Technische Universität Berlin

Scheduling Stochastic Jobs with Release Dates on a Single Machine

Sven Jäger

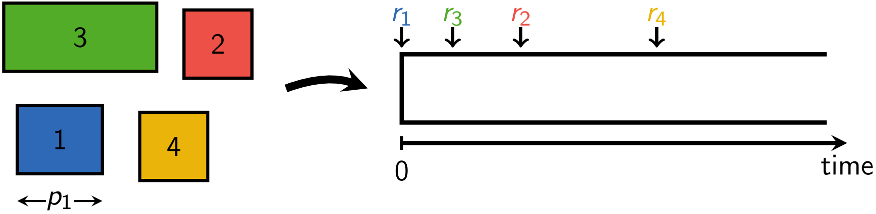
Combinatorial Optimization and Graph Algorithms

Dagstuhl Seminar Scheduling
February 18, 2020

Deterministic Problem $1|r_j|\sum w_j C_j$

Given: processing times p_j, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

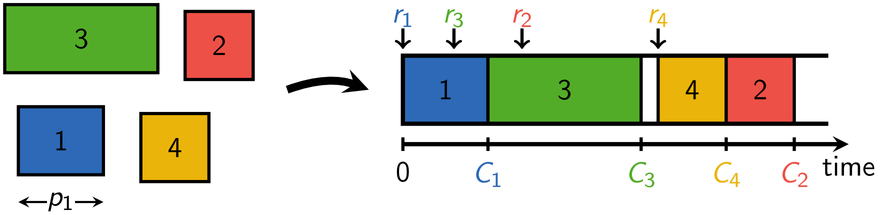
Task: schedule jobs non-preemptively so that $\sum_{j=1}^{n} w_j C_j$ is minimal.



Deterministic Problem $1|r_j|\sum w_j C_j$

Given: processing times p_j, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

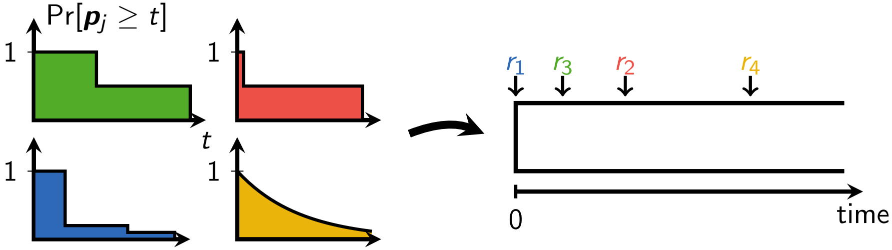
Task: schedule jobs non-preemptively so that $\sum_{j=1}^{n} w_j C_j$ is minimal.



Stochastic Problem $1|r_j|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

Given: distributions of random processing times $\boldsymbol{p}_j$, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

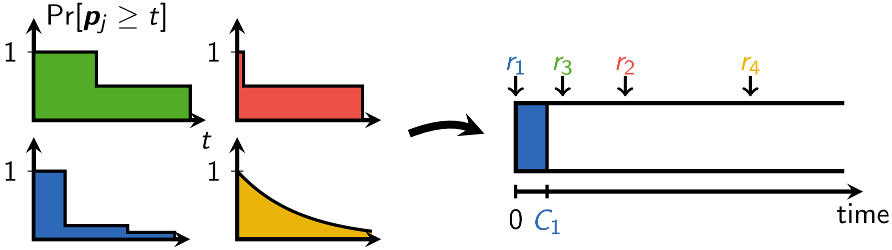
Task: find non-preemptive scheduling policy minimizing $\mathrm{E}\left[\sum_{j=1}^{n} w_j \boldsymbol{C}_j\right]$.



