UNIVERSITY OF TWENTE.

Stochastic Scheduling on Unrelated Machines

Marc Uetz
m.uetz@utwente.nl

joint work with

M. Skutella & M. Sviridenko

Scheduling Workshop 2014

Why This Talk?

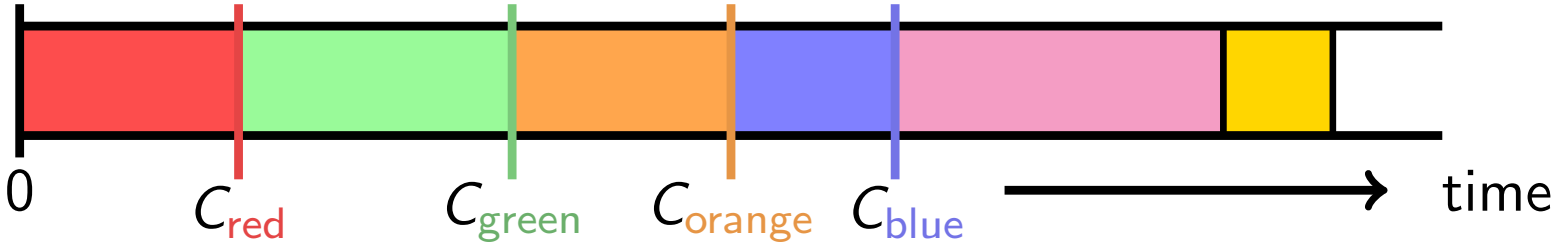
1. Don't know anything about stochastic scheduling?
   get acquainted with **interesting non-standard problem**

2. Do know something about stochastic scheduling?
   first results for **unrelated machines**, first time putting **time-indexed LP-relaxation** to work

Single Machine Scheduling

Given: n jobs j with weights $w_j > 0$, processing times $p_j \in \mathbb{Z}_{>0}$;

Task: sequence jobs on 1 machine; at most one job at a time;



Objective: minimize $\sum_j w_j C_j$ where $C_j = j$'s completion time;

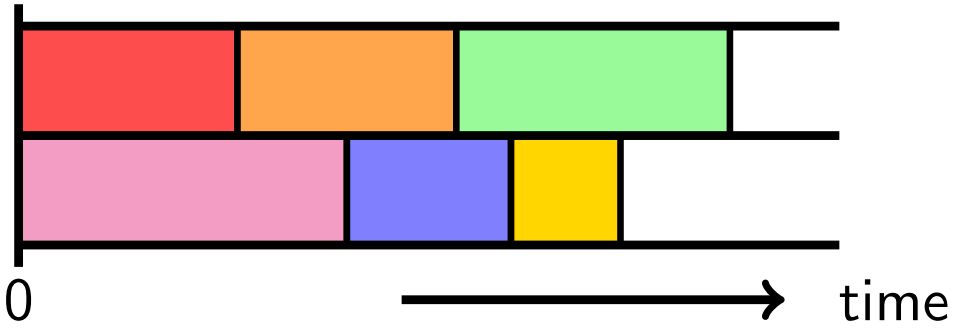
Theorem (Smith 1956)

