Approximation algorithms for the two-stage stochastic scheduling problem with reservation costs

Lin Chen

Joint worked with Nicole Megow, Roman Rischke and Leen Stougie
Goal: Buy time slots on the machine to schedule jobs
Introduction

Pay 5c units of money and get 5 time slots
Stochastic setting: distribution over scheduling instances
-- polynomial scenarios, independent-activation, black-box, etc.
Introduction—polynomial scenarios

Two buying options: Global (first stage) buying—buy before I see the instance

Scenario 1 ($p_1$)

Scenario 2 ($p_2$)

Machine

Pay 6c units of money and get 6 time slots

Lin Chen
Introduction—polynomial scenarios

Two buying options: Local (second stage) buying—buy after I see the instance

Scenario 1 \( (p_1) \)
- Pay \( 5\lambda_1 c \) money and get 5 slots

Scenario 2 \( (p_2) \)
- Pay \( 2\lambda_2 c \) money and get 2 slots

Lin Chen
Two buying options: Combining global and local buying

- Pay $2c + 3\lambda_1 c$ money and get 5 slots
- Pay $2c$ money and get 2 slots
Input: --- Distribution over scheduling instances
--- Global buying cost $c$, local buying cost $\lambda_i c$

Objective: Minimizing the expected buying cost plus scheduling cost
--- Buying cost
$c \cdot \text{# slots globally bought} + \sum p_i \cdot (\lambda_i \cdot \text{# slots locally bought in scenario } i)$
--- Scheduling cost
$\sum p_i \cdot (\sum w_{ij} C_{ij})$
Introduction—Equivalent formulation

Identical machines

- Machine $M_1$
- Machine $M_2$
- Machine $M_N$

Dedicated jobs

- Blue
- Yellow
- Red

Lin Chen
Introduction—Equivalent formulation

Machine (scenario) 1: Identical machines

Machine (scenario) 2: Dedicated jobs

Machine (scenario) N: Global buying

Lin Chen
Introduction—Unrelated machines

Machine $M_1$

Machine $M_2$

Machine $M_m$

Scenario 1

......

Scenario N

Scenario-Dedicated jobs
Introduction—Unrelated machines

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Scenario N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine $M_1$</td>
<td>Machine $M_1$</td>
</tr>
<tr>
<td>Machine $M_2$</td>
<td>Machine $M_m$</td>
</tr>
<tr>
<td>Machine $M_m$</td>
<td>Machine $M_m$</td>
</tr>
</tbody>
</table>

Global buying—buying across scenarios

Scenario-Dedicated jobs
Introduction—Unrelated machines

Local buying—buying per scenario

Scenario 1
Machine $M_1$
Machine $M_2$
Machine $M_m$

......

Scenario N
Machine $M_1$
Machine $M_m$

Scenario-Dedicated jobs
Introduction—Unknown distribution?

**Sample Average Approximation (SAA) method**

--- Sample N times from scenario distribution (black box)

--- Solve the sample-average problem:

*Approximating the scenario probability by sampling frequency*

1. The sample average problem (SAA-problem) has a polynomial number of scenarios
2. With a proper number of samples, an $\alpha$-approx. for SAA-problem implies an $\left(\alpha + \epsilon\right)$-approx. for the two-stage problem (Charikar, Chekuri, Pal 2005)
Related work

- On stochastic optimization
  - Stochastic programming since fifties (Beale, 1955; Danzig, 1955)
  - Approximations for two/multi-stage stochastic variants of classical optimization problems
    - Service-provisioning (Dye, Stougie, Tomasgard '03)
    - Covering and network design (Immorlica, Karger, Minkok, Mirrokni '04)
    - Subset selection problem (Shmoys, Sozio '07)

- On $R|\text{pmtn}, r_j| \sum w_j C_j$
  - 3-approx. (Skutella '01)
  - $(2 + \epsilon)$-approx. (Queyranne & Sviridenko '02)
  - 2.78-approx. (1.81-approx. if $r_j = 0$) (Sitters '08)
Our results

- 8-approx. for two-stage unrelated machine scheduling
- Better approximation algorithms for some special cases:
  - 3-approx. for unrelated machines if there is only global (first stage) buying
  - 3-approx. for two stage single machine scheduling
  - 2-approx. for single machine if there is only global (first stage) buying
- Polynomial algorithms for some special cases (constant number of scenarios)
Brief introduction of the algorithm

\[
\begin{align*}
\text{min} & \quad \sum_t c \cdot x_t^l + \sum_{i=1}^k p_i \left( \sum_{j \in S_i} w_{ij} C_{ij}^{LP} + c \cdot \lambda_i \sum_t x_{it}^{ll} \right) \\
\text{s.t.} & \quad \sum_{j \in S_i} y_{hijt} \leq x_t^l + x_{it}^{ll} \quad \forall \text{scenario } i, \text{ machine } h, \text{ time } t \\
& \quad \sum_h y_{hijt} \leq x_t^l + x_{it}^{ll} \quad \forall \text{scenario } i, \text{ job } j, \text{ time } t \\
& \quad x_t^l + x_{it}^{ll} \leq 1 \quad \forall \text{scenario } i, \text{ time } t \\
& \quad \sum_t \sum_h \frac{y_{hijt}}{p_{hij}} = 1 \quad \forall \text{scenario } i, \text{ job } j \\
& \quad C_{ij}^{LP} = \sum_t \sum_h t \cdot \frac{y_{hijt}}{p_{hij}} \quad \forall \text{scenario } i, \text{ job } j \\
& \quad y_{hijt} = 0 \quad \forall t \leq r_{ij} \\
& \quad x_t^l, x_{it}^{ll}, y_{hijt} \in [0, 1].
\end{align*}
\]
Brief introduction of the algorithm