Stochastic Problem $1|r_j|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

Given: distributions of random processing times $\boldsymbol{p}_j$, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

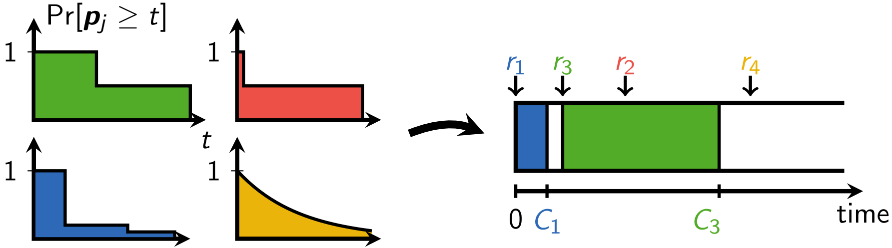
Task: find non-preemptive scheduling policy minimizing $\mathrm{E}\left[\sum_{j=1}^{n} w_j \boldsymbol{C}_j\right]$.



Stochastic Problem $1|r_j|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

Given: distributions of random processing times $\boldsymbol{p}_j$, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

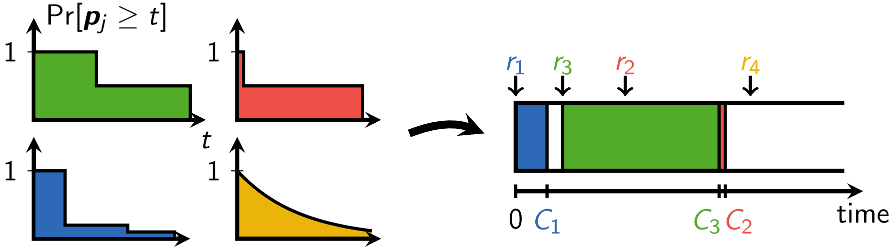
Task: find non-preemptive scheduling policy minimizing $\mathrm{E}\left[\sum_{j=1}^{n} w_j \boldsymbol{C}_j\right]$.



Stochastic Problem $1|r_j|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

Given: distributions of random processing times $\boldsymbol{p}_j$, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

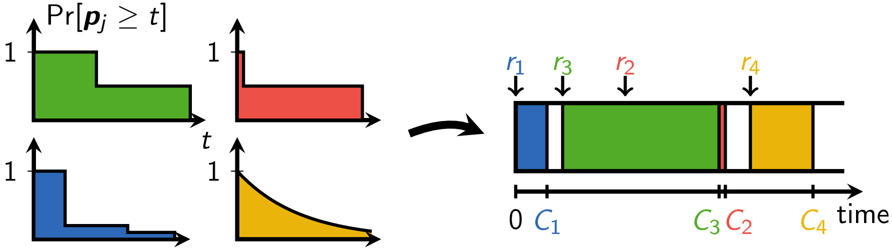
Task: find non-preemptive scheduling policy minimizing $\mathrm{E}\left[\sum_{j=1}^{n} w_j \boldsymbol{C}_j\right]$.



Stochastic Problem $1|r_j|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

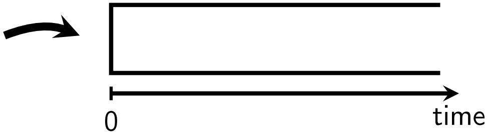
Given: distributions of random processing times $\boldsymbol{p}_j$, release dates r_j, and weights w_j of jobs $j = 1, \ldots, n$,

Task: find non-preemptive scheduling policy minimizing $\mathrm{E}\left[\sum_{j=1}^{n} w_j \boldsymbol{C}_j\right]$.



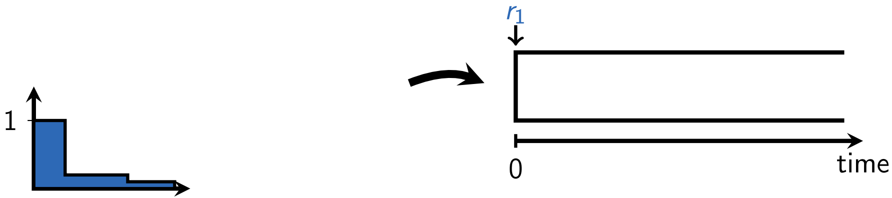
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j C_j\right]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



