Late Work Scheduling in Online and Offline Mode

Małgorzata Sterna
Xin Chen, Jacek Błażewicz, Xin Han
Kateryna Czerniachowska

Poznań University of Technology
Institute of Computing Science
Poland

(malgorzata.sterna@cs.put.poznan.pl)

New Challenges in Scheduling Theory – Aussois
March 29 – April 2, 2016
Outline

- Late Work Scheduling
- Problem Definition
- Offline Cases
- Online Cases
- Conclusions & Future Research Directions
Late Work

- late work criteria
  minimize the amount of work executed after the predefined due date
  (Błażewicz, 1984)

\[
Y_j = \min\{\max\{0, C_j - d_j\}, p_j\}
\]
Late Work

\[ Y_j = \min\{\max\{0, C_j - d_j\}, p_j\} \]

\[ = \min\{D_j, p_j\} \]

New Challenges in Scheduling – Aussois – March 30, 2016
Motivations/Applications

jobs: data from sensing devices, crops, fertilizers/pesticides, customer orders, pieces of software

machine(s): control algorithm, agriculture machine(s), production system, software developer(s)


M. Sterna, A survey of scheduling problems with late work criteria, Omega 39 (2011) 120–129
Single Machine Problems

- **polynomially solvable**
  - $1|\text{pmtn}|Y$ (Potts, Van Wassenhove, 1991)
  - $1|r_j, \text{pmtn}|Y$ (Hochbaum, Shamir, 1990)
  - $1|\text{pmtn}|Y_w$ (Hariri, Potts, Van Wassenhove, 1995)
  - $1|r_j, \text{pmtn}|Y_w$ (Hochbaum, Shamir, 1990)
  - $1|d_j=d|Y$ (Potts, Van Wassenhove, 1991)
  - $1|d_j=d|Y_w$ (Hariri, Potts, Van Wassenhove, 1995)

- **NP-hard**
  - $1||Y$ (Potts, Van Wassenhove, 1991)
  - $1|r_j|Y$ (Potts, Van Wassenhove, 1991)
  - $1||Y_w$ (Hariri, Potts, Van Wassenhove, 1995)
  - $1|B\geq n|Y$ (Zhang and Wang, 2005)
  - $1|p_j=1, \text{chains}|Y$ (Sterna, 2000)

- **DP-benevolent problem** (Woeginger, 2000)

- $1|p_j=p|Y$ (Lin, Hsu, 2005)

- **New Challenges in Scheduling – Aussois – March 30, 2016**
Dedicated Machines Problems

polynomially solvable

\[ O_2 | d_j = d | Y \]
(Błażewicz, Pesch, Sterna Werner, 2004)

\[ O | r_j, \text{ pmtn} | Y_w \]
(Błażewicz, Pesch, Sterna Werner, 2004)

NP-hard

\[ O_2 | d_j \in \{d_1, d_2\} | Y \]  (Leung, 2004)

\[ F_2 | d_j = d | Y \]  (Lin, Lin, Lee, 2006)
  (Sterna, 2007)

\[ F_2 | d_j \in \{d_1, d_2\} | Y \]  (Leung, 2004)

\[ F_2 || Y \]  (Leung, 2004)

\[ F | r_j | Y \]  (Pesch, Sterna, 2009)

\[ O_2 | d_j = d | Y_w \]
(Błażewicz, Pesch, Sterna, Werner, 2004)

\[ F_2 | d_j = d | Y_w \]
(Błażewicz, Pesch, Sterna, Werner, 2005)

\[ J_2 | d_j = d, n_j \leq 2 | Y_w \]
(Błażewicz, Pesch, Sterna, Werner, 2007)
Parallel Machines Problems

- Polynomially solvable
  - $P | r_j, p_{mtn} | Y$ (Leung, 2004)
  - $P | r_j, p_{mtn} | Y_w$ (Błażewicz, 1984)  
    (Błażewicz, Finke, 1987)
  - $Q | r_j, p_{mtn} | Y$ (Leung, 2004)
  - $Q | r_j, p_{mtn} | Y_w$ (Błażewicz, 1984)  
    (Błażewicz, Finke, 1987)  
    (Leung, 2004)
  - $P | r_j, p_j = 1 | Y_w$  
    (Sterna, 2000)
  - $Q | p_j = 1 | Y_w$

- NP-hard
  - $P | Y$ (Błażewicz, 1984)
  - $P_2 | p_j = 1, \text{chains} | Y$
Problem Definition – P|d_j=d|Y

- a set of jobs J={J_1, ..., J_n}
  with processing time p_j for job J_j
- a set of identical parallel machines M={M_1, ..., M_m}
- common due date d
- criterion: minimizing total late work

- offline case: the set of jobs J is known in advance
- online case: jobs appear „over list”

- P2|d_j=d|Y and P2|d_j=d, online over list|Y
- P|d_j=d|Y and P|d_j=d, online over list|Y
Problem Definition

- **P2\( |d_j=d\), online over list\(|Y\)**
  - lower bound
  - upper bound
  (online algorithm with finite competitive ratio)

