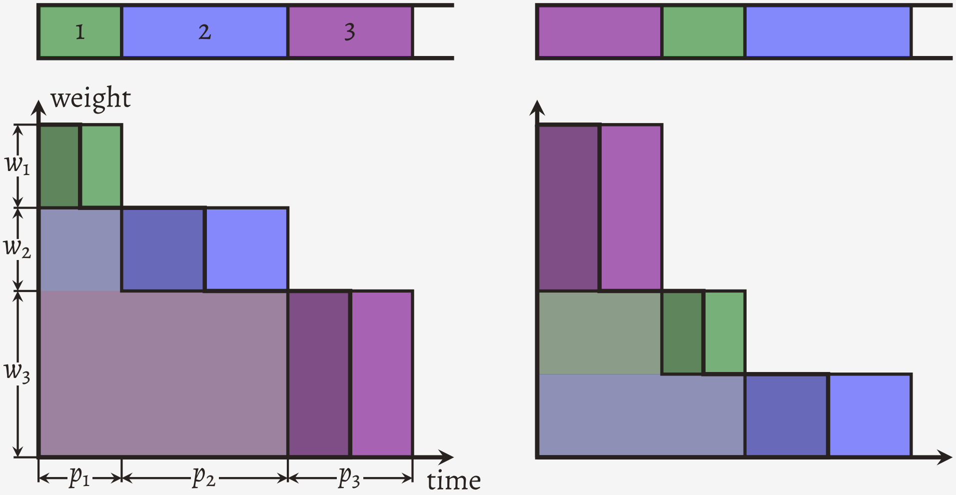
Two-Dimensional Gantt Charts

Eastman et al. 1964



Two-Dimensional Gantt Charts

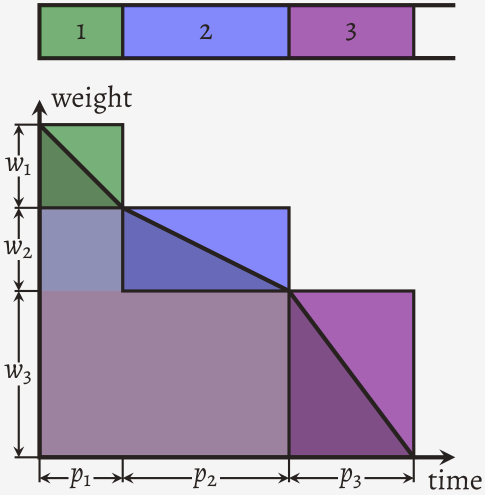
Eastman et al. 1964

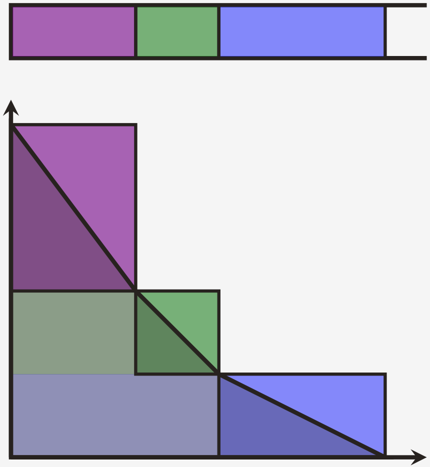


$$\sum_{j=1}^{n} w_j \cdot C_j = \sum_{j=1}^{n} w_j \cdot M_j + \frac{1}{2} \cdot \sum_{j=1}^{n} w_j \cdot p_j$$

Two-Dimensional Gantt Charts

Eastman et al. 1964



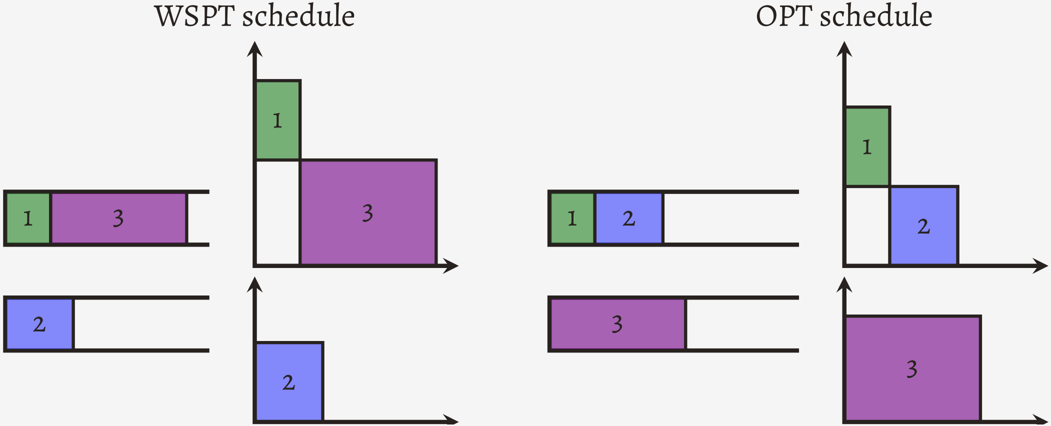


$$\sum_{j=1}^{n} w_j \cdot C_j = \sum_{j=1}^{n} w_j \cdot M_j + \frac{1}{2} \cdot \sum_{j=1}^{n} w_j \cdot p_j$$

Reduction to Unit Smith Ratio

Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

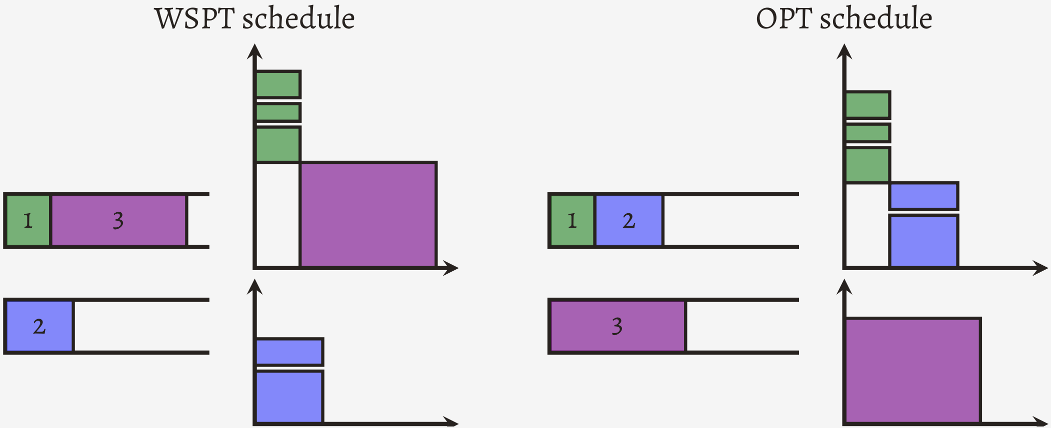
On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ for all j.



Reduction to Unit Smith Ratio

Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

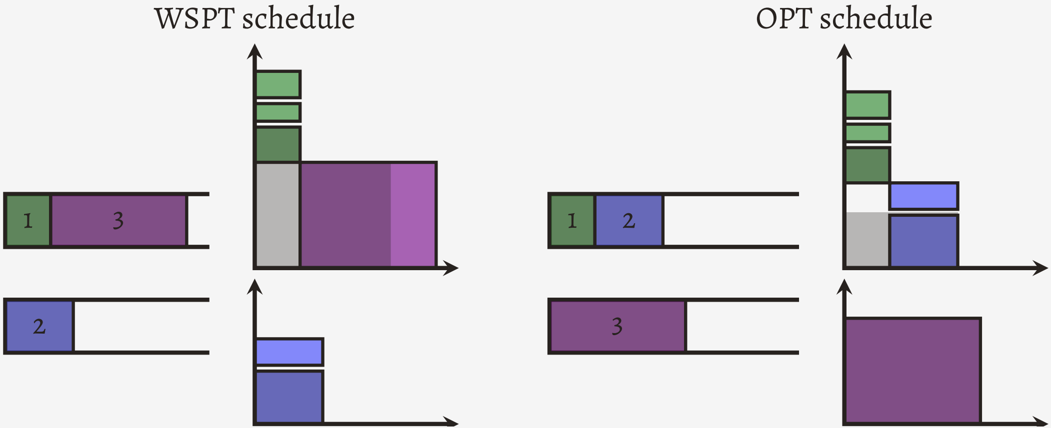
On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ *for all* j.



Reduction to Unit Smith Ratio

Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

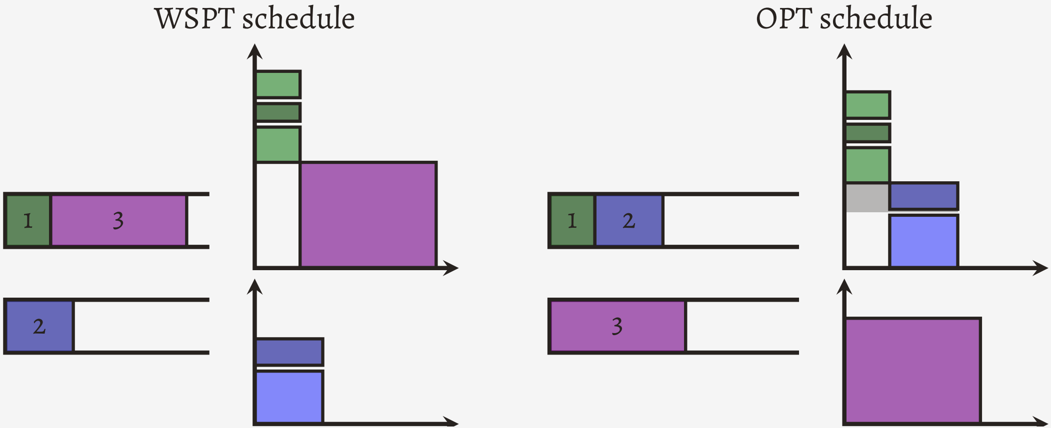
On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ for all j.



