Scheduling with a Processing Time Oracle

**Context**

The Problem

- **Single machine**

  - \( n \) jobs, known parameters \( p, x > 0 \)
  - Each job is either short: processing time \( p \)
    - or long: processing time \( p + x \)

Objective:

- Sum of completion times

Processing time is hidden to the machine but can be revealed for a given job by calling a processing time oracle (such as best tests 1 time unit). AWS

Example

2 Extreme Strategies

- Executing in order of increasing processing time is optimal.
- Not testing any job: worst case is order of decreasing processing times.
- Testing all jobs: permits optimal order but test caused delay.

**Between these extremes, what is the best strategy?**

**performance is measured by the Competitive Ratio**

\[ \text{ALG} \geq \frac{\text{OPT}}{\text{ALG}} \]

*Figure 1: Some schedules with four jobs A, B, C, D.*
Our model differs from

1. Job lengths drawn from hidden distribution and job weights, test permits to learn these parameters

2. Test = different meaning

\[ \text{execute untested} \rightarrow \text{proc.time} = \hat{p}_i \]
\[ \text{execute tested} \rightarrow \text{proc.time} \in [0, \hat{p}_j] \]

[Simplification, Dominance]

1. Dominant behavior:
   - Tested short jobs should be executed immediately
   - Tested long jobs should be postponed towards the end

2. Algorithm processes jobs in arbitrary order:
   - Each job is either tested (T) or executed untested (E)

3. Strategy of algorithm \( e \) for \( E \)
   - Strategy of adversary \( e \) for \( T \)

4. Conjecture:
   - Optimal algorithm consists in two phases:
     - Test some number of jobs
     - Execute remaining jobs untested
Our contributions

performance measure = competitive ratio = \( \max \frac{\text{cost of alg.}}{\text{optimal cost}} \)

doesn’t need to test

input: n, p, x

non adaptive setting we can compute optimal strategy in time \( O(n^2) \)

adaptive setting we can compute optimal \textbf{two phase} strategy in time \( O(n^3) \)

open problem are all optimal strategies two phase?

---

**Non Adaptive Algorithm**

*Example n=2, p=2, x=1*

<table>
<thead>
<tr>
<th></th>
<th>EE</th>
<th>ET</th>
<th>TE</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>pp</td>
<td>( \frac{5p}{3} \times 1 = \frac{5p}{3} )</td>
<td>( \frac{3p+1}{3p} = \frac{4}{3} )</td>
<td>( \frac{3p+2}{3p} = \frac{1}{3} )</td>
<td>( \frac{3p+3}{3p} = 2 )</td>
</tr>
<tr>
<td>px</td>
<td>( \frac{3p+x}{3p+x} = 1 )</td>
<td>( \frac{3p+x+1}{3p+x} = \frac{9}{7} )</td>
<td>( \frac{3p+x+2}{3p+x} = \frac{9}{7} )</td>
<td>( \frac{3p+x+3}{3p+x} = 10 )</td>
</tr>
<tr>
<td>ex</td>
<td>( \frac{3p+2x}{3p+2x} = \frac{11}{9} )</td>
<td>( \frac{3p+2x+1}{3p+2x} = \frac{11}{9} )</td>
<td>( \frac{3p+2x+2}{3p+2x} = \frac{11}{9} )</td>
<td>( \frac{3p+2x+3}{3p+2x} = \frac{11}{9} )</td>
</tr>
<tr>
<td>xx</td>
<td>( \frac{3p+3x}{3p+3x} = 1 )</td>
<td>( \frac{3p+3x+1}{3p+3x} = \frac{16}{15} )</td>
<td>( \frac{3p+3x+2}{3p+3x} = \frac{17}{15} )</td>
<td>( \frac{3p+3x+3}{3p+3x} = \frac{19}{15} )</td>
</tr>
</tbody>
</table>

Strategies lead to schedules of the form

\((Tx)^x (Tx)^x (Ex)^x (Eo)^x\)

\(O(n^3)\) possible schedules

but using 2nd order analysis we can compute the equilibrium schedule and hence also the optimal strategy for the algorithm

in time \( O(n^2) \)

We can compute the asymptotic competitive ratio, it is

\[
\begin{align*}
    & \left\{ \begin{array}{ll}
        \sqrt{1 + \frac{x}{p}} & \text{if } 0 \leq x < 2 + \frac{1}{p} \\
        1 + \frac{x^2 - px - 4 + \Delta}{2px^2} & \text{if } x \geq 2 + \frac{1}{p}
    \end{array} \right.
\end{align*}
\]

for \( \Delta = 8p(x-1)x^2 + (1 + px - x^2)^2 \)
Adaptive Algorithm

Compute Optimal Strategy

We represent interaction between algorithm and adversary by a walk on a grid.
- Tp: step down
- Tx: step right
- Switch to execution phase: stop walk

\[ \text{Stop Ratio}(c, d, e) = \text{Stop Ratio}(c, d, e) \]

Cost generated by test

Adversary chooses path
Algorithm chooses where to stop

\[ d \]

\[ C \]

Compute optimal adversarial strategy
Iteratively:
- \( P^* \): currently best path
- \( R^* \): minimal stop ratio on \( P^* \)
- \( P \): boundary of marked cells
- \( (c, d) \): cell of min. stop ratio \( R \)
- \( R^* \): ratio of marked cell \( R \) and its left and below
- If \( R > R^* \), then update \( P^* \) and \( R^* \)
Experiments \( n = 6 \)

number of tests in the equilibrium schedule

\[
\begin{array}{c}
\text{adaptive} \\
\text{non-adaptive}
\end{array}
\]

Does not provide insight to show the conjecture

Regions not convex

price of non-adaptivity

not monotone in \( p \) nor in \( x \)

Conclusion

- natural next steps: randomized algorithm
  nonuniform processing time domain \([x, y]\), testing time \(t_j\)

- develop this general framework of optimization under explorable uncertainty

THANK YOU FOR YOUR ATTENTION