- **P2\( |d_j=d\)\(|Y\)**
  - computational complexity
Offline Case – P2|d_j=d|Y

- NP-hard due to the transformation from the Partition Problem

**Partition Problem**

\[ A = \{a_1, a_2, a_3, a_4, a_5, a_6\} \]

\[ \sum_{a_j \in A'} s(a_j) = \sum_{a_j \in A \setminus A'} s(a_j) ? \]

**Decision counterpart of P2|d_j=d|Y**

\[ Y = 0 ? \]

**New Challenges in Scheduling – Aussois – March 30, 2016**
Offline Case – P2|d_j=d|Y

- binary NP-hard due to the existence of pseudopolynomial dynamic programming $O(nd^2)$

$$f(j, A, B) = \min \{ f(j-1, \max\{0, A-p_j\}, B) + \max\{0, p_j-A\},$$

$$f(j-1, A, \max\{0, B-p_j\}) + \max\{0, p_j-B\} \}$$

$$Y^* = f(n, d, d)$$
binary NP-hard due to the existence of pseudopolynomial dynamic programming $O(nd^2)$

\[
f(j, A, B) = \min \{ f(j-1, \max\{0, A-p_j\}, B) + \max\{0, p_j-A\}, \]
\[
f(j-1, A, \max\{0, B-p_j\}) + \max\{0, p_j-B\} \}
\]

$Y^* = f(n, d, d)$
Offline Cases – \( P|d_j=d|Y \)

- \( P|d_j=d|Y \) is unary NP-hard due to the transformation from 3-Partition Problem
Online Case – P|d_j=d, online over list|Y

- To evaluate online methods the concept of early work is used

- Minimizing late work – maximizing early work for a common due date is similar to bin packing problem

- Online algorithm – Extended First Fit with finite competitive ratio

$$r_m = \frac{\sqrt{2m^2 - 2m + 1} - 1}{m - 1}$$
Extended First Fit with competitive ratio $r_m$

- set $t = 1$, initialize machine loads ($L_{i_0}^t = 0$)
- when job $J_t$ with processing time $p_t$ arrives, assign it to the first fitting machine i.e. not violating assumed ratio $(L_{i_{t-1}}^t + p_t \leq r_m d)$
- update machine loads and set $t = t + 1$

proof of competitive ratio based on case analysis
proof of lower bound of competitive ratio

\[ \sqrt{5} - 1 \approx 1.236 \]

based on classical adversary sequence

\[ d = \frac{\sqrt{5} + 1}{2} \]
Algorithm Extended First Fit with competitive ratio

\[ r_m = \frac{\sqrt{2m^2 - 2m + 1} - 1}{m - 1} \]

is optimal for \( m = 2 \)

\( r_2 \) equals to the lower bound \( \sqrt{5} - 1 \)
List Algorithms

offline algorithms
- assigning jobs to the machine with the minimum load in
  - shortest processing time (SPT) order
  - longest processing time (LPT) order

online algorithms
- assigning jobs in the input order
  - to the machine with minimum makespan (MM)
  - according to Extended First Fit (EFF)
List Algorithms

![Bar Chart]

The chart shows the optimal early work as a portion of the criterion upper bound versus the due date as a portion of the total processing time. The percentage values range from 99.70% to 100.00%.
List Algorithms

- Competitive ratio (optimal early work to heuristic early work)

Due date as a portion of the total processing time

Graph showing competitive ratios for different due dates compared to optimal work and heuristic early work for different algorithms.
List Algorithms

**Percentage of Best Heuristic Solutions**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
<th>80%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>83.82</td>
<td>37.17</td>
<td>81.55</td>
<td>89.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFF</td>
<td>37.17</td>
<td>37.17</td>
<td>81.55</td>
<td>89.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPT</td>
<td>81.55</td>
<td>81.55</td>
<td>81.55</td>
<td>81.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPT</td>
<td>89.80</td>
<td>89.80</td>
<td>89.80</td>
<td>89.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Percentage of Best Unique Heuristic Solutions**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>0.23</td>
<td>0.10</td>
<td>0.00</td>
<td>4.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFF</td>
<td>0.23</td>
<td>0.10</td>
<td>0.00</td>
<td>4.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPT</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Percentage of Instances Solved Optimally**

- **MM**
- **EFF**

**Due Date as a Portion of the Total Processing Time**

- **MM**
- **EFF**

---

New Challenges in Scheduling – Aussois – March 30, 2016
List Algorithms

small instances $n \leq 20$

large instances $n \leq 150$
List Algorithms

![Graph showing competitive ratio and percentage of instances with EFF solution not worse than other solutions.](image-url)
Conclusions

- offline scheduling
  - P2|d_j=d|Y is binary NP-hard
  - P|d_j=d|Y is unary NP-hard

- online scheduling
  - online algorithm with finite competitive ratio for P|d_j=d, online over list|Y
  - optimal for P2|d_j=d, online over list|Y


- simple list algorithms are very efficient
- most instances are trivial
Future research

- Formulating dominance relations for two-machine case
- Constructing approximation algorithms for offline case
- Extending theoretical results for the problem with a given number of machines
- Studying other scheduling problems with late work criterion
Late Work Scheduling in Online and Offline Mode

Małgorzata Sterna
Xin Chen, Jacek Błażewicz, Xin Han
Kateryna Czerniachowska

Thank you for your attention!