Reduction to Unit Smith Ratio

Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

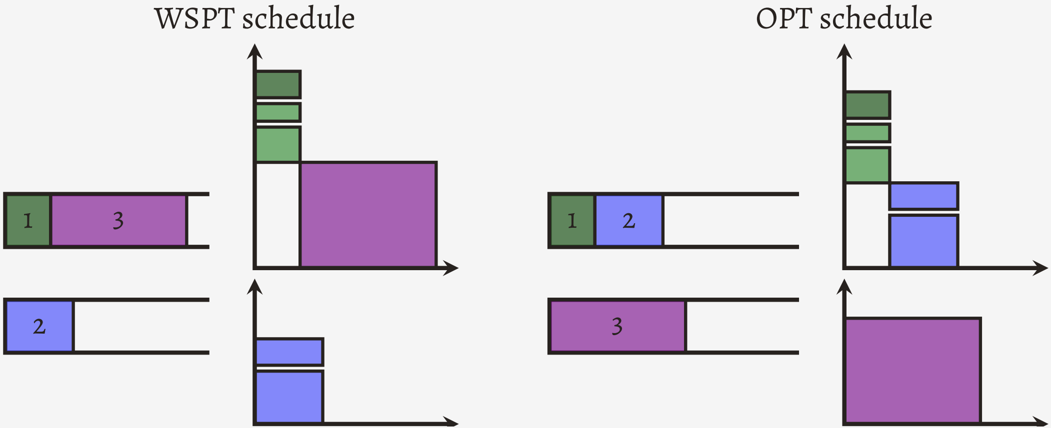
On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ for all j.



Reduction to Unit Smith Ratio

Lemma (Kawaguchi, Kyan 1986; Schwiegelshohn 2011)

On any number of machines the worst-case approximation ratio is attained when $w_j = p_j$ for all j.



Approximation of the WSPT Rule for Sum of Weighted α-Points

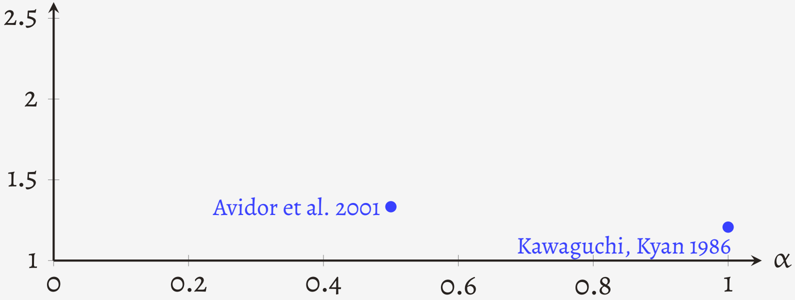
For $\alpha \in (0, 1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$

Approximation of the WSPT Rule for Sum of Weighted α-Points

For $\alpha \in (0, 1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

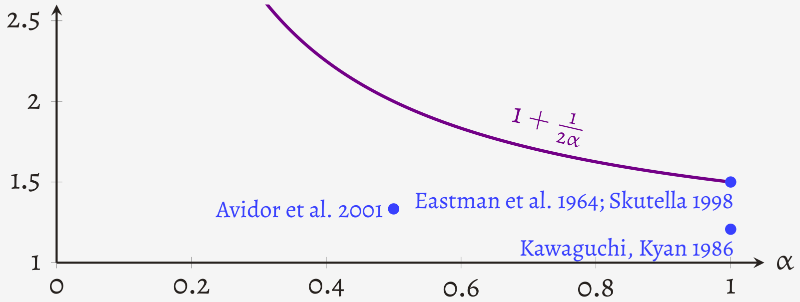
- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$



Approximation of the WSPT Rule for Sum of Weighted α-Points

For $\alpha \in (0,1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

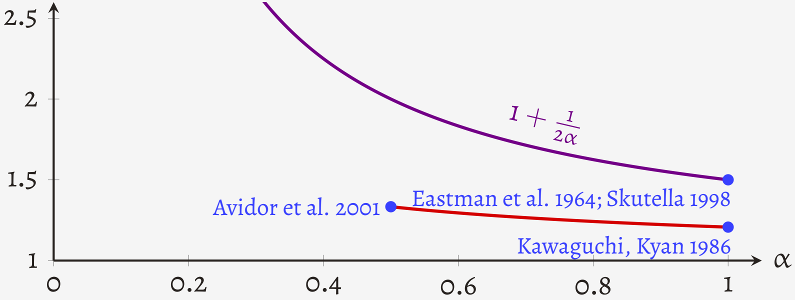
- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$



Approximation of the WSPT Rule for Sum of Weighted α-Points

For $\alpha \in (0, 1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$



Theorem

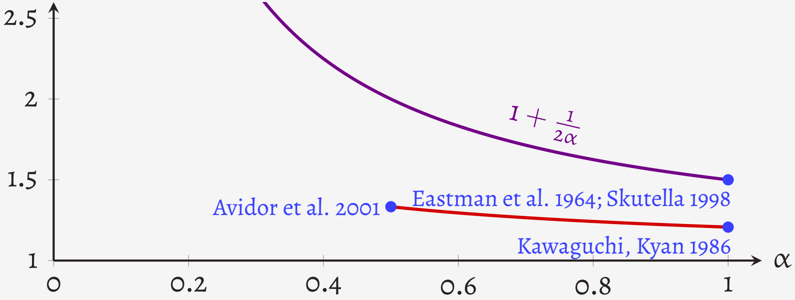
For $\alpha \in \left[\frac{1}{2}, 1\right]$ the WSPT rule has approximation ratio

$$1 + \frac{1}{2\alpha + \sqrt{8\alpha}}.$$

Approximation of the WSPT Rule for Sum of Weighted α-Points

For $\alpha \in (0, 1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$



Theorem

For $\alpha \in \left[\frac{1}{2}, 1\right]$ the WSPT rule has approximation ratio

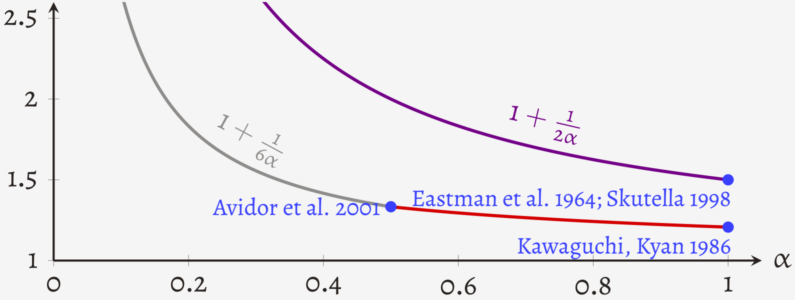
$$1 + \frac{1}{2\alpha + \sqrt{8\alpha}}.$$

- Also computed worst case for $\alpha \in \left(\frac{1}{2}, 1\right]$ on $m \geq 2$ machines.

Approximation of the WSPT Rule for Sum of Weighted α-Points

For $\alpha \in (0,1]$ the α-point $C_j(\alpha)$ of job j is the first moment when an α-fraction of j has been finished.

- $C_j(1) = C_j$
- $C_j(\frac{1}{2}) = M_j$



Theorem

For $\alpha \in [\frac{1}{2}, 1]$ the WSPT rule has approximation ratio

$$1 + \frac{1}{2\alpha + \sqrt{8\alpha}}.$$

- Also computed worst case for $\alpha \in (\frac{1}{2}, 1]$ on $m \geq 2$ machines.

Open Problem: Online List Assignment Model

- jobs arrive one by one; must be assigned to machines immediately
- on each machine, assigned jobs are optimally sequenced (WSPT)

Min-increase algorithm

Assign each job to a machine minimizing the increase of the current objective value.

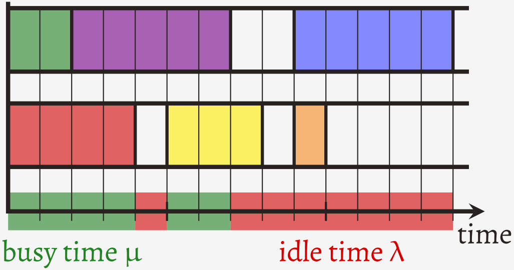
Known results:

- $\left(\frac{3}{2} - \frac{1}{2m}\right)$-competitive (cp. e.g. Megow et al. 2006)
- If jobs arrive in order of non-increasing or non-decreasing $\frac{w_j}{p_j}$, then Min-increase achieves competitive ratio $\frac{1+\sqrt{2}}{2}$.

Conjecture (Stougie 2017)

Min-increase has competitive ratio $\frac{1+\sqrt{2}}{2}$.

Scheduling with Precedence Delays on Identical Machines



Consider partial schedule when the blue job has been assigned.