Smith's rule, sequencing jobs in $\searrow$ order w_j/p_j is optimal $\square$

Identical Parallel Machine Scheduling

Given: n jobs as above; m identical parallel machines

Task: schedule each job on one machine; minimize $\sum_j w_j C_j$

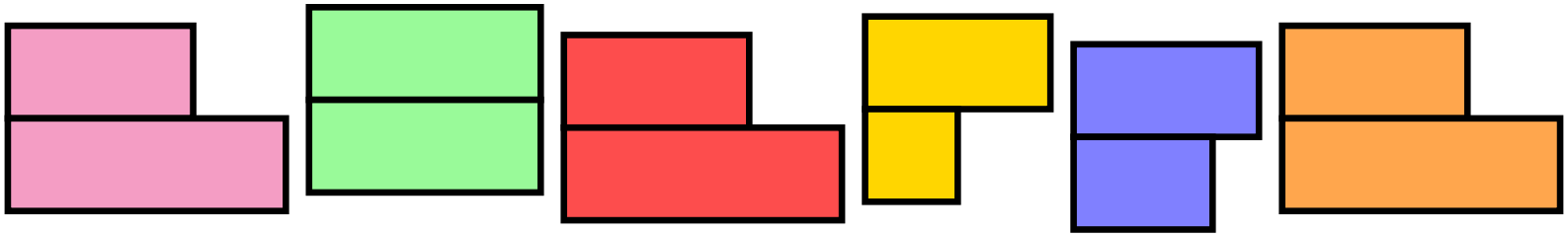


Theorem

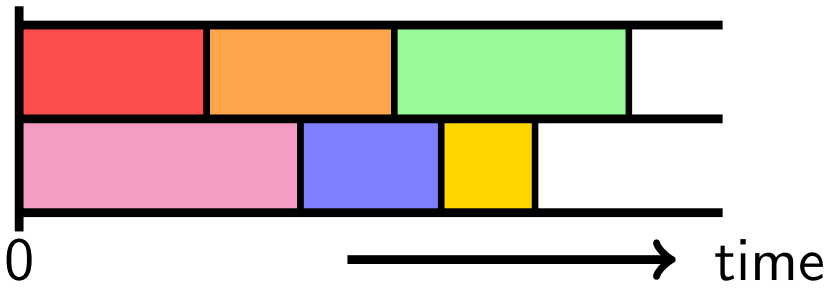
Problem is strongly NP-hard (Garey & Johnson, Problem SS13)
Smith's rule: tight 1.21-approximation (Kawaguchi & Kyan, 1986)
There exists a PTAS (Skutella & Woeginger, 2000)

Unrelated Machine Scheduling

Given: m machines, machine-dependent processing times p_{ij}



Task: schedule each job on one machine; minimize $\sum_j w_j C_j$



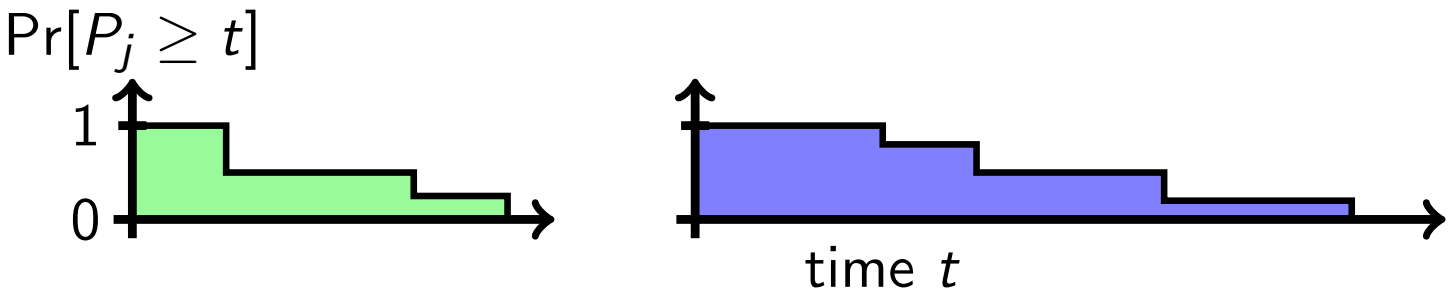
Theorem

Problem is APX-hard (Hoogeveen et al., 2002).

Exists $\frac{3}{2}$*-approximation* (Schulz & Skutella, 2002; Skutella, 2001).