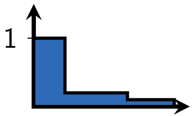
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j C_j\right]$

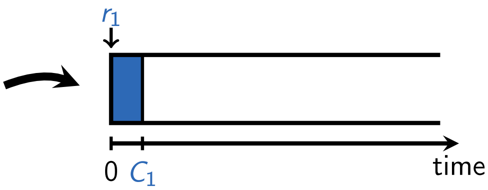
- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j C_j\right]$

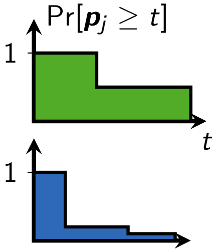
- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.

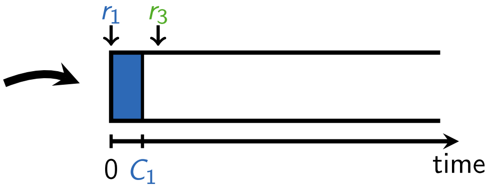




Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

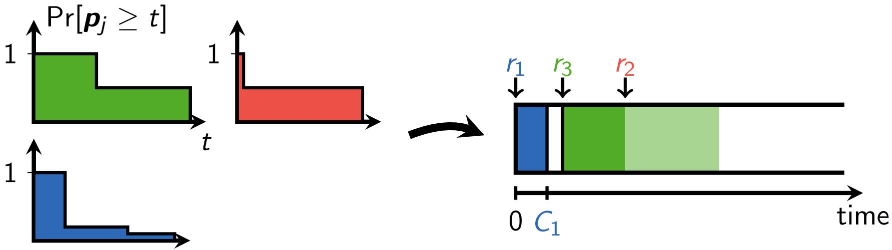
- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.





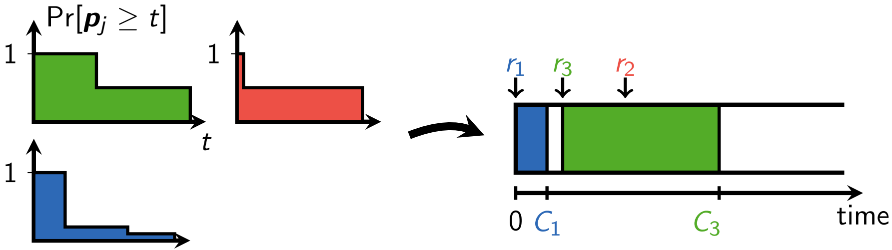
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j C_j\right]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



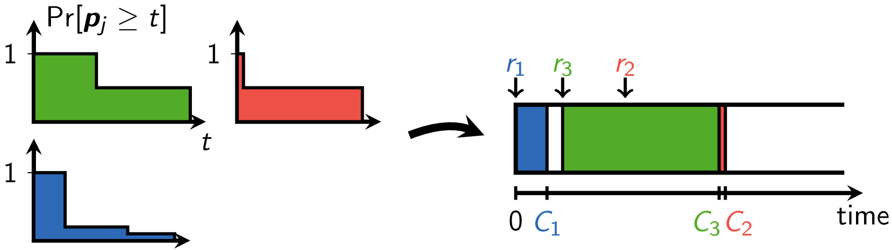
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



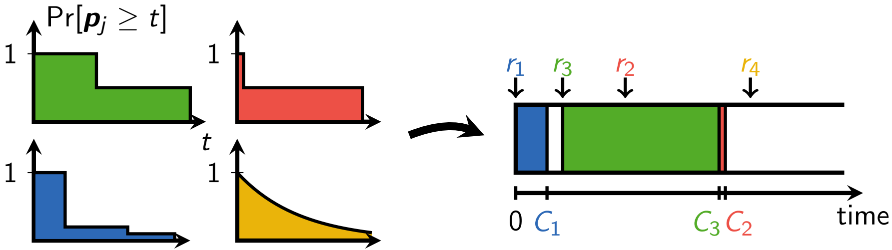
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}[\sum w_j C_j]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