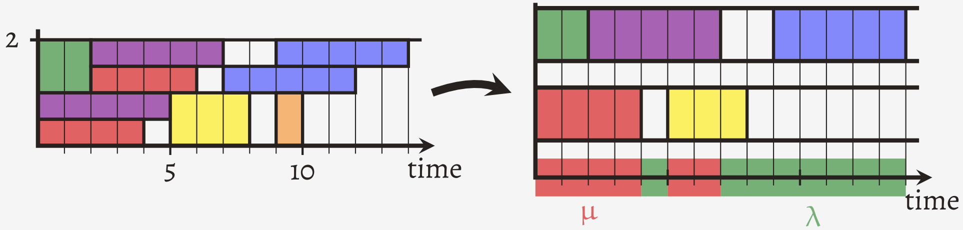
$$C_{\text{blue}} = \mu + \lambda.$$

For $\alpha \in (0, \frac{1}{2}]$ it holds that

$$\mu \leq \frac{1}{\alpha} \cdot C_{\text{blue}}^{\text{LP}}, \qquad \lambda \leq \frac{1}{1-\alpha} \cdot C_{\text{blue}}^{\text{LP}}.$$

For $\alpha = \frac{1}{2}$ this gives a 4-approximation algorithm.

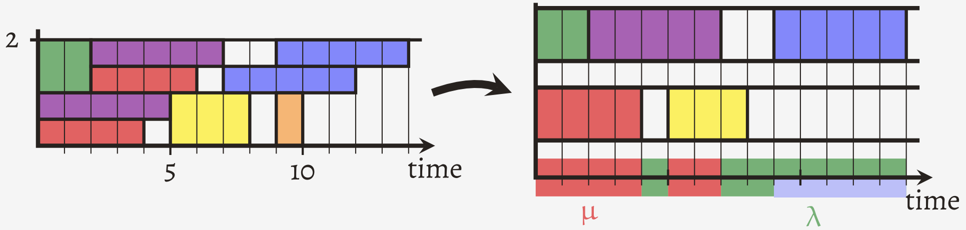
Bounding the Idle Time



Trace back why the blue job is not completed earlier.

$\lambda \quad \leq \dots$

Bounding the Idle Time

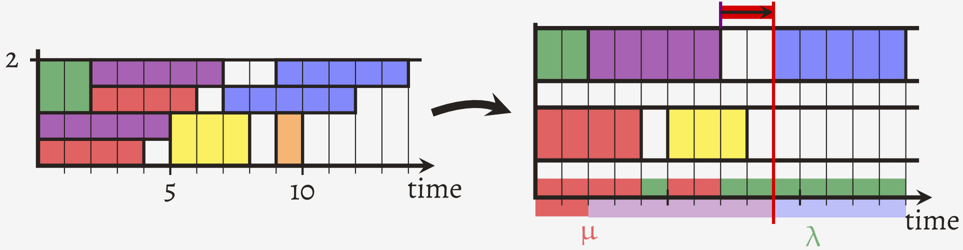


Trace back why the blue job is not completed earlier.

- The blue job has to be processed (can be idle time).

$$\lambda \leq p_{\text{blue}} + \dots$$

Bounding the Idle Time

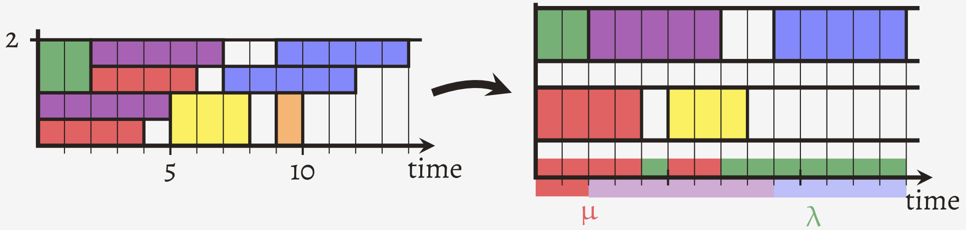


Trace back why the blue job is not completed earlier.

- The blue job has to be processed (can be idle time).
- The blue job has to wait for the purple job (can be idle time).

$$\lambda \leq p_{\text{blue}} + d_{\text{purple,blue}} + p_{\text{purple}}$$

Bounding the Idle Time

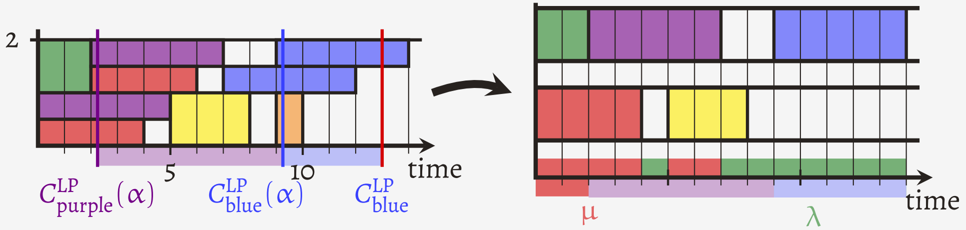


Trace back why the blue job is not completed earlier.

- The blue job has to be processed (can be idle time).
- The blue job has to wait for the purple job (can be idle time).
- The purple job has to wait because before all machines are busy (cannot be idle time).

$$\lambda \leq p_{\text{blue}} + d_{\text{purple,blue}} + p_{\text{purple}}$$

Bounding the Idle Time

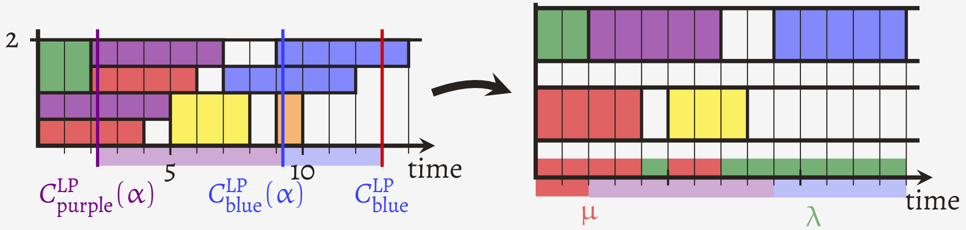


Trace back why the blue job is not completed earlier.

- The blue job has to be processed (can be idle time).
- The blue job has to wait for the purple job (can be idle time).
- The purple job has to wait because before all machines are busy (cannot be idle time).

$$\lambda \leq \underbrace{p_{blue}}_{=\frac{1}{1-\alpha}(C^{LP}_{blue} - C^{LP}_{blue}(\alpha))} + \underbrace{d_{purple,blue} + p_{purple}}_{\leq \frac{1}{1-\alpha}(C^{LP}_{blue}(\alpha) - C^{LP}_{purple}(\alpha))}$$

Bounding the Idle Time

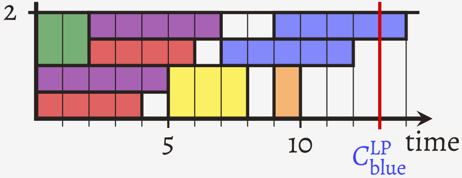


Trace back why the blue job is not completed earlier.

- The blue job has to be processed (can be idle time).
- The blue job has to wait for the purple job (can be idle time).
- The purple job has to wait because before all machines are busy (cannot be idle time).

$$\lambda \leq \underbrace{p_{blue}}_{=\frac{1}{1-\alpha}(C^{LP}_{blue}-C^{LP}_{blue}(\alpha))} + \underbrace{d_{purple,blue} + p_{purple}}_{\leq \frac{1}{1-\alpha}(C^{LP}_{blue}(\alpha)-C^{LP}_{purple}(\alpha))} \leq \frac{1}{1-\alpha} \cdot C^{LP}_{blue}.$$

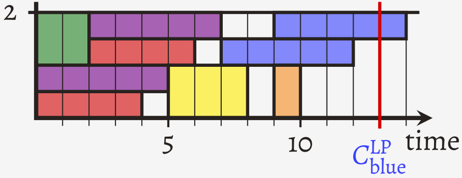
Bounding the Expected Busy Time



For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{ j \in N \mid C_j^{LP}(\alpha) \le C^{LP}_{blue} \right\}$.

Upper bound on busy time: $\mu \le \dfrac{p(J_\alpha)}{m}$.

Bounding the Expected Busy Time

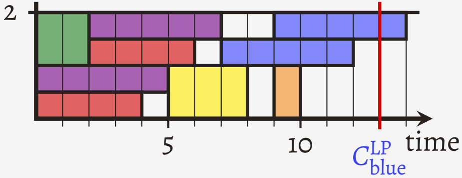


For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{ j \in N \mid C^{LP}_j(\alpha) \le C^{LP}_{blue} \right\}$.

$$\text{Upper bound on busy time: } \mu \le \frac{p(J_\alpha)}{m}.$$

$$\text{Lower bound on LP-completion time: } C^{LP}_{blue} \ge \frac{E\left[p(J_\alpha)\right]}{2m}.$$

Bounding the Expected Busy Time



For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{j \in N \mid C_j^{\text{LP}}(\alpha) \leq C^{\text{LP}}_{\text{blue}}\right\}$.

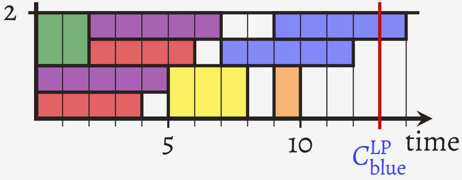
$$\text{Upper bound on busy time: } \mu \leq \frac{p(J_\alpha)}{m}.$$

$$\text{Lower bound on LP-completion time: } C^{\text{LP}}_{\text{blue}} \geq \frac{E\left[p(J_\alpha)\right]}{2m}.$$

$$E[\mu] \leq 2 \cdot C^{\text{LP}}_{\text{blue}}.$$

Bounding the Expected Busy Time

Lower bound on LP-completion time

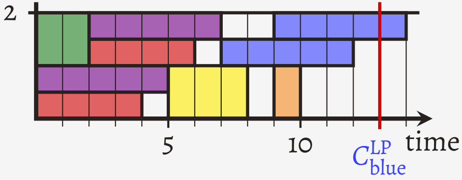


For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{ j \in N \mid C_j^{LP}(\alpha) \leq C^{LP}_{blue} \right\}$.

