SCHEDULING WITH EXPLORABLE UNCERTAINTY

C. Dürr (Sorbonne University, CNRS)  
Thomas Erlebach (Leicester)  
Nicole Megow (Bremen)  
Julie Meißner (Berlin)

aussois scheduling workshop 2018
**INTRODUCTION**

- **s-t cut**
  - minimum
  - graph
    - planarity test
      - yes: Dijkstra
      - no: Dinic

- **send file**
  - send
  - gzip

- **formally**
  - job $j$?
    - $u_j$ known
      - test
        - $p_j \in [0, u_j]$  
        - formally
          - $1$ reveals $p_j$

  - obj: $= \sum C_j$
  - goal: $= \min$ regret: $= \text{alg/opt}$
  - (competitive ratio)

- **warmup: single job**
  - Algorithm
    - job $j$?
      - $u_j$ known
        - test
          - $p_j = u_j$
          - $1$ reveals $p_j$

  - Worst optimum
    - $p_j = 0$

  - Ratio
    - $u_j$
    - $(1 + u_j)/u_j$

  - Worst input
    - $u_j = \varphi = 1.618$
  - Best algorithm
    - test iff $u_j \geq \varphi$
## RESULTS

<table>
<thead>
<tr>
<th>competitive ratio</th>
<th>lower bound</th>
<th>upper bound</th>
<th>algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>determinstic ratio</td>
<td>1.8546</td>
<td>2</td>
<td>THRESHOLD</td>
</tr>
<tr>
<td>randomized ratio</td>
<td>1.6257</td>
<td>1.7453</td>
<td>RANDOM</td>
</tr>
<tr>
<td>det. ratio. when ∀j: ui=p</td>
<td>1.8546</td>
<td>1.9338</td>
<td>BEAT</td>
</tr>
<tr>
<td>det. ratio. when ∀j: ui=p, pj∈{0,p}</td>
<td>1.8546</td>
<td>1.8668</td>
<td>UTE</td>
</tr>
</tbody>
</table>

**Algorithm Diagram:**

```
ALG: u1 1 1 p3 1 1 p2
p4=p5=0
```

**More Jobs Diagram:**

```
defer
```

**More Jobs Text:**

`more jobs`
### Deterministic Lower Bound

<table>
<thead>
<tr>
<th>competitive ratio</th>
<th>lower bound</th>
<th>upper bound</th>
<th>algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>determinstic ratio</td>
<td>1.8546</td>
<td>2</td>
<td>THRESHOLD</td>
</tr>
<tr>
<td>randomized ratio</td>
<td>1.6257</td>
<td>1.7453 (asymptotic ratio)</td>
<td>RANDOM</td>
</tr>
<tr>
<td>det. ratio. when (\forall j: u_j = p)</td>
<td>1.8546</td>
<td>1.9338</td>
<td>BEAT</td>
</tr>
<tr>
<td>det. ratio. when (\forall j: u_j = p, p_j \in {0, p})</td>
<td>1.8546</td>
<td>1.8668</td>
<td>UTE</td>
</tr>
</tbody>
</table>

- n uniform jobs with upper limit \(p\)
- Index jobs in order they are touched by algorithm (tested or executed untested)
- \(p_j = 0\) if \(j \geq \delta n\) or job \(j\) is executed untested by algo.
- \(p_j = p\) otherwise
- Algorithm gets even to know \(\delta\)
- Any decent algorithm produces a schedule with above structure for parameters \(\nu, \lambda\) with \(\nu + \lambda \leq \delta\)
- The competitive ratio is \(\frac{\text{ALG}(\delta, \nu, \lambda, n)}{\text{OPT}(\delta, \nu, n)}\)
- Algorithm (minimizer) chooses \(\nu, \lambda\)
- Adversary (maximizer) chooses \(n, \delta\)
- Analyzing local optima yields ratio 1.854628

**ALG:**

```
| p | p | p | 1 | p | 1 | p | 1 | p | 1 | 1 | 1 | 1 | 1 | p | p | p |
```

**OPT:**

```
| 1 | 1 | 1 | 1 | 1 | p | p | p | p | p | p | p | p |
```

\(\nu\) \hspace{1cm} \lambda \hspace{1cm} \delta - \nu - \lambda \hspace{1cm} 1 - \delta \hspace{1cm} \delta - \nu - \lambda \hspace{1cm} \delta - \nu - \lambda \hspace{1cm} \delta - \nu - \lambda \hspace{1cm} \delta - \nu - \lambda \hspace{1cm} 1 + \nu - \delta \hspace{1cm} \delta - \nu\)
### ALGORITHM THRESHOLD