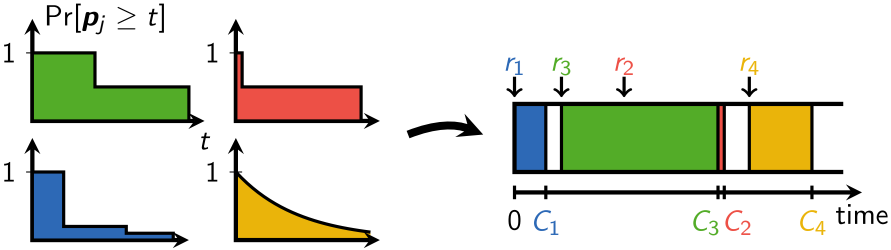
Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



Stochastic Online Problem $1|r_j \text{ online}|\mathrm{E}\left[\sum w_j \boldsymbol{C}_j\right]$

- When a job is released, its weight and its processing time distribution become known.
- Actual processing time of a job becomes known when it completes.



Performance of Randomized Stochastic Online Policy $\mathbf{\Pi}$

$$\sup_{\mathbb{P}_{\boldsymbol{p}}, r, w} \frac{\mathrm{E}_{\boldsymbol{p},\mathbf{\Pi}}\left[\sum_{j=1}^{n} w_j \cdot \boldsymbol{C}_j^{\mathbf{\Pi}}\right]}{\mathrm{E}_{\boldsymbol{p}}\left[\sum_{j=1}^{n} w_j \cdot \boldsymbol{C}_j^{\Pi^*}\right]}$$

Adversarial policy Π^*

- knows all processing time distributions, release dates, and weights at the beginning,
- learns the actual processing time of a job when it completes.

$\Longrightarrow$ Π^* is optimal solution to stochastic offline problem.

Schulz' Randomized Stochastic Online Policy (2008)

- Virtually construct preemptive WSPT schedule for deterministic counterparts with $p_j := \mathrm{E}[\boldsymbol{p}_j]$.
- The α-point of a job is the first point in time at which an α-fraction of its deterministic counterpart has been completed in the virtual schedule.
- Draw $\boldsymbol{\alpha}_j \in (0, 1]$ for all jobs j independently.

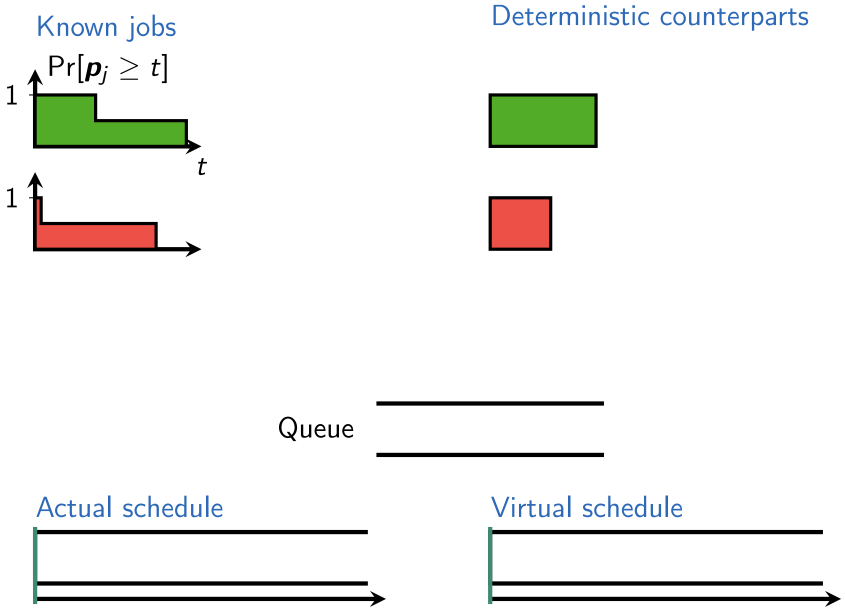
Randomized Stochastic Online Scheduling (RSOS)

Whenever $\boldsymbol{\alpha}_j$-point of a job j is reached, append j to a queue.

