Scheduling with Uncertain Processing Times

Marc Uetz
m.uetz@utwente.nl

Berkeley, 2016
This Talk

Mixture of

1. (mini) introductory lecture, stochastic scheduling

2. along with some recent results (unrelated machine scheduling) [Skutella, Sviridenko & U. 2016]
1 Setting the Scene

2 Stochastic Scheduling

3 Unrelated Machines & Time-indexed LP

4 Final Remarks
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 order \( w_j/p_j \) is optimal
Identical Parallel Machine Scheduling

Given: \( n \) jobs as above; \( m \) identical parallel machines

Task: schedule each job on any one machine; minimize \( \sum_j w_j C_j \)

**Theorem**

*Problem is strongly NP-hard* \[ \text{[Garey & Johnson, Problem SS13]} \]

*Smith’s rule: tight 1.21-approximation* \[ \text{[Kawaguchi & Kyan, 1986]} \]

*There exists a PTAS* \[ \text{[Skutella & Woeginger, 2000]} \]
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

Exists \( (\frac{3}{2} - c) \)-approximation

\[ \text{[Hoogeveen et al., 2002]} \]

\[ \text{[Bansal, Srinivasan, Svensson, 2016]} \]
Main Result

Theorem (Skutella, Sviridenko, U. 2016)

**Stochastic unrelated machine scheduling** has a $(\frac{3+\Delta}{2})$-approximation.

$\Delta = \text{bounds the (squared) coeff. of variation of processing times}$
1 Setting the Scene

2 Stochastic Scheduling

3 Unrelated Machines & Time-indexed LP

4 Final Remarks
Stochastic Scheduling

Processing time = (independent) random variables $P_j$ (or $P_{ij}$); all known to us

$$\Pr[P_j \geq t]$$

Solution: Non-anticipatory scheduling policy $\Pi$

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

On instance $I$ with policy $\Pi$

$\Pi(I) := \text{cost of policy } \Pi \text{ on } I$, is a random variable

**Definition (Optimal Policy)**

Call $\Pi^{OPT}$ optimal if it achieves

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

Existence follows from [Möhring, Radermacher, Weiss 1985]
An Example

$n = 4$ jobs, unit weights

blue jobs: $P_j = 1$

green jobs: $P_j = \begin{cases} 
0 \text{ probability } 4/5 \\
10 \text{ probability } 1/5 
\end{cases}$ (note $\mathbb{E}[P_j] = 2$)
Stochastic World

Tradeoff: better to delay

large $\mathbb{E}[P_j]$ or large $\text{Pr}(P_j \text{“large”})$ (heavy tail) ?

Claim

Unique 2-machine optimal policy: green, blue $\rightarrow$ green $\rightarrow$ blue

[with $\Pi(I) = \mathbb{E}[\sum_j C_j] = 6.92$.]

Just work it out
Stochastic World: Deliberate Idleness

Theorem (U. 2003)

There are instances where only optimal policy \textit{deliberately leaves machines idle}.
Approximation Algorithms

Optimal policies

- intuitively complex, exponential size decision tree; definitely NP(AXP)-hard, . . .
- only computing $\mathbb{E}[\Pi(I)]$ can be $\#P$-hard

[Hagstrom, 1988]

Definition (Approximation)

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

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

Our adversary is non-anticipatory, too!
Approximation Algorithms

Möhring, Schulz & U. [JACM, 1999]
First LP-based approximation algorithms
e.g.: Smith’s rule has performance guarantee $(\frac{3+\Delta}{2})$.

Skutella & U. [SICOMP, 2005]
Extension to problems w. precedence constraints.

Megow, U. & Vredeveld [MOR, 2006] as well as Chou et al. [2006], Schulz [2008]
Stochastic jobs that arrive online.

All for **identical** machines; use LP lower bound on $\Pi^{OPT}$
LP Relaxation Identical Machines

Core ingredient: stochastic version of load inequalities
[Möhring, Schulz & U., 1999]

\[
\sum_{j \in S} \mathbb{E}[P_j] \mathbb{E}[C_j] \geq \frac{1}{2m} \left( \sum_{j \in S} \mathbb{E}[P_j] \right)^2 + \frac{1}{2} \sum_{j \in S} \mathbb{E}[P_j]^2 - \frac{m - 1}{2m} \sum_{j \in S} \text{Var}[P_j] \quad \forall \text{ subsets } S
\]

Generalizes LPs used earlier [Wolsey, 1985; Queyranne, 1993 & 1995; Hall, Schulz, Shmoys & Wein 1997]

But: doesn’t generalize to unrelated machines
1. Setting the Scene
2. Stochastic Scheduling
3. Unrelated Machines & Time-indexed LP
4. Final Remarks
Structuring The Input