<table>
<thead>
<tr>
<th>competitive ratio</th>
<th>lower bound</th>
<th>upper bound</th>
<th>algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterministic ratio</td>
<td>1.8546</td>
<td>2</td>
<td>THRESHOLD</td>
</tr>
<tr>
<td>randomized ratio</td>
<td>1.6257</td>
<td>1.7453 (asymptotic ratio)</td>
<td>RANDOM</td>
</tr>
<tr>
<td>det. ratio. when ∀j: u_j=p</td>
<td>1.8546</td>
<td>1.9338</td>
<td>BEAT</td>
</tr>
<tr>
<td>det. ratio. when ∀j: u_j=p, p_j∈{0,p}</td>
<td>1.8546</td>
<td>1.8668</td>
<td>UTE</td>
</tr>
</tbody>
</table>

- Execute untested all jobs $j$ with $u_j \leq 2$ in order...
- Test all other jobs in arbitrary order. If $p_j \leq 2$, execute, otherwise defer.
- Execute all deferred jobs in order...
- Worst case instance: 
  - a jobs $u_j=2, p_j=0$
  - b jobs $u_j=p_j=2$
  - c jobs $u_j=p_j=2+\epsilon$
- Simple arithmetics: 
  - $\text{ALG}(a,b,c) \leq 2 \cdot \text{OPT}(a,b,c)$

**Example:**

**ALG:**

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2+\epsilon</th>
<th>2+\epsilon</th>
<th>2+\epsilon</th>
</tr>
</thead>
</table>

| c | b | a | c |

**OPT:**

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2+\epsilon</th>
<th>2+\epsilon</th>
<th>2+\epsilon</th>
</tr>
</thead>
</table>

| a | b | c |


In this paper we have introduced an adversarial model of scheduling with testing where a competitive ratio against an oblivious adversary.

Remark.

If \( \gamma \geq \beta \), then all jobs are executed without testing. Execute right after their test the first \( \max\{0, \beta\} \) fraction of jobs. Then only if \( p_j = 0 \). Finally execute deferred jobs.

**Worst case instance** defined by length \( p \) fraction \( \gamma \): the first \( \gamma n \) tested jobs have \( p_j = p \) and the remaining \( p_j = 0 \).

**Second order analysis** to optimize \( p, \gamma \) and \( \beta \)

\[
\begin{align*}
\text{ALG} & \quad \text{if } \gamma \leq 1 - \beta: \\
1 & \quad p & \quad 1 & \quad p & \quad 1 & \quad p & \quad 1 & \quad 1 & \quad 1 & \quad 1 & \quad p & \quad p & \quad p \\
\beta & \quad 1 - \beta - \gamma & \quad \gamma & \quad 1 - \beta - \gamma \\
\text{ALG} & \quad \text{if } \gamma \geq 1 - \beta: \\
1 & \quad p & \quad 1 & \quad p & \quad 1 & \quad p & \quad 1 & \quad 1 & \quad p & \quad 1 & \quad p & \quad 1 & \quad 1 \\
1 - \gamma & \quad \gamma & \quad \gamma \\
\text{OPT:} & \quad 1 & \quad 1 & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p & \quad p \\
\gamma & \quad 1 - \gamma & \quad 1 - \gamma
\end{align*}
\]
**Algorithm Random**

<table>
<thead>
<tr>
<th>Competitive Ratio</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic Ratio</td>
<td>1.8546</td>
<td>2</td>
<td>Threshold</td>
</tr>
<tr>
<td>Randomized Ratio</td>
<td>1.6257</td>
<td>1.7453 (asymptotic ratio)</td>
<td>Random</td>
</tr>
<tr>
<td>Det. Ratio. when ( \forall j: u_j = p )</td>
<td>1.8546</td>
<td>1.9338</td>
<td>Beat</td>
</tr>
<tr>
<td>Det. Ratio. when ( \forall j: u_j = p, p_j \in {0, p} )</td>
<td>1.8546</td>
<td>1.8668</td>
<td>UTE</td>
</tr>
</tbody>
</table>

- Has parameters \( T \geq E \)
- Schedule untested all jobs with upper limit \( < T \) in increasing upper limit order.
- Test in random order all larger jobs \( j \), if \( p_j \leq E \) execute immediately, else defer their execution.
- Finally schedule deferred jobs in increasing processing time order.

- Worst case instances:
  - \((1-\alpha-\beta-\gamma)\) fraction of jobs: \( u_j = T, p_j = 0 \)
  - \( \alpha n \) jobs have \( u_j = T, p_j = T \)
  - \( \beta n \) jobs have \( u_j = E, p_j = E \)
  - \( \gamma n \) jobs have \( u_j = E+\epsilon, p_j = E+\epsilon \)

- Ratio \( \leq T \) iff
  \[ G := OPT \cdot T - ALG \geq 0 \]

- Algorithm chooses \( T, E \) to max. \( G \)
- Adversary chooses \( \alpha, \beta, \gamma \) to min. \( G \)
Every decision we make, is the wrong one.

-Murphy