Claim: $C^{LP}_{blue} \geq \frac{1}{2} \cdot \frac{E\left[p(J_\alpha)\right]}{m}$.

Bounding the Expected Busy Time

Lower bound on LP-completion time



For $\alpha \in (0, \frac{1}{2}]$ let $J_\alpha := \{j \in N \mid C_j^{LP}(\alpha) \le C^{LP}_{blue}\}$.

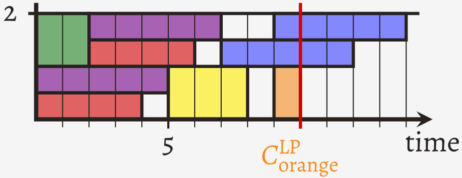
Claim: $C^{LP}_{blue} \ge \frac{1}{2} \cdot \frac{E[p(J_\alpha)]}{m}$.

In the example $C_j^{LP}(\frac{1}{2}) \le C^{LP}_{blue}$ for all $j \in N$, whence $J_\alpha = N$ for all α. Therefore,

$$C^{LP}_{blue} \ge \frac{1}{2} \cdot \frac{p(N)}{m} = \frac{1}{2} \cdot \frac{E[p(J_\alpha)]}{m}.$$

Bounding the Expected Busy Time

Lower bound on LP-completion time

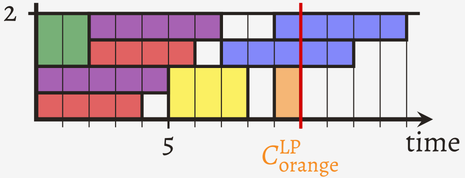


For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{ j \in N \mid C_j^{\text{LP}}(\alpha) \le C^{\text{LP}}_{\text{orange}} \right\}$.

Claim: $C^{\text{LP}}_{\text{orange}} \ge \frac{1}{2} \cdot \frac{E\left[p(J_\alpha)\right]}{m}$.

Bounding the Expected Busy Time

Lower bound on LP-completion time



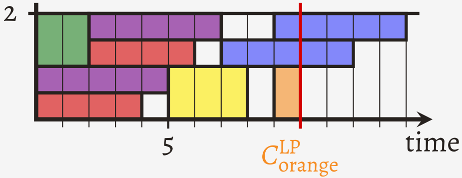
For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha \coloneqq \left\{ j \in N \mid C_j^{LP}(\alpha) \le C^{LP}_{\text{orange}} \right\}$.

Claim: $C^{LP}_{\text{orange}} \ge \frac{1}{2} \cdot \frac{E\left[p(J_\alpha)\right]}{m}$.

$$p(J_\alpha) = \begin{cases} p_{\text{green}} + p_{\text{purple}} + p_{\text{red}} + p_{\text{yellow}} + p_{\text{orange}} + p_{\text{blue}} & \text{if } \alpha \le \frac{2}{5}; \\ p_{\text{green}} + p_{\text{purple}} + p_{\text{red}} + p_{\text{yellow}} + p_{\text{orange}} & \text{if } \alpha > \frac{2}{5}. \end{cases}$$

Bounding the Expected Busy Time

Lower bound on LP-completion time



For $\alpha \in (0, \frac{1}{2}]$ let $J_\alpha := \{j \in N \mid C_j^{LP}(\alpha) \leq C^{LP}_{orange}\}$.

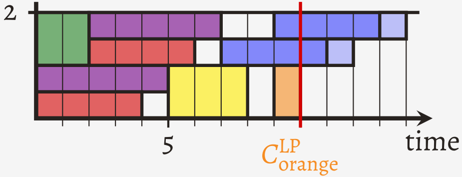
Claim: $C^{LP}_{orange} \geq \frac{1}{2} \cdot \frac{E[p(J_\alpha)]}{m}$.

$$p(J_\alpha) = \begin{cases} p_{green} + p_{purple} + p_{red} + p_{yellow} + p_{orange} + p_{blue} & \text{if } \alpha \leq \frac{2}{5}; \\ p_{green} + p_{purple} + p_{red} + p_{yellow} + p_{orange} & \text{if } \alpha > \frac{2}{5}. \end{cases}$$

$$E[p(J_\alpha)] = p_{green} + p_{purple} + p_{red} + p_{yellow} + p_{orange} + \frac{4}{5} \cdot p_{blue}$$

Bounding the Expected Busy Time

Lower bound on LP-completion time



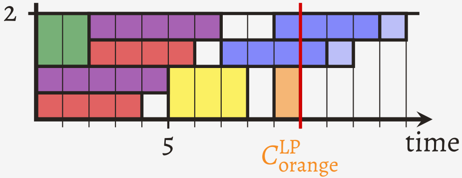
For $\alpha \in (0, \frac{1}{2}]$ let $J_\alpha := \{j \in N \mid C_j^{LP}(\alpha) \leq C^{LP}_{orange}\}$.

Claim: $C^{LP}_{orange} \geq \frac{1}{2} \cdot \frac{E[p(J_\alpha)]}{m}$.

$$E[p(J_\alpha)] = p_{green} + p_{purple} + p_{red} + p_{yellow} + p_{orange} + \frac{4}{5} \cdot p_{blue}$$

Bounding the Expected Busy Time

Lower bound on LP-completion time



For $\alpha \in \left(0, \frac{1}{2}\right]$ let $J_\alpha := \left\{ j \in N \mid C_j^{\text{LP}}(\alpha) \le C^{\text{LP}}_{\text{orange}} \right\}$.

Claim: $C^{\text{LP}}_{\text{orange}} \ge \frac{1}{2} \cdot \frac{E\left[p(J_\alpha)\right]}{m}$.

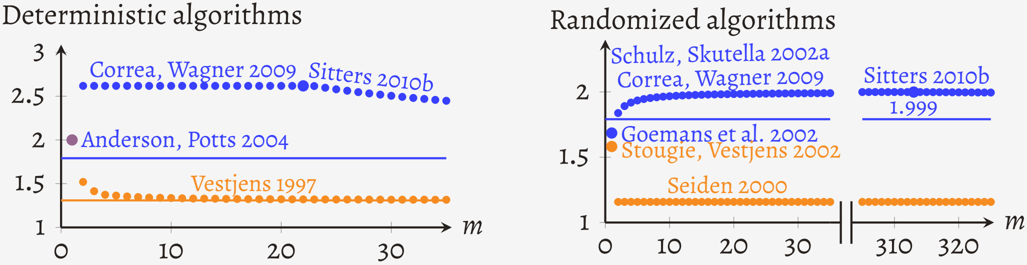
$$E[p(J_\alpha)] = p_{\text{green}} + p_{\text{purple}} + p_{\text{red}} + p_{\text{yellow}} + p_{\text{orange}} + \frac{4}{5} \cdot p_{\text{blue}}$$

The modified jobs J' have midpoints bounded by $C^{\text{LP}}_{\text{orange}}$. Therefore,

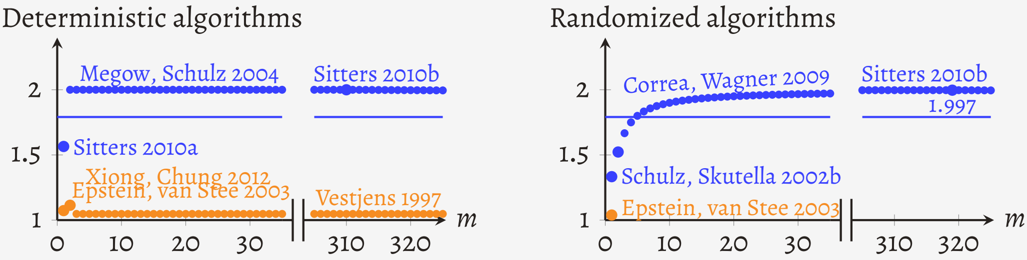
$$C^{\text{LP}}_{\text{orange}} \ge \frac{1}{2} \cdot \frac{p(J')}{m} = \frac{1}{2} \cdot \frac{E\left[p(J_\alpha)\right]}{m}.$$

Clairvoyant Online Time Model on Identical Machines

Non-preemptive scheduling



Preemptive scheduling



Non-Migratory Preemptive Scheduling on Unrelated Machines

Assume that all $p_{ij}, r_j \in 2\mathbb{Z}$, and let $T \in \mathbb{Z}$ be an upper bound on the makespan.