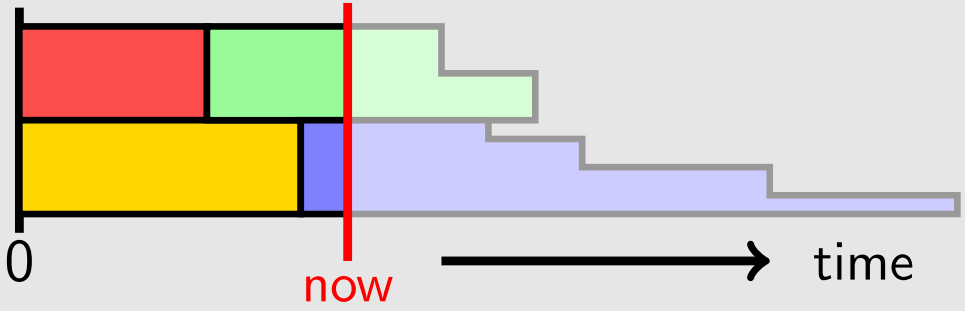
Stochastic Scheduling

- processing times (independent) random variables P_j (or P_{ij})
- assumption: probability distributions of all jobs are known



Solution: Non-anticipatory scheduling policy Π

Decisions based on information **up to now** and a priori knowledge about P_j (or P_{ij}); no further information about the future.



Optimal Policies

Definition (Optimality)

On instance I, $\Pi(I) =$ random variable; call Π^{OPT} optimal if it achieves

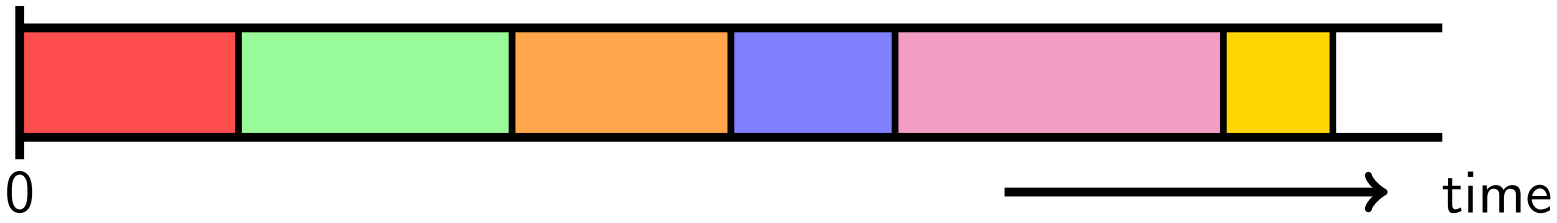
$$\inf\{\ \mathbb{E}[\Pi(I)] \mid \Pi \text{ non-anticipatory policy }\}$$

Simple & Greedy Scheduling Policy: WSEPT

Equivalent of Smith's rule:
Schedule jobs greedily in order of decreasing $w_j/\mathbb{E}[P_j]$

Theorem (Rothkopf 1966)

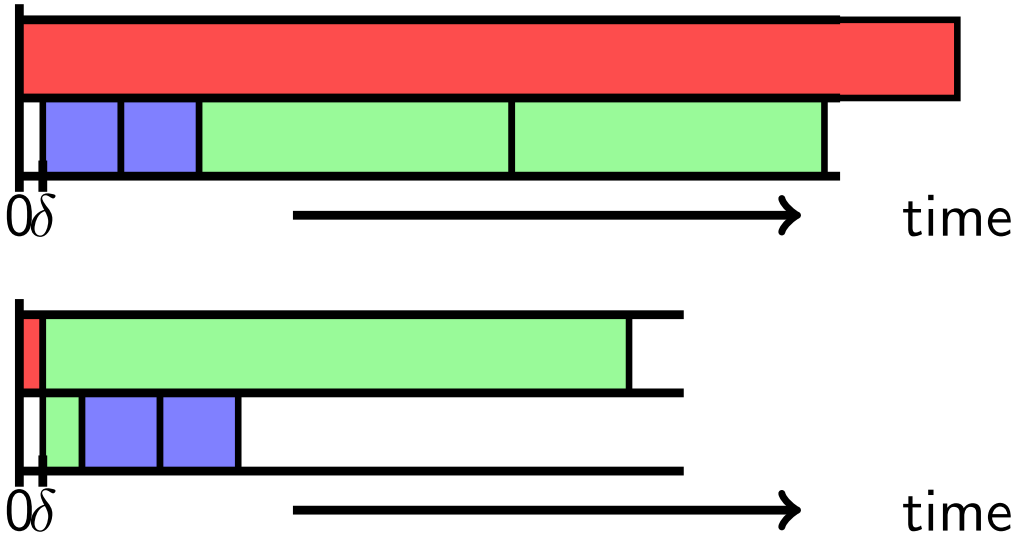
For single machine scheduling, WSEPT is optimal.