Whenever the actual machine is idle and the queue is non-empty, start first job from queue.

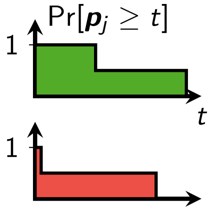
Schulz: Draw $\boldsymbol{\alpha}_j$ according to uniform distribution.

Example with unit weights



Example with unit weights

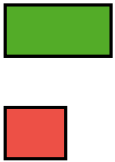
Known jobs



$\alpha_1 = 0.33$

$\alpha_2 = 0.63$

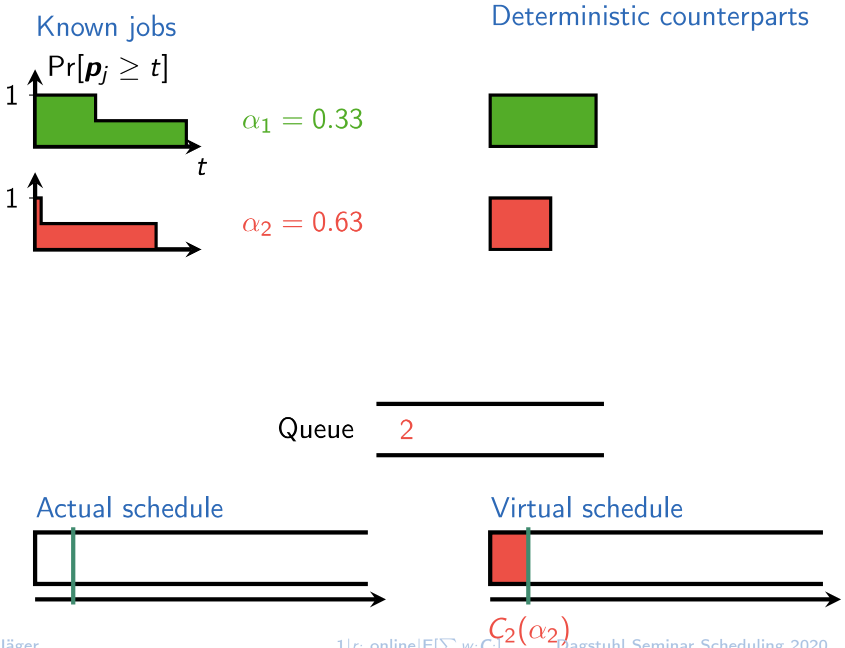
Deterministic counterparts



Actual schedule

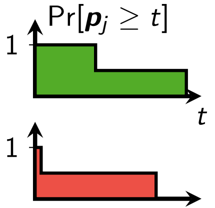
Virtual schedule

Example with unit weights



Example with unit weights

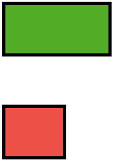
Known jobs



$\alpha_1 = 0.33$

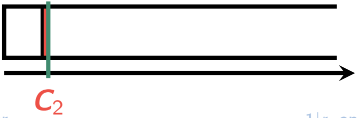
$\alpha_2 = 0.63$

Deterministic counterparts



Queue

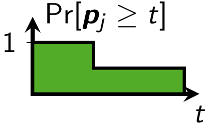
Actual schedule



Virtual schedule

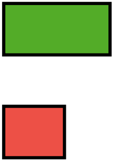
Example with unit weights

Known jobs



$\alpha_1 = 0.33$

Deterministic counterparts



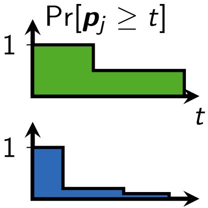
Queue

Actual schedule

Virtual schedule

Example with unit weights

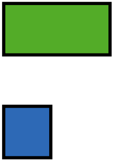
Known jobs



$\alpha_1 = 0.33$

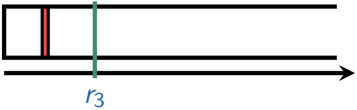
$\alpha_3 = 0.52$

Deterministic counterparts

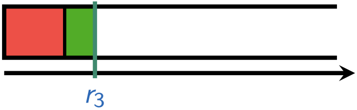


Queue

Actual schedule

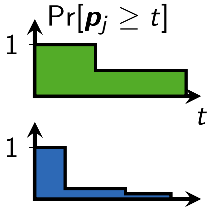


Virtual schedule



Example with unit weights

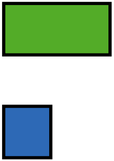
Known jobs



$\alpha_1 = 0.33$