Theorem

At a cost of $(1 + \varepsilon)$, may assume w.l.o.g. input is integer valued.
Instance $I$ and non-anticipatory policy $\Pi$, define

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

second blue job, $j = 4$, has

$$x_{1,4,0} = 16/25$$
$$x_{2,4,1} = 8/25$$
$$x_{1,4,10} = 1/25$$
Time-Indexed LP Relaxation

Instance $I$ and non-anticipatory policy $\Pi$

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

Properties of $x_{ijt}$ ($\Pi$ non-anticipatory!):

- $\mathbb{E}[C_j] = \sum_{i,t} (t + \mathbb{E}[P_{ij}]) x_{ijt}$
- $\sum_{i,t} x_{ijt} = 1$ for all jobs $j$
- $\Pr[i \text{ processes } j \text{ in } [s, s + 1]] = \sum_{t=0}^{s} x_{ijt} \Pr[P_{ij} > s - t]$
- $\sum_{j} \sum_{t=0}^{s} x_{ijt} \Pr[P_{ij} > s - t] \leq 1$ for each machine $i$ and time $s$
Time-Indexed LP Relaxation

\[ \min \sum_{i,j,t} w_j \left( t + \mathbb{E}[P_{ij}] \right) x_{ijt} \]

s.t.
\[ \sum_{i,t} x_{ijt} = 1 \quad \text{jobs } j, \]
\[ \sum_j \sum_{t=0}^s x_{ijt} \Pr[P_{ij} > s - t] \leq 1 \quad \text{machines } i, \text{ times } s, \]
\[ x_{ijt} \geq 0 \quad \text{jobs } j, \text{ machines } i, \text{ times } t. \]

Example:
Technical Detail: Infinite LP Solution?

Two identical jobs with exponentially distributed processing times:

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

Theorem

∃ finite optimal LP solution; LP can be solved efficiently (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
   \[
   \Pr[j \text{ assigned to } i] = \sum_t x_{ijt} ;
   \]
3. apply Smith’s rule on each machine (that’s optimal!);

**Theorem (Skutella, Sviridenko & U. 2014)**

*This algorithm is a \((3 + \Delta \over 2)\)-approximation.*

\[
\Delta \geq CV^2[P_{ij}] := \frac{\text{Var}[P_{ij}]}{E^2[P_{ij}]} \quad \text{for all } P_{ij}
\]
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

Key Lemma

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

Which yields

$$E[C_j] \leq \sum_i \sum_{s \in \mathbb{Z}_{\geq 0}} (s + \frac{1}{2} + E[P_{ij}]) \Pr [j \rightarrow (i, s)]$$

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

Performance bounds are tight . . .

Theorem

Any “fixed assignment” policy can have optimality gap $\Delta/2$.
[identical machines].

Theorem

The time-indexed LP can have optimality gap $\Delta/2$.
[1 machine].
1 Setting the Scene

2 Stochastic Scheduling

3 Unrelated Machines & Time-indexed LP

4 Final Remarks
Final Remarks

To improve, **only adaptivity** is not enough

[Smith’s rule, even though adaptive, can be as bad as $\Omega(\sqrt{\Delta})$]

open problems are

1. const. approximation (indep. of $\Delta$)?
   
   **Im, Moseley & Pruhs** [STACS 2015] get
   
   $$O(\log^2 n + m \log n)$$ — approximation

   (identical machines); balancing $\mathbb{E}[P_j]$ vs. $\Pr[P_j \text{ "large"}]$

2. nontrivial hardness / bounds on approximation?
Beating $\Omega(\Delta)$: Im, Moseley, Pruhs 2015

Which jobs first? we want to exclude jobs with...

- $\Pr[P_j = \text{“large”}]$ maximal
- $\mathbb{E}[P_j]$ maximal

if ALG schedules $k$ jobs $A$, OPT schedules $k$ jobs $A^*$, then

Core Lemma

$\Pr[\text{ALG}(A) \text{ blocks all machines beyond } \tau] \leq \Pr[\text{OPT}(A^*) \text{ blocks all machines beyond } \tau/\alpha]\$

$\alpha = \Theta(m \log n)$
Some Reading

- Approximation in Stochastic Scheduling: J ACM 46, 1999
- When Greediness Fails: Examples from Stochastic Scheduling OR Lett 31, 2003
- Stochastic Machine Scheduling with Precedence Constraints SIAM Comp 34, 2005
- Models and Algorithms for Stochastic Online Scheduling MOR 31, 2006
- Unrelated Machine Scheduling with Stochastic Processing Times MOR 41, 2016
  
  [w/ Megow, Möhring, Schulz, Skutella, Sviridenko, Vredeveld]

- Stochastic Scheduling of Heavy-Tailed Jobs Im, Moseley & Pruhs, STACS, 2015