Asymptotic optimality of WSEPT in stochastic scheduling on identical parallel machines was proved by Weiss (1990, 1992).

On Optimal Policies (U. 2003)

Skipping details; $\exists$ (weird) instances where optimum policy must use deliberate idleness. . .



An optimal policy Π

- need not be **greedy** (deliberate idleness)
- need not be **elementary** (jobs started not only upon C_j)

Performance Metric

Recall Definition Optimality

On instance I, $\Pi(I) =$ random variable; call Π^{OPT} optimal if it achieves

$$\inf\{\ \mathbb{E}[\Pi(I)] \mid \Pi \text{ non-anticipatory policy }\}$$

- optimal policies can be (very) complicated (idleness)
- optimal policies NP-hard to compute; even PSPACE-hard?
- given Π, computing $\mathbb{E}[\Pi(I)]$ can be #P-hard (Hagstrom, 1988)

Definition (Approximation)

Policy Π has **performance guarantee** $\alpha \geq 1$, if for all instances I

$$\mathbb{E}[\Pi(I)] \ \leq \ \alpha \ \ \mathbb{E}[\Pi^{\mathsf{OPT}}(I)]$$

Other metrics are possible, e.g. Steger et al. (2002, 2004)

Approximation Algorithms Stochastic Scheduling

Möhring, Schulz & U. (1999)
First approximation algorithms based on LP relaxations in completion time variables, C_j^{LP}.

E.g., WSEPT has a performance guarantee $(\frac{3}{2} + \frac{\Delta}{2})$.

Skutella & U. (2005)
Extension to scheduling with precedence constraints.

Megow, U. & Vredeveld (2006), Chou et al. (2006)
Combination of stochastic and online scheduling.

Schulz (2008)
Some improved and simpler results.

All results restricted to **identical machines**; rely on one and the same **LP relaxation in completion time variables**.

LP Relaxation in C_j Variables

At the **core of all results**: LP relaxations that use this class of valid inequalities (U., 1996)

$$\sum_{j \in S} \mathbb{E}[P_j]\,\mathbb{E}[C_j^{\Pi}] \;\geq\; \frac{1}{2m}\left(\sum_{j \in S} \mathbb{E}[P_j]\right)^2 + \frac{1}{2}\sum_{j \in S} \mathbb{E}[P_j]^2 - \frac{m-1}{2m}\sum_{j \in S} \mathbb{V}\mathrm{ar}[P_j]$$

Generalizes relaxations for deterministic models (Wolsey, 1985; Queyranne, 1993 & 1995; Hall et al., 1997)

But: Not clear how to generalize to **unrelated machines**

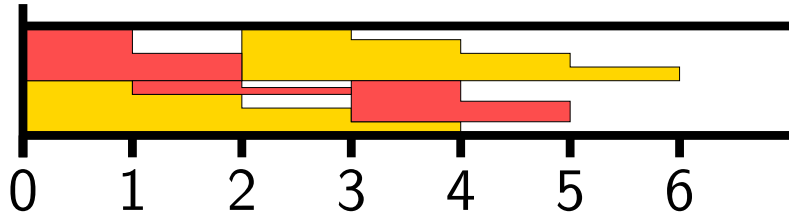
Towards Time-Indexed LP Relaxation

Consider instance I and non-anticipatory policy Π:

$$x_{ijt} := \Pr[\Pi \text{ starts job } j \text{ on machine } i \text{ at time } t \in \mathbb{Z}_{\geq 0}]$$