$\alpha_3 = 0.52$

Deterministic counterparts

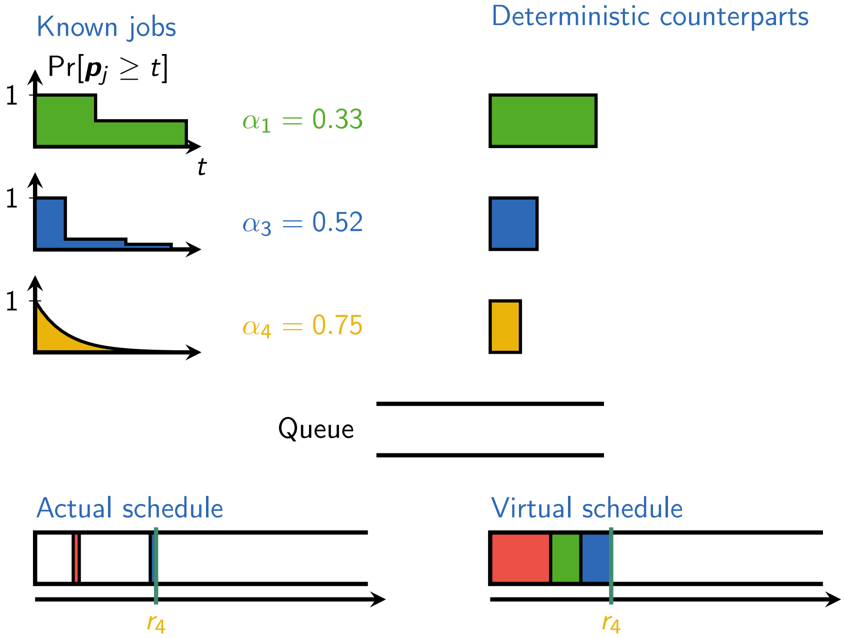


Queue

Actual schedule

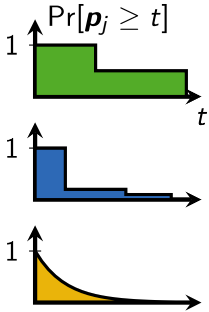
Virtual schedule

Example with unit weights



Example with unit weights

Known jobs

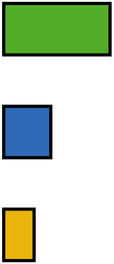


$\alpha_1 = 0.33$

$\alpha_3 = 0.52$

$\alpha_4 = 0.75$

Deterministic counterparts



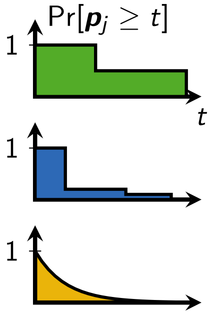
Queue 4

Actual schedule

Virtual schedule

Example with unit weights

Known jobs

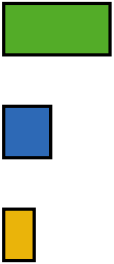


$\alpha_1 = 0.33$

$\alpha_3 = 0.52$

$\alpha_4 = 0.75$

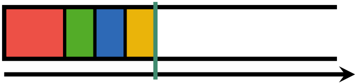
Deterministic counterparts



Queue 4

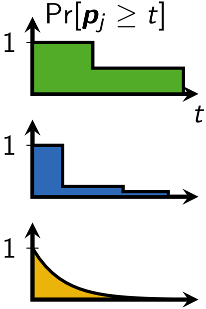
Actual schedule

Virtual schedule



Example with unit weights

Known jobs

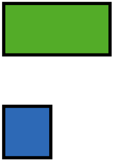


$\alpha_1 = 0.33$

$\alpha_3 = 0.52$

$\alpha_4 = 0.75$

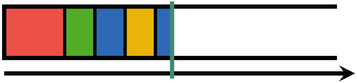
Deterministic counterparts



Queue

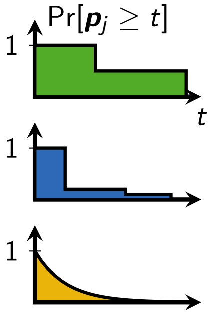
Actual schedule

Virtual schedule



Example with unit weights

Known jobs



$\alpha_1 = 0.33$

$\alpha_3 = 0.52$

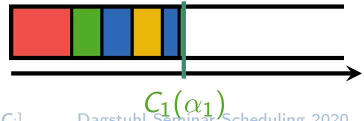
$\alpha_4 = 0.75$

Deterministic counterparts

Queue 4 1

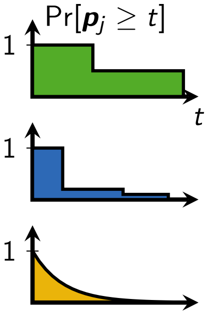
Actual schedule

Virtual schedule



Example with unit weights

Known jobs



$\alpha_1 = 0.33$

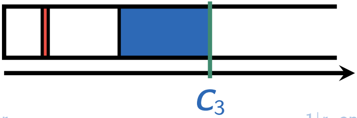
$\alpha_3 = 0.52$

$\alpha_4 = 0.75$

Deterministic counterparts