Variables: y_{ijt}, $i \in [m]$, $j \in [n]$, $t \in \{r_j, \dots, T-1\}$, indicating how long job j is processed on machine i during time slot $(t, t+1]$.

$$
\begin{aligned}
\text{(LP)}\ \min \quad & \sum_{j=1}^{n} w_j \cdot \sum_{i=1}^{m} \sum_{t=r_j}^{T-1} \left(\frac{y_{ijt}}{2} + \frac{y_{ijt}}{p_{ij}} \cdot \left(t + \frac{1}{2}\right) \right) \\
\text{s. t.} \quad & \sum_{i=1}^{m} \sum_{t=r_j}^{T-1} \frac{y_{ijt}}{p_{ij}} = 1 && \forall j \in [n] \\
& \sum_{j: r_j \geq t} y_{ijt} \leq 1 && \forall i \in [m],\ t \in \{0, \dots, T-1\} \\
& y_{ijt} \geq 0 && \forall i \in [m],\ j \in [n],\ t \in \{r_j, \dots, T-1\}
\end{aligned}
$$

Dual LP

$$
\begin{aligned}
\text{(LP)}\ \min\ & \sum_{j=1}^{n} w_j \cdot \sum_{i=1}^{m} \sum_{t=r_j}^{T-1} \left(\frac{y_{ijt}}{2} + \frac{y_{ijt}}{p_{ij}} \cdot \left(t + \frac{1}{2}\right) \right) \\
\text{s.t.}\ & \sum_{i=1}^{m} \sum_{t=r_j}^{T-1} \frac{y_{ijt}}{p_{ij}} = 1 && \forall j \in [n] \\
& \sum_{j: r_j \geq t} y_{ijt} \leq 1 && \forall i \in [m],\ t \in \{0, \ldots, T-1\} \\
& y_{ijt} \geq 0 && \forall i \in [m],\ j \in [n],\ t \in \{r_j, \ldots, T-1\}
\end{aligned}
$$

$$
\begin{aligned}
\text{(D)}\ \max\ & \sum_{j=1}^{n} \chi_j - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \psi_{it} \\
\text{s.t.}\ & \frac{\chi_j}{p_{ij}} \leq \psi_{it} + w_j \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2} \right) && \forall i \in [m],\ j \in [n],\ t \in \{r_j, \ldots, T-1\} \\
& \psi_{it} \geq 0 && \forall i \in [m],\ t \in \{0, \ldots, T-1\}
\end{aligned}
$$

Dual Solution

- For $j \in [n]$ let $i(j)$ be the machine to which j is assigned by the Greedy-Assignment WSRPT algorithm.
- For $i \in [m]$ and $t \in \mathbb{Z}_{\geq 0}$ let $U_{i,t}$ be the set of jobs completed after time t on machine i in the Greedy-Assignment WSRPT schedule.

$$\chi_j := \frac{1}{2} \cdot \text{cost}\big(j \to i(j)\big) \qquad \text{for all } j \in [n],$$

$$\psi_{it} := \frac{w(U_{i,2t})}{2} \qquad \text{for all } i \in [m],\ t \in \{0, \ldots, T-1\},$$

Cost of Dual Solution

$$
\begin{aligned}
\text{(D)}\ \max \quad & \sum_{j=1}^{n} \chi_j - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \psi_{it} \\
\text{s. t.} \quad & \frac{\chi_j}{p_{ij}} \leq \psi_{it} + w_j \cdot \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2} \right) && \forall i, j, t \\
& \psi_{it} \geq 0 && \forall i, t
\end{aligned}
$$

Cost of Dual Solution

$$(D)\ \max \quad \sum_{j=1}^{n} \frac{\text{cost}(j \to i(j))}{2} - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \frac{w(U_{i,2t})}{2}$$

$$\text{s. t.} \quad \frac{\chi_j}{p_{ij}} \leq \psi_{it} + w_j \cdot \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2} \right) \qquad \forall i, j, t$$

$$\psi_{it} \geq 0 \qquad \forall i, t$$

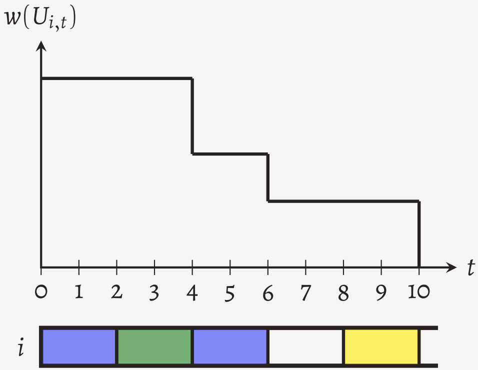
Cost of Dual Solution

$$\sum_{j=1}^{n} \frac{\text{cost}\big(j \to i(j)\big)}{2} - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \frac{w(U_{i,2t})}{2}$$

Cost of Dual Solution

$$\sum_{j=1}^{n} \frac{\text{cost}(j \to i(j))}{2} - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \frac{w(U_{i,2t})}{2}$$

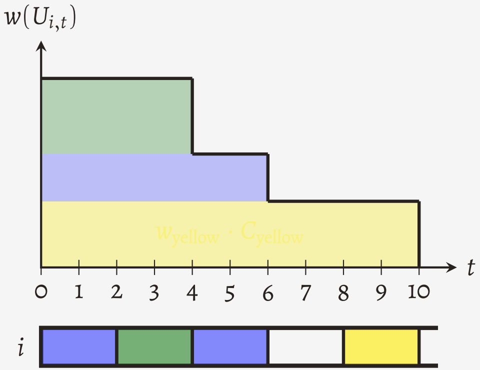
$$= \frac{1}{2} \cdot z^{\text{GA-WSRPT}} - \frac{1}{4} \cdot z^{\text{GA-WSRPT}} = \frac{1}{4} \cdot z^{\text{GA-WSRPT}}$$



Cost of Dual Solution

$$\sum_{j=1}^{n} \frac{\mathrm{cost}\big(j \to i(j)\big)}{2} - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \frac{w(U_{i,2t})}{2}$$

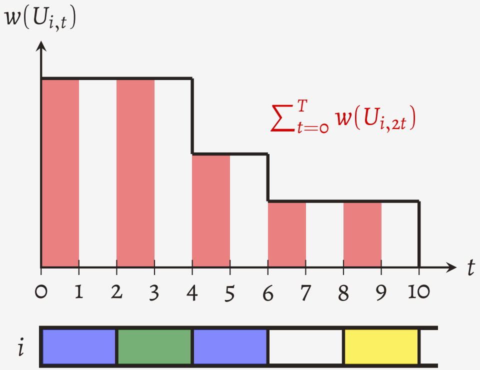
$$= \frac{1}{2} \cdot z^{\text{GA-WSRPT}} - \frac{1}{4} \cdot z^{\text{GA-WSRPT}} = \frac{1}{4} \cdot z^{\text{GA-WSRPT}}$$



Cost of Dual Solution

$$\sum_{j=1}^{n} \frac{\text{cost}\big(j \to i(j)\big)}{2} - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \frac{w(U_{i,2t})}{2}$$

$$= \frac{1}{2} \cdot z^{\text{GA-WSRPT}} - \frac{1}{4} \cdot z^{\text{GA-WSRPT}} = \frac{1}{4} \cdot z^{\text{GA-WSRPT}}$$



Feasibility of Dual Solution

$$\text{(D)} \quad \max \quad \sum_{j=1}^{n} \chi_j - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \psi_{it}$$

$$\text{s.t.} \quad \frac{\chi_j}{p_{ij}} \leq \psi_{it} + w_j \cdot \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2} \right) \quad \forall i, j, t$$

$$\psi_{it} \geq 0 \quad \forall i, t$$

Feasibility of Dual Solution

$$
\begin{aligned}
\text{(D)}\ \max\quad & \sum_{j=1}^{n} \chi_j - \sum_{i=1}^{m} \sum_{t=0}^{T-1} \psi_{it} \\
\text{s. t.}\quad & \frac{\mathrm{cost}(j \to i(j))}{2 \cdot p_{ij}} \le \frac{w(U_{i,2t})}{2} + w_j \cdot \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2}\right) && \forall i, j, t \\
& \frac{w(U_{i,2t})}{2} \ge 0 && \forall i, t
\end{aligned}
$$

Feasibility of Dual Solution

$$\frac{\text{cost}(j \rightarrow i(j))}{2 \cdot p_{ij}} \leq \frac{w(U_{i,2t})}{2} + w_j \cdot \left(\frac{t + 1/2}{p_{ij}} + \frac{1}{2} \right) \quad \forall i, j, t$$

Weighted Shortest Expected Processing Time First Rule

WSEPT rule

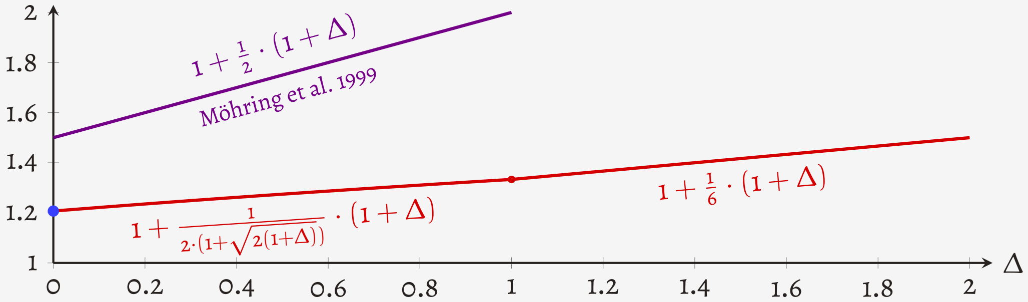
Whenever a machine is free, start available job with maximum ratio $w_j / E[\boldsymbol{p}_j]$ on it.