Important properties of x_{ijt} (Π non-anticipatory!):

- $\mathbb{E}[C_j] = \sum_{i,t} \big(t + \mathbb{E}[P_{ij}]\big)\, x_{ijt}$
- $\sum_{i,t} x_{ijt} = 1 \quad$ for all jobs j
- $\Pr\big[i \text{ processes } j \text{ in } [s, s+1]\big] = \sum_{t=0}^{s} x_{ijt} \Pr[P_{ij} > s - t]$

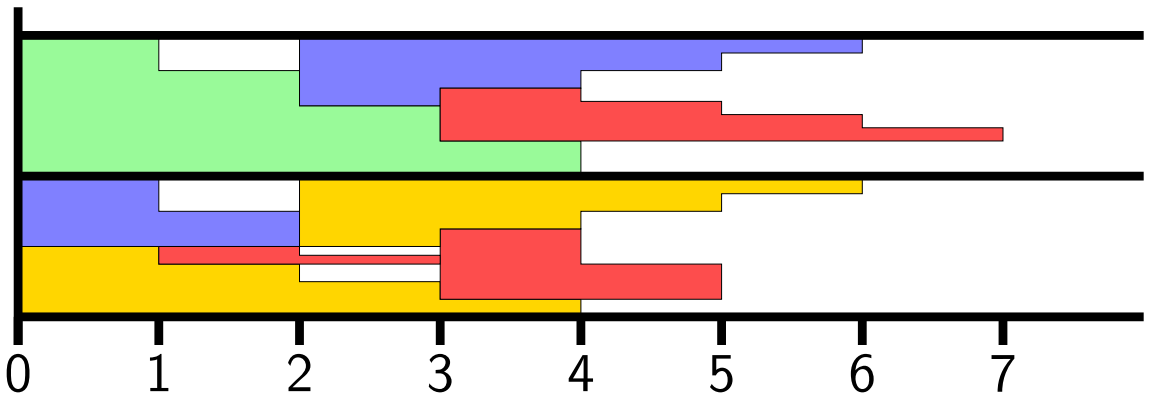


- $\displaystyle\sum_j \sum_{t=0}^{s} x_{ijt} \Pr[P_{ij} > s - t] \leq 1 \quad$ for each machine i and time s

Time-Indexed LP Relaxation for Stochastic Scheduling on Unrelated Machines

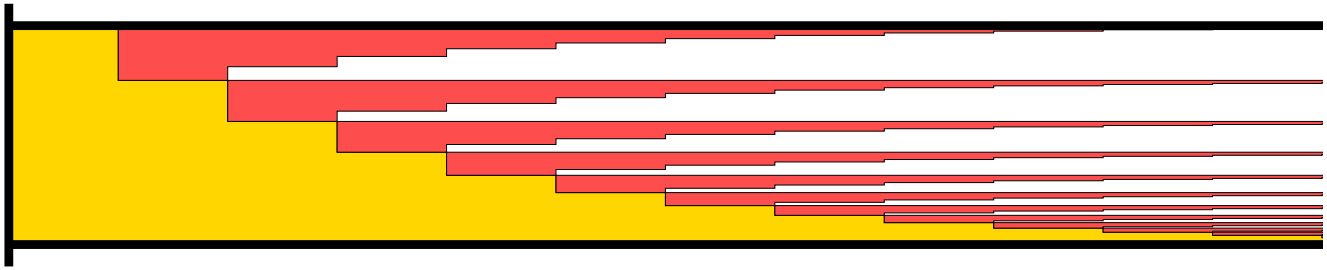
$$
\begin{aligned}
\min \quad & \sum\nolimits_{i,j,t} w_j \left(t + \mathbb{E}[P_{ij}]\right) x_{ijt} \\
\text{s.t.} \quad & \sum\nolimits_{i,t} x_{ijt} = 1 && \text{jobs } j, \\
& \sum\nolimits_{j} \sum\nolimits_{t=0}^{s} x_{ijt} \Pr[P_{ij} > s - t] \leq 1 && \text{machines } i, \text{ times } s, \\
& x_{ijt} \geq 0 && \text{jobs } j, \text{ machines } i, \text{ times } t.
\end{aligned}
$$