Queue 4 1

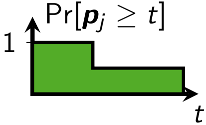
Actual schedule



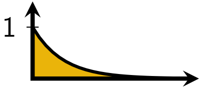
Virtual schedule

Example with unit weights

Known jobs



$\alpha_1 = 0.33$

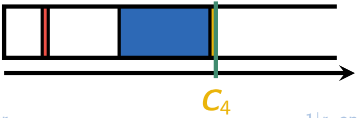


$\alpha_4 = 0.75$

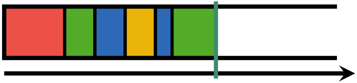
Deterministic counterparts

Queue 1

Actual schedule

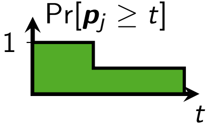


Virtual schedule



Example with unit weights

Known jobs



$\alpha_1 = 0.33$

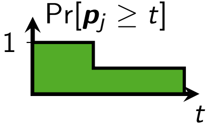
Deterministic counterparts

Actual schedule

Virtual schedule

Example with unit weights

Known jobs

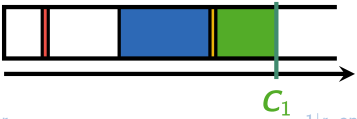


$\alpha_1 = 0.33$

Deterministic counterparts

Queue

Actual schedule



Virtual schedule

Result

The processing times are δ-NBUE if

$$\mathrm{E}[\boldsymbol{p}_j - t \mid \boldsymbol{p}_j > t] \leq \delta \cdot \mathrm{E}[\boldsymbol{p}_j]$$

for all $t \geq 0$ and $j = 1, \ldots, n$.

Proposition

If the processing times are δ-NBUE and the $\boldsymbol{\alpha}_j$ are chosen according to a distribution with density function f such that for all $x \in (0, 1]$

- i $\int_0^x \frac{\delta}{\delta + x - \alpha} \cdot f(\alpha)\,\mathrm{d}\alpha \leq (c-1) \cdot x$,
- ii $\left(1 + \int_0^1 \frac{\delta}{\delta + 1 - \alpha} \cdot f(\alpha)\,\mathrm{d}\alpha\right) \cdot \int_{1-x}^1 f(\alpha)\,\mathrm{d}\alpha \leq c \cdot x$,

for some $c > 1$, then RSOS is c-competitive.