- WSEPT is optimal if
  - there is only one machine (Rothkopf 1966)
  - all jobs have unit weight, and the processing times are pairwise stochastically comparable (Weber et al. 1986)
- Even for unit-weight jobs WSEPT has no constant performance guarantee. (Cheung et al. 2014; Im et al. 2015)
- The approximation ratio can be bounded in terms of

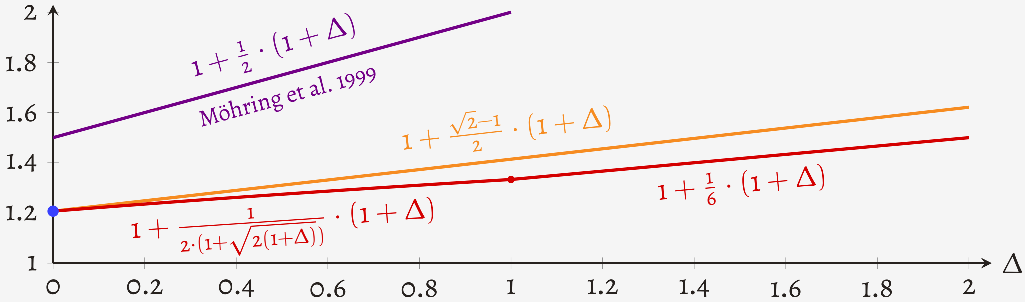
$$\Delta := \max_{j \in \{1,\dots,n\}} \frac{\mathrm{Var}[\boldsymbol{p}_j]}{E[\boldsymbol{p}_j]^2}.$$

(Möhring et al. 1999)

Performance Guarantees for the WSEPT Rule



Performance Guarantees for the WSEPT Rule



Proof of the Performance Guarantee

Idea: Consider random weights $\boldsymbol{w}_j' := \frac{\boldsymbol{p}_j}{E[\boldsymbol{p}_j]} \cdot w_j$.

Proof of the Performance Guarantee

Idea: Consider random weights $\mathbf{w}_j' := \frac{\mathbf{p}_j}{E[\mathbf{p}_j]} \cdot w_j$.

$\implies$ in every realization the WSEPT rule schedules the jobs in non-increasing order of $\frac{\mathbf{w}_j'}{\mathbf{p}_j}$.

$$\sum_{j=1}^{n} \mathbf{w}_j' \cdot \mathbf{C}_j^{\mathrm{WSEPT}} \leq \frac{1+\sqrt{2}}{2} \cdot \sum_{j=1}^{n} \mathbf{w}_j' \cdot \mathbf{C}_j^{\mathrm{OPT}(\mathbf{p},\mathbf{w}')} \leq \frac{1+\sqrt{2}}{2} \cdot \sum_{j=1}^{n} \mathbf{w}_j' \cdot \mathbf{C}_j^{\mathrm{OPT}}.$$

Proof of the Performance Guarantee

Idea: Consider random weights $\boldsymbol{w}_j' := \frac{\boldsymbol{p}_j}{E[\boldsymbol{p}_j]} \cdot w_j$.

$\Longrightarrow$ in every realization the WSEPT rule schedules the jobs in non-increasing order of $\frac{\boldsymbol{w}_j'}{\boldsymbol{p}_j}$.

$$E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{WSEPT}}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}(\boldsymbol{p},\boldsymbol{w}')}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}}\right].$$

Proof of the Performance Guarantee

Idea: Consider random weights $\boldsymbol{w}_j' := \frac{\boldsymbol{p}_j}{E[\boldsymbol{p}_j]} \cdot w_j$.

$\Longrightarrow$ in every realization the WSEPT rule schedules the jobs in non-increasing order of $\frac{\boldsymbol{w}_j'}{\boldsymbol{p}_j}$.

$$E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\mathrm{WSEPT}}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\mathrm{OPT}(\boldsymbol{p},\boldsymbol{w}')}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\mathrm{OPT}}\right].$$

Lemma

For any stochastic scheduling policy Π it holds that

$$E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\Pi}\right] = E\left[\sum_{j=1}^{n} w_j \cdot \boldsymbol{C}_j^{\Pi}\right] + \underbrace{\sum_{j=1}^{n} w_j \cdot \frac{\mathrm{Var}[\boldsymbol{p}_j]}{E[\boldsymbol{p}_j]}}_{=:c}.$$

Proof of the Performance Guarantee

Idea: Consider random weights $\boldsymbol{w}_j' := \frac{\boldsymbol{p}_j}{E[\boldsymbol{p}_j]} \cdot w_j$.

$\Longrightarrow$ in every realization the WSEPT rule schedules the jobs in non-increasing order of $\frac{\boldsymbol{w}_j'}{\boldsymbol{p}_j}$.

$$E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{WSEPT}}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}(\boldsymbol{p},\boldsymbol{w}')}\right] \leq \frac{1+\sqrt{2}}{2} \cdot E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}}\right].$$

Lemma

For any stochastic scheduling policy Π it holds that

$$E\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\Pi}\right] = E\left[\sum_{j=1}^{n} w_j \cdot \boldsymbol{C}_j^{\Pi}\right] + \underbrace{\sum_{j=1}^{n} w_j \cdot \frac{\text{Var}[\boldsymbol{p}_j]}{E[\boldsymbol{p}_j]}}_{=:c}.$$

Proof

$$E[\boldsymbol{p}_j \boldsymbol{C}_j^{\Pi}] = E[\boldsymbol{p}_j \boldsymbol{S}_j] + E[\boldsymbol{p}_j^2] = E[\boldsymbol{p}_j] \cdot \big(E[\boldsymbol{S}_j] + E[\boldsymbol{p}_j]\big) + \text{Var}[\boldsymbol{p}_j] = E[\boldsymbol{p}_j] \cdot E[\boldsymbol{C}_j] + \text{Var}[\boldsymbol{p}_j]. \quad \square$$

Proof of the Performance Guarantee

Idea: Consider random weights $\boldsymbol{w}_j' := \frac{\boldsymbol{p}_j}{\mathrm{E}[\boldsymbol{p}_j]} \cdot w_j$.

$\Longrightarrow$ in every realization the WSEPT rule schedules the jobs in non-increasing order of $\frac{\boldsymbol{w}_j'}{\boldsymbol{p}_j}$.

$$\mathrm{E}\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{WSEPT}}\right] \leq \frac{1+\sqrt{2}}{2} \cdot \mathrm{E}\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}(\boldsymbol{p},\boldsymbol{w}')}\right] \leq \frac{1+\sqrt{2}}{2} \cdot \mathrm{E}\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\text{OPT}}\right].$$

Lemma

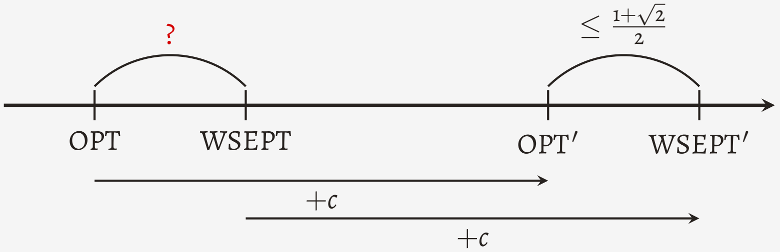
For any stochastic scheduling policy Π it holds that

$$\mathrm{E}\left[\sum_{j=1}^{n} \boldsymbol{w}_j' \cdot \boldsymbol{C}_j^{\Pi}\right] = \mathrm{E}\left[\sum_{j=1}^{n} w_j \cdot \boldsymbol{C}_j^{\Pi}\right] + \underbrace{\sum_{j=1}^{n} w_j \cdot \frac{\mathrm{Var}[\boldsymbol{p}_j]}{\mathrm{E}[\boldsymbol{p}_j]}}_{=:c \leq \sum_j w_j \cdot \Delta \cdot \mathrm{E}[\boldsymbol{p}_j] \leq \Delta \mathrm{OPT}}.$$

Proof

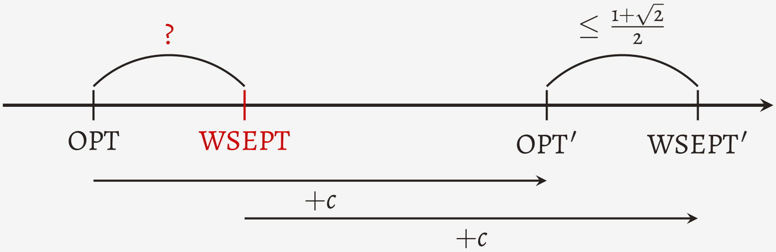
$$\mathrm{E}[\boldsymbol{p}_j \boldsymbol{C}_j^{\Pi}] = \mathrm{E}[\boldsymbol{p}_j \boldsymbol{S}_j] + \mathrm{E}[\boldsymbol{p}_j^2] = \mathrm{E}[\boldsymbol{p}_j] \cdot \big(\mathrm{E}[\boldsymbol{S}_j] + \mathrm{E}[\boldsymbol{p}_j]\big) + \mathrm{Var}[\boldsymbol{p}_j] = \mathrm{E}[\boldsymbol{p}_j] \cdot \mathrm{E}[\boldsymbol{C}_j] + \mathrm{Var}[\boldsymbol{p}_j]. \quad \square$$

Proof of the Performance Guarantee (Cont.)



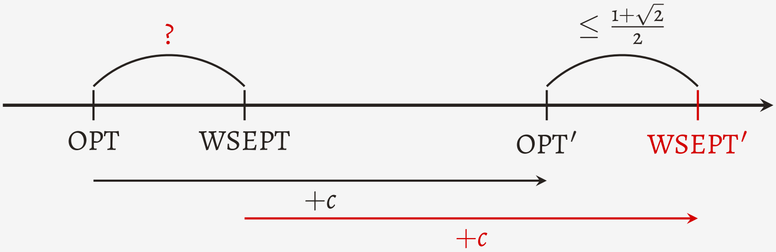
WSEPT

Proof of the Performance Guarantee (Cont.)