- The LP handles well the trade-off between buying and scheduling, except that
  - $C_{ij}^{LP}$ is the 'average' completion time instead of the real completion time
  - Slots are fractionally bought according to $x_t^l$ and $x_{it}^{ll}$
How to handle the completion times?

- If we expand the whole solution by $1/\alpha$ times, the completion time of job $j$ becomes $\lceil 1/\alpha C_{ij}(\alpha) \rceil$ (Schulz & Skutella, 1997)

\[
\int_0^1 C_{ij}(\alpha) d\alpha = \sum_t \sum_h y_{ht} / p_{ht} \cdot (t - 1/2) = C_{ij}^{LP} - 1/2
\]

(Goemans, 1997. Queyranne & Sviridenko, 2001)
How to handle the buying (assume that there is only global buying)?
Brief introduction of the algorithm—only global buying

**Aggregated buying**

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Machine $M_1$</th>
<th>Machine $M_2$</th>
<th>Machine $M_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

**Scenario N**

<table>
<thead>
<tr>
<th>Machine $M_1$</th>
<th>Machine $M_m$</th>
</tr>
</thead>
</table>
Brief introduction of the algorithm—only global buying

Aggregated buying

Scenario 1
- Machine $M_1$
- Machine $M_2$
- Machine $M_m$

......

Scenario N
- Machine $M_1$
- Machine $M_m$
Brief introduction of the algorithm—only global buying

Scenario 1

Machine $M_1$

Machine $M_2$

Machine $M_m$

......

Scenario N

Machine $M_1$

Machine $M_m$
Brief introduction of the algorithm—only global buying

Postponing too much
— Extra buying

Scenario 1

Machine $M_1$

Machine $M_2$

Machine $M_m$

......

Scenario N

Machine $M_1$

Machine $M_m$
Theorem

Suppose there is only global buying. If in a fractional solution the total buying cost is $LP^p$ and job $j$ (of scenario $i$) completes at time $\hat{C}_{ij}$, then with an integral buying of cost at most $2LP^p$, a new solution could be derived so that job $j$ completes at time $\hat{C}_{ij} + 1$. 
Algorithm for the special case with only the global (first stage) buying

- Specify $\alpha \in (0, 1)$ according to some distribution $f(\alpha)$
- Solve LP and expand the solution by $1/\alpha$ times
- Apply the theorem to get an integral buying
- Construct a feasible schedule
Observation: An $\alpha$-approx. for only global buying also implies an $\alpha$-approx. for only local buying

- The problem with only local buying could be viewed as $|S|$ separate problems of only global buying (and one scenario)
Algorithm for the two-stage stochastic scheduling problem

- Separate jobs into $J_1$ and $J_2$ \((\text{How?})\)
- Apply $\alpha$-approx. for $J_1$ with only global buying
- Apply $\alpha$-approx. for $J_2$ with only local buying
- Merge the two solutions \((\text{How?})\)
**Lemma**

Given \((x^I_t, x^II_{it}, y_{hijt})\) as an optimum solution for LP, there exists a feasible solution \((x^I_t(sep), x^II_{it}(sep), y_{hijt}(sep))\) satisfying the following separation property:

- A slot is either bought in the first stage or in the second stage, i.e.,
  \(ST = \{1, 2, \cdots, T\} = ST_1 \cup ST_2, \text{ s.t.}
  \)
  - \(x^I_t(sep) = 0, \text{ for } t \in ST_2,
  \)
  - \(x^II_{i,t}(sep) = 0, \text{ for } t \in ST_1.
  
- A job is either completely scheduled in slots bought in the first stage, or completely scheduled in slots bought in the second stage, i.e.,
  \(J = J_1 \cup J_2, \text{ s.t.}
  \)
  - \(j \in J_1, \text{ then } y_{h,i,j,t}(sep) = 0 \text{ for } t \in ST_2,
  \)
  - \(j \in J_2, \text{ then } y_{h,i,j,t}(sep) = 0 \text{ for } t \in ST_1.
  
- Objective value at most \(4 \text{OPT}_{LP}\) (or even \((2 + \epsilon) \text{OPT}\)).
Brief introduction of the algorithm—Merge solutions

Odd slots—global
Even slots—local

Scenario 1
Machine $M_1$
Machine $M_2$
Machine $M_m$

......

Scenario N
Machine $M_1$
Machine $M_m$
Thank you!