Remark: Uniform distribution yields $c = 2$.

Proof Ideas

Let M_j be the mean busy time of job j in the virtual schedule.

- For every policy Π it holds that

$$\sum_{j=1}^{n} w_j \cdot \mathrm{E}[\boldsymbol{C}_j^{\Pi}] \geq \sum_{j=1}^{n} w_j \cdot (M_j + \mathrm{E}[\boldsymbol{p}_j]/2).$$

  (similar to Schulz (2008))

- Partition completion time of job j under RSOS into
  - start time in virtual schedule,
  - delay $\alpha_j \cdot \mathrm{E}[\boldsymbol{p}_j]$ caused by partially processing j in virtual schedule,
  - delay caused by jobs started before j in virtual schedule,
  - delay caused by jobs interrupting j in virtual schedule,
  - processing time $\boldsymbol{p}_j$.

  (similar to Goemans, Queyranne, Schulz, Skutella, and Wang (2002))

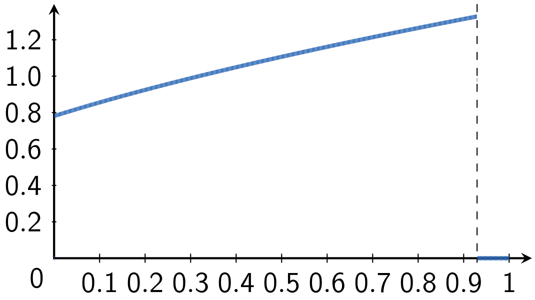
- Bound the expected values separately, resulting in

$$\mathrm{E}[\boldsymbol{C}_j^{\mathbf{RSOS}}] \leq c \cdot (M_j + \mathrm{E}[\boldsymbol{p}_j]/2).$$

The End

Numeric Computations

For $\delta = 1$ (NBUE) the following density function yields competitive ratio smaller than 1.783.



Derandomization

RSOS can be transformed to a deterministic offline policy.

Thank you!

References

- A. S. Schulz: *Stochastic Online Scheduling Revisited*, COCOA 2008
- M. X. Goemans, M. Queyranne, A. S. Schulz, M. Skutella, and Y. Wang: *Single Machine Scheduling with Release Dates*, SIAM J. Discrete Math., 15(2):165–192, 2002

Known and New Upper Bounds

	approximation algorithm		rand. online algorithm		det. online algorithm	
	1	P	1	P	1	P
deterministic processing times	$1+\varepsilon$[1]		1.686[2]	$2 - o_m(1)$[3,4]	2[5]	2.62[4] $1.791 + o_m(1)$[6]
stochastic processing times	~~3~~[7]	$3 - \frac{1}{m} + \max\{1, \frac{m-1}{m}\Delta\}$[7]	2 1.783 for $\delta = 1$	$2+\Delta$[8]	$\phi + 1$ ~~$2+\delta$~~[9]	$\max\{\phi+1, \frac{\phi+1}{2}\cdot\Delta + \frac{\phi+3}{2}\}$[8] $\frac{3}{2} + \delta\frac{2m-1}{2m} + \frac{\sqrt{4\delta^2+1}}{2}$[9]

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

$$\delta = \max_{j\in\{1,\dots,n\}} \sup\left\{ \frac{\mathrm{E}[\boldsymbol{p}_j - t \mid \boldsymbol{p}_j \geq t]}{\mathrm{E}[\boldsymbol{p}_j]} \;\middle|\; t \geq 0,\ \Pr[\boldsymbol{p}_j \geq t] > 0 \right\}$$

Remarks: $\delta \geq 1, \quad \Delta \leq 2\cdot\delta - 1.$

- $\delta = 1$: NBUE processing times

1: Afrati et al. 1999; 2: Goemans et al. 2002; 3: Schulz & Skutella 2002,
4: Correa & Wagner 2009; 5: Anderson & Potts 2004; 6: Sitters 2010;
7: Möhring, Schulz, & Uetz 1999; 8: Schulz 2008; 9: Megow, Uetz, & Vredeveld 2006