Example:

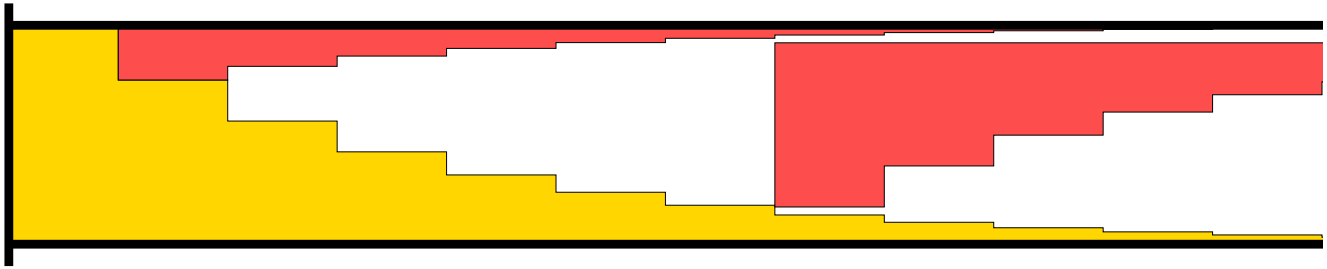


Optimal Policy May Yield Infinite LP Solution

Two identical jobs with exponentially distributed processing times:



But: There are feasible LP solutions that are finite, e.g.



Theorem

i *There is always an optimal LP solution that is finite.*

ii *The LP can be solved efficiently by an FPTAS.*

LP-Based Scheduling Policy

Algorithm

1. find an optimal (or approximate) LP solution (x_{ijt});
2. assign each job j independently at random to a machine i with
$$\Pr\big[j \text{ assigned to } i\big] = \sum_t x_{ijt} \quad ;$$
3. apply WSEPT rule on each machine;

Theorem

The expected value of the schedule is at most $\frac{3}{2} + \frac{\Delta}{2}$ times the value of the underlying LP solution x.

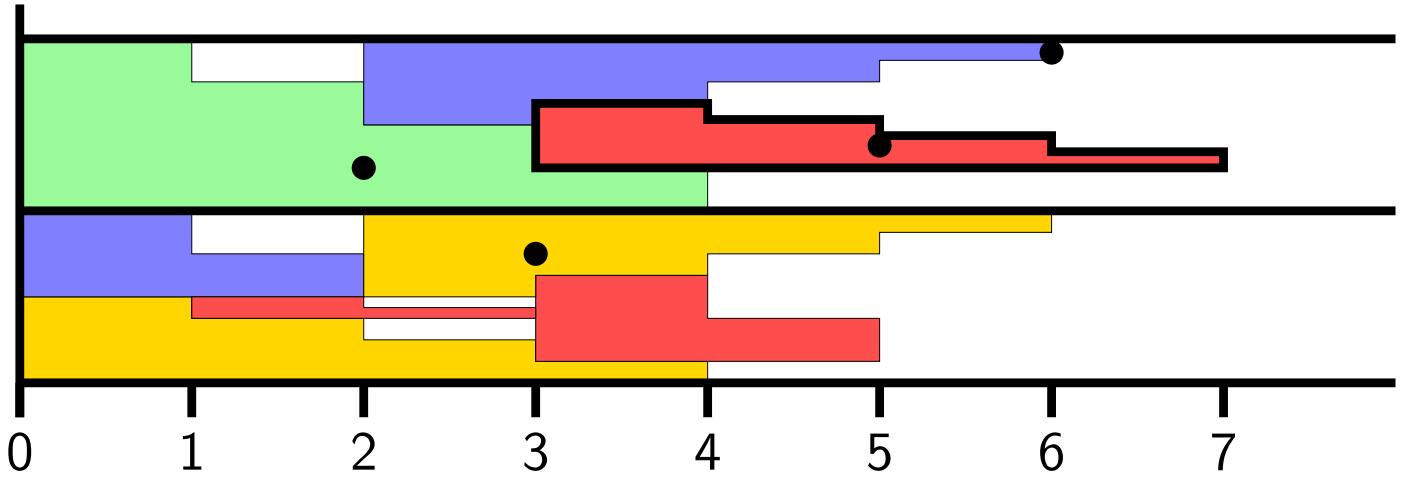
$$\Delta \geq \mathbb{CV}^2[P_{ij}] := \frac{\mathbb{V}\mathrm{ar}[P_{ij}]}{\mathbb{E}^2[P_{ij}]} \qquad \text{for all } P_{ij}$$