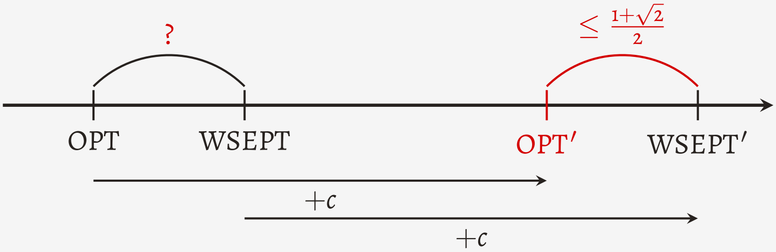
WSEPT

Proof of the Performance Guarantee (Cont.)



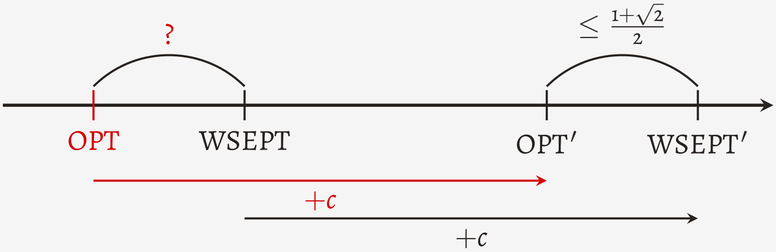
$$\text{WSEPT} = \text{WSEPT}' - c$$

Proof of the Performance Guarantee (Cont.)



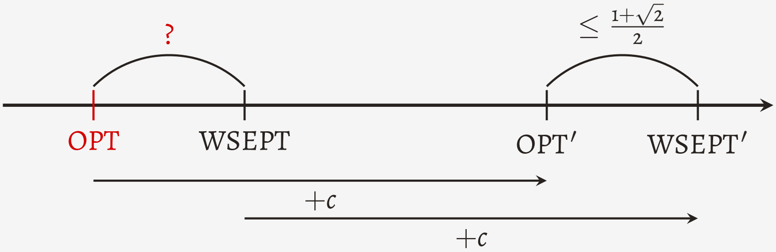
$$\mathrm{WSEPT} = \mathrm{WSEPT}' - c \leq \frac{1+\sqrt{2}}{2} \cdot \mathrm{OPT}' - c$$

Proof of the Performance Guarantee (Cont.)



$$\text{WSEPT} = \text{WSEPT}' - c \leq \frac{1+\sqrt{2}}{2} \cdot \text{OPT}' - c = \frac{1+\sqrt{2}}{2} \cdot (\text{OPT} + c) - c$$

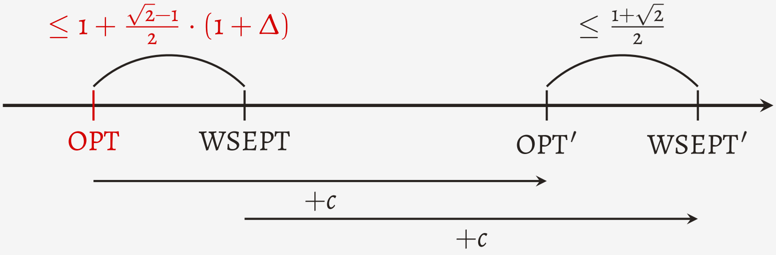
Proof of the Performance Guarantee (Cont.)



$$\text{WSEPT} = \text{WSEPT}' - c \leq \frac{1+\sqrt{2}}{2} \cdot \text{OPT}' - c = \frac{1+\sqrt{2}}{2} \cdot (\text{OPT} + c) - c$$

$$= \text{OPT} + \frac{\sqrt{2}-1}{2} \cdot (\text{OPT} + c)$$

Proof of the Performance Guarantee (Cont.)

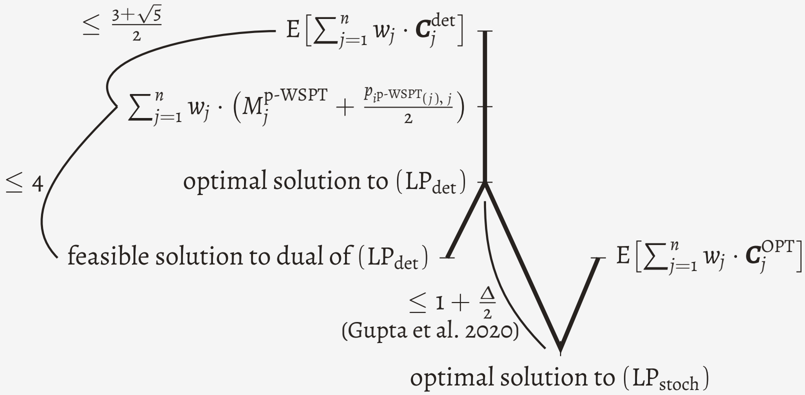


$$\text{WSEPT} = \text{WSEPT}' - c \leq \frac{1+\sqrt{2}}{2} \cdot \text{OPT}' - c = \frac{1+\sqrt{2}}{2} \cdot (\text{OPT} + c) - c$$

$$= \text{OPT} + \frac{\sqrt{2}-1}{2} \cdot (\text{OPT} + c) \leq \left(1 + \frac{\sqrt{2}-1}{2} \cdot (1+\Delta)\right) \cdot \text{OPT}$$

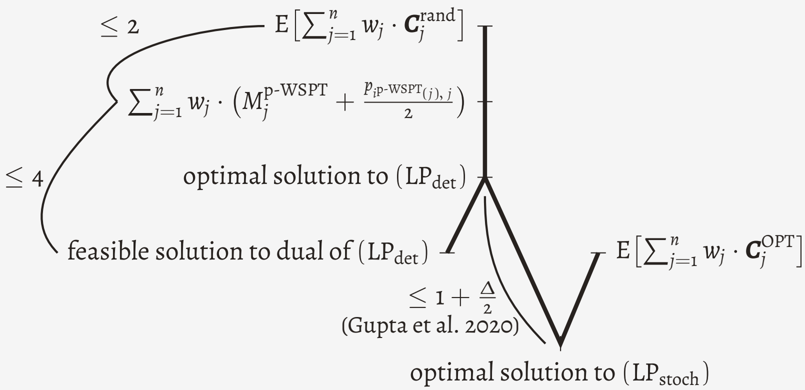
$c \leq \Delta\text{OPT}$

Analysis of Online Scheduling Policies for Unrelated Machines



deterministic online $\quad \frac{3+\sqrt{5}}{2} \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = (3+\sqrt{5}) \cdot (2+\Delta)$

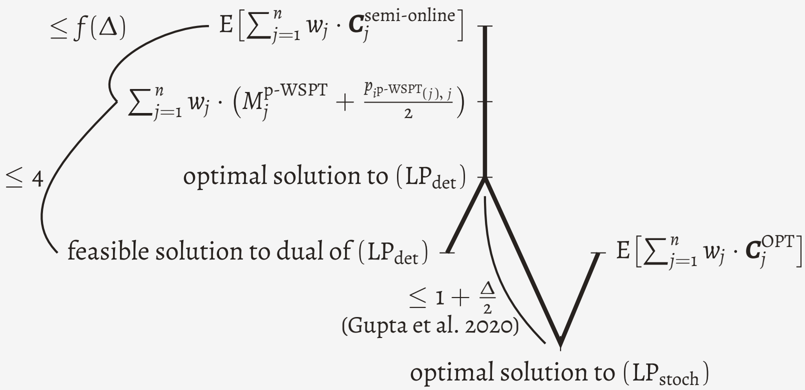
Analysis of Online Scheduling Policies for Unrelated Machines



deterministic online $\quad \frac{3+\sqrt{5}}{2} \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = (3 + \sqrt{5}) \cdot (2 + \Delta)$

randomized online $\quad 2 \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = 4 \cdot (2 + \Delta)$

Analysis of Online Scheduling Policies for Unrelated Machines

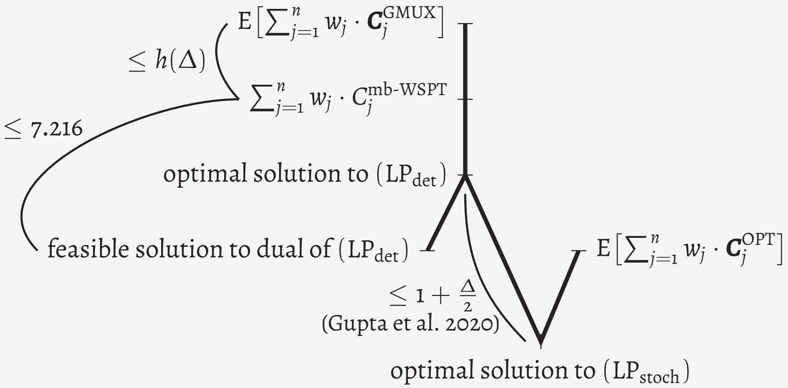


deterministic online $\quad \frac{3+\sqrt{5}}{2} \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = (3+\sqrt{5}) \cdot (2+\Delta)$

randomized online $\quad 2 \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = 4 \cdot (2+\Delta)$

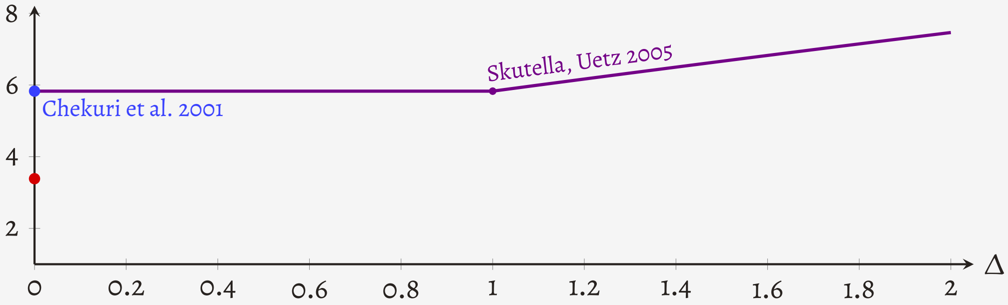
semi-online $\quad f(\Delta) \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = f(\Delta) \cdot 2 \cdot (2+\Delta)$

Analysis of Online Scheduling Policies for Unrelated Machines



deterministic online	$\frac{3+\sqrt{5}}{2} \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = (3+\sqrt{5}) \cdot (2+\Delta)$
randomized online	$2 \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = 4 \cdot (2+\Delta)$
semi-online	$f(\Delta) \cdot 4 \cdot \left(1 + \frac{\Delta}{2}\right) = f(\Delta) \cdot 2 \cdot (2+\Delta)$
Gupta et al. 2021+	$h(\Delta) \cdot 7.216 \cdot \left(1 + \frac{\Delta}{2}\right) = 3.608 \cdot h(\Delta) \cdot (2+\Delta)$

Stochastic Scheduling with Precedence Constraints and Release Dates



Anand, S., Garg, N., Kumar, A. (2012). "Resource Augmentation for Weighted Flow-Time Explained by Dual Fitting". *Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pp. 1228–1241.

