Triangle Scheduling

Christoph Dürr, Zdeněk Hanzálek, Christian Konrad, Yasmina Seddik, René Sitters, Óscar Carlos Vásquez, Gerhard Woeginger,
March 2nd 2016, seminar S
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_j$
- equal processing time $x$
- decide starting times for jobs prior to knowledge of $x$
- job $j$ is removed from schedule if $x > p_j$
- minimize makespan
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_j$
- equal processing time $x$
- decide starting times for jobs prior to knowledge of $x$
- job $j$ is removed from schedule if $x > p_j$
- minimize makespan
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_i$
- Equal processing time $x$
- Decide starting times for jobs prior to knowledge of $x$
- Job $j$ is removed from schedule if $x>p_i$
- Minimize makespan
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_j$
- equal processing time $x$
- decide starting times for jobs prior to knowledge of $x$
- job $j$ is removed from schedule if $x > p_j$
- minimize makespan
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_i$
- equal processing time $x$
- decide starting times for jobs prior to knowledge of $x$
- job $j$ is removed from schedule if $x > p_i$
- minimize makespan

Diagram:

- Jobs: $x$, $x$, $x$
- Schedules: $S_1$, $S_2$, $S_3$, $S_4$
A scheduling problem

- Single machine
- n jobs, with priorities $p_j$
- equal processing time $x$
- decide starting times for jobs prior to knowledge of $x$
- job $j$ is removed from schedule if $x > p_j$
- minimize makespan
A scheduling problem

- Single machine
- $n$ jobs, with priorities $p_j$
- Equal processing time $x$
- Decide starting times for jobs prior to knowledge of $x$
- Job $j$ is removed from schedule if $x > p_j$
- Minimize makespan
A geometric problem

- Single machine
- $n$ triangles, with sizes $p_i$
- decide starting times $S_j \geq 0$ for job
- such that $|S_i - S_j| \geq \min\{p_i, p_j\}$
- minimize makespan $\max S_j + p_j$
- place triangles on the time line without overlapping
motivation  Mixed criticality scheduling

## Results

<table>
<thead>
<tr>
<th>complexity ?</th>
<th>unary NP-hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>where is the barrier ?</td>
<td>binary tree ratio polynomial if $\leq 2$ NP-hard if $&gt;2$</td>
</tr>
<tr>
<td>approximation algorithm ?</td>
<td>Greedy is a 1.5 approximation</td>
</tr>
<tr>
<td>ratio tight ?</td>
<td>Greedy’s ratio $\geq 1.05$</td>
</tr>
<tr>
<td>APX-hard ?</td>
<td>No, there is a QPTAS</td>
</tr>
</tbody>
</table>
Greedy

• Process jobs in order $p_1 \geq \ldots \geq p_n$

• Place job $j$ in gap of maximum size $s$, right shift jobs following gap by $2p_j - s$ if $2p_j > s$
A lower bound for OPT

- assign every gap \([S_i, S_j]\) to smallest among jobs \(i, j\)
- For every job \(j\) let \(a_j \in \{0, 1, 2\}\) be the number of assigned gaps
- Property: \(\sum a_j = n\) (number of jobs)
- Property: gap size \(\leq\) assigned job size
- Lower bound: \(\sum a_j p_j \leq OPT\) for any \(a \in \{0, 1, 2\}^n\) with \(\sum a_j = n\)
- Hence (\(n\) even): 2 times the smallest half of \(\{p_1, \ldots, p_n\}\) \(\leq\) OPT
Greedy is a 1.5 approximation

- Wlog suppose no job can be shifted to the right
- Truncate job sizes from \( p \) to \( p' \), \( p_j = \text{size of gap starting at } S_j \)
- Let \( A \) be total \( p' \)-sizes of larger half of jobs and \( B \) of smaller half
- Makespan produced by Greedy is \( A + B \)
- Wlog suppose insertion of job \( n \) increased makespan
- Hence all gaps have sizes less than \( 2p_n \), hence \( A < 2B \)
- But \( \text{OPT} \geq 2B \)
- Makespan = \( A + B \leq 3B \leq \frac{2}{3} \text{OPT}(p') \leq \frac{2}{3} \text{OPT}(p) \)
NP-hardness

reduction from 3-dimensional numerical matching

- given $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n, D$
  partition into triplets $(a_i, b_j, c_k)$ with $a_i + b_j + c_k = D$

- generate $5n$ triangles
  $M$ is some arbitrary constant

  - $E$ (size $8M + 5D$)
  - $F$ (size $4M$)
  - $A_i$ (size $2M + 2a_i + D$)
  - $B_j$ (size $2M + b_j$)
  - $C_k$ (size $M + c_k + D$)
Binary tree ratio

- Suppose order $p_1 \geq \ldots \geq p_n$

- **Formally** ratio is $\max (p_{\text{ceil}(i/2)} / p_i)$

- NP-hardness proof generates instances with binary tree ratio $> 2$ (arbitrarily close)

- Greedy is optimal on instances with binary tree ratio $\leq 2$.

- **Informally** it is the maximum ratio between vertex and successor if jobs are placed in row order on this tree
Greedy is optimal when binary tree ratio \( \leq 2 \).

- We construct weights \( a_j \in \{0, 1, 2\} \) such that \( \sum a_j p_j \) is the makespan and the lower bound for \( \text{OPT} \).
Thank you for your attention,
danke schön,
Děkuji, Merci,
Gracias, dank je