Proof of Performance Ratio

Idea: Analyze more complicated and provably worse algorithm:

1. find an optimal (or approximate) LP solution (x_{ijt});
2. for each job j
   a) choose pair (i, t) independently at random with probability x_{ijt};
   b) choose $r \in \mathbb{Z}_{\geq 0}$ indep. at random with probability $\frac{\Pr[P_{ij} > r]}{\mathbb{E}[P_{ij}]}$;
   c) set the *tentative start time* of j to $s := t + r$;
3. on each machine, sequence jobs by incr. tentative start times;

Example:



Proof of Performance Ratio

Lemma

Total exp. processing before job $j \to (i,s) \leq$ tent. start time $s + \frac{1}{2}$

$$\text{Thus, } \mathbb{E}[C_j] \leq \sum_i \sum_{s \in \mathbb{Z}_{\geq 0}} \left(s + \tfrac{1}{2} + \mathbb{E}[P_{ij}]\right) \Pr\big[j \to (i,s)\big]$$

$$\text{with } \Pr\big[j \to (i,s)\big] = \sum_{t=0}^{s} x_{ijt} \frac{\Pr[P_{ij} > s - t]}{\mathbb{E}[P_{ij}]} \qquad //s = t + \underbrace{(s-t)}_{r}$$

$$\Rightarrow \mathbb{E}[C_j] \leq \sum_i \sum_{t \in \mathbb{Z}_{\geq 0}} x_{ijt} \left(t + \mathbb{E}[P_{ij}] + \sum_{r \in \mathbb{Z}_{\geq 0}} (r + \tfrac{1}{2}) \frac{\Pr[P_{ij} > r]}{\mathbb{E}[P_{ij}]} \right)$$

$$\leq \left(\frac{3}{2} + \frac{\Delta}{2} \right) C_j^{\mathrm{LP}}$$

□

"2nd moment" Lemma

$$\sum_{r=0}^{\infty} (r + \tfrac{1}{2}) \Pr[P_{ij} > r] = \frac{1 + \mathbb{CV}[P_{ij}]^2}{2} \mathbb{E}[P_{ij}]^2$$

Matching Lower Bounds

Our scheduling policy is a fixed assignment policy, i.e.

- jobs are assigned to machines right in the beginning
- assignment not changed after collecting information over time

Performance bound has "right" order of magnitude...

Theorem

Even for identical machines, any fixed assignment policy can have a performance ratio $\geq \frac{(1-\delta)\Delta}{2}$ *for any* $\delta > 0$*, for large* m*.*

Theorem

Even for a single machine, the LP can have an optimality gap as large as $\Delta/2$*.*

Concluding Remarks

- performance guarantee is $2 + \Delta + \varepsilon$ when jobs also have individual release dates r_j
- match best known bounds for identical machines (up to ε)
- match best known bounds for deterministic scheduling (up to ε)
- **still open:** getting rid of Δ; need (meaningful bounds for) adaptive policies

Thanks for listening !