Anderson, E. J., Potts, C. N. (2004). "Online Scheduling of a Single Machine to Minimize Total Weighted Completion Time". *Mathematics of Operations Research* 29(3), pp. 686–697.

Avidor, A., Azar, Y., Sgall, J. (2001). "Ancient and New Algorithms for Load Balancing in the ℓ_p Norm". *Algorithmica* 29(3), pp. 422–441.

Bansal, N., Pruhs, K. (2004). "Server Scheduling in the Weighted ℓ_p Norm". *LATIN 2004: Theoretical Informatics*, pp. 434–443.

Bruno, J., Coffman Jr., E. G., Sethi, R. (1974). "Scheduling Independent Tasks to Reduce Mean Finishing Time". *Communications of the ACM* 17(7), pp. 382–387.

Chakrabarti, S. et al. (1996). "Improved Scheduling Algorithms for Minsum Criteria". *Automata, Languages and Programming: 23rd International Colloquium* (ICALP), pp. 646–657.

Chekuri, C., Motwani, R., Natarajan, B., Stein, C. (2001). "Approximation Techniques for Average Completion Time Scheduling". *SIAM Journal on Computing* 31(1), pp. 146–166.

Cheung, W. C., Fischer, F., Matuschke, J., Megow, N. (2014). "An $\Omega(\Delta^{1/2})$ Gap example on the (W)SEPT Policy". unpublished note. URL: http://www.maths.qmul.ac.uk/~ffischer/publications/cfmm_wsept.pdf.

Correa, J. R., Wagner, M. R. (2009). “LP-based online scheduling: from single to parallel machines”. *Mathematical Programming* 119(1), pp. 109–136.

Eastman, W. L., Even, S., Isaacs, I. M. (1964). “Bounds for the Optimal Scheduling of n Jobs on m Processors”. *Management Science* 11(2), pp. 268–279.

Epstein, L., Stee, R. van (2003). “Lower bounds for on-line single-machine scheduling”. *Theoretical Computer Science* 299(1), pp. 439–450.

Garg, N., Gupta, A., Kumar, A., Singla, S. (2019). “Non-Clairvoyant Precedence Constrained Scheduling”. *46th International Colloquium on Automata, Languages, and Programming* (ICALP), no. 63.

Goemans, M. X., Queyranne, M., Schulz, A. S., Skutella, M., Wang, Y. (2002). “Single Machine Scheduling with Release Dates”. *SIAM Journal on Discrete Mathematics* 15(2), pp. 165–192.

Gupta, V., Moseley, B., Uetz, M., Xie, Q. (2020). “Greed Works—Online Algorithms for Unrelated Machine Stochastic Scheduling”. *Mathematics of Operations Research* 45(2), 2.

— (2021+). “Corrigendum: Greed Works – Online Algorithms For Unrelated Machine Stochastic Scheduling”. *Mathematics of Operations Research*. To appear.

Hall, L. A., Schulz, A. S., Shmoys, D. B., Wein, J. (1997). "Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms". *Mathematics of Operations Research* 22(3), pp. 513–544.

Im, S., Li, S. (2016). "Better Unrelated Machine Scheduling for Weighted Completion Time via Random Offsets from Non-Uniform Distributions". *57th Annual IEEE Symposium on Foundations of Computer Science* (FOCS), pp. 138–147.

Im, S., Moseley, B., Pruhs, K. (2015). "Stochastic Scheduling of Heavy-tailed Jobs". *32nd Symposium on Theoretical Aspects of Computer Science* (STACS), pp. 474–486.

Kawaguchi, T., Kyan, S. (1986). "Worst Case Bound of an LRF Schedule for the Mean Weighted Flow-Time Problem". *SIAM Journal on Computing* 15(4), pp. 1119–1129.

Lageweg, B. J., Lenstra, J. K. (1977). Cited as private communication in M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

Li, S. (2020). "Scheduling to Minimize Total Weighted Completion Time via Time-Indexed Linear Programming Relaxations". *SIAM Journal on Computing* 49(4), FOCS17-409–FOCS17-440.

Megow, N., Schulz, A. S. (2004). "On-line scheduling to minimize average completion time revisited". *Operations Research Letters* 32(5), pp. 485–490.

- Megow, N., Uetz, M., Vredeveld, T. (2006). "Models and Algorithms for Stochastic Online Scheduling". *Mathematics of Operations Research* 31(3), pp. 513–525.
- Möhring, R. H., Radermacher, F. J. (1985). "Introduction to Stochastic Scheduling Problems". *Contributions to Operations Research: Proceedings of the Conference on Operations Research Held in Oberwolfach*, pp. 72–130.
- Möhring, R. H., Schulz, A. S., Uetz, M. (1999). "Approximation in Stochastic Scheduling: The Power of LP-based Priority Policies". *Journal of the ACM* 46(6), pp. 924–942.
- Rothkopf, M. H. (1966). "Scheduling with Random Service Times". *Management Science* 12(9), pp. 707–713.
- Sahni, S. K. (1976). "Algorithms for Scheduling Independent Tasks". *Journal of the ACM* 23(1), pp. 116–127.
- Schulz, A. S. (2008). "Stochastic Online Scheduling Revisited". *Combinatorial Optimization and Applications: Second International Conference* (COCOA), pp. 448–457.
- Schulz, A. S., Skutella, M. (2002a). "Scheduling Unrelated Machines by Randomized Rounding". *SIAM Journal on Discrete Mathematics* 15(4), pp. 450–469.
- — (2002b). "The Power of α-Points in Preemptive Machine Scheduling". *Journal of Scheduling* 5(2), pp. 121–133.

Schwiegelshohn, U. (2011). “An alternative proof of the Kawaguchi–Kyan bound for the Largest-Ratio-First rule”. *Operations Research Letters* 39(4), pp. 255–259.

Seiden, S. S. (2000). “A Guessing Game and Randomized Online Algorithms”. *Thirty-Second Annual ACM Symposium on Theory of Computing* (STOC), pp. 592–601.

Sitters, R. (2010a). “Competitive analysis of preemptive single-machine scheduling”. *Operations Research Letters* 38(6), pp. 585–588.

— (2010b). “Efficient Algorithms for Average Completion Time Scheduling”. *Integer Programming and Combinatorial Optimization: 14th International Conference* (IPCO), pp. 411–423.

Sitters, R., Yang, L. (2018). “A $(2+\epsilon)$-approximation for precedence constrained single machine scheduling with release dates and total weighted completion time objective”. *Operations Research Letters* 46(4), pp. 438–442.

Skutella, M. (1998). “Approximation and Randomization in Scheduling”. PhD thesis. Technische Universität Berlin.

— (2016). “A 2.542-approximation for precedence constrained single machine scheduling with release dates and total weighted completion time objective”. *Operations Research Letters* 44(5), pp. 676–679.

Skutella, M., Sviridenko, M., Uetz, M. (2016). "Unrelated Machine Scheduling with Stochastic Processing Times". *Mathematics of Operations Research* 41(3), pp. 851–864.

Skutella, M., Uetz, M. (2005). "Stochastic Machine Scheduling with Precedence Constraints". *SIAM Journal on Computing* 34(4), pp. 788–802.

Skutella, M., Woeginger, G. J. (2000). "A PTAS for Minimizing the Total Weighted Completion Time on Identical Parallel Machines". *Mathematics of Operations Research* 25(1), pp. 63–75.

Stougie, L., Vestjens, A. P. A. (2002). "Randomized algorithms for on-line scheduling problems: how low can't you go?" *Operations Research Letters* 30(2), pp. 89–96.

Vestjens, A. P. A. (1997). "On-line machine scheduling". PhD thesis. Technische Universiteit Eindhoven.

Weber, R. R., Varaiya, P., Walrand, J. (1986). "Scheduling Jobs with Stochastically Ordered Processing Times on Parallel Machines to Minimize Expected Flowtime". *Journal of Applied Probability* 23(3), pp. 841–847.

Xiong, B., Chung, C. (2012). "Completion Time Scheduling and the WSRPT Algorithm: Revised Selected Papers". *Combinatorial Optimization: Second International Symposium (ISCO)*, pp. 416–426.