Online Scheduling Problems in the Random Order Model
Benedikt Maximilian Plank, supervised by Prof. Dr. Nicole Megow

Technische Universität München – Fakultät für Mathematik
Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

Abstract

In combinatorial optimization, the term online refers to any problem setting where decisions have to be made based on incomplete information. The random order model, in which the behaviour of algorithms for online problems is analysed in expectation over a randomly chosen arrival order of the input sequence, is one of the methods suggested in the literature to move beyond classical worst case analysis and its various drawbacks. We apply the random order model to makespan minimization in the online restricted assignment problem and show that no randomised algorithm can have a random order competitive ratio better than $\Omega(\log \log m)$ where $m$ is the number of machines.

Online

**Definition:** Competitive Ratio (Adversarial Order)

$$\gamma_{\text{opt}}(\sigma) = \frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)}$$

\(\sigma\): set of all input instances

For certain problems, this ratio is unbounded when instances get larger. In this case, an asymptotic version of the definition can be used in connection with Landau notation.

**Drawback:** Worst case analysis often does not explain behaviour in practice. Worst case instances usually are artificial and rely on a specific arrival order.

Scheduling Problems

In general: “Deciding, when and where a number of jobs/tasks is processed on a set of available machines/servers”

Here: Makespan minimization under restricted assignment

**Problem:** $m$ jobs with specified “length” $p_i$, arriving one by one; each job must be placed on one of a given subset of $m$ machines.

**Objective:** Minimizing the makespan ("total length") of the schedule.

Example:

```
[ 1, 3 ]   [ 1, 2, 3 ]
1 2 3
```

Input

```
Schedule computed by ALG
```

Further Steps

- Lower bound should also hold for randomised algorithms.
- Algorithm might use the first sub-instances for training and would then be able to handle a sub-instance arriving in original order better.

These difficulties can be overcome by an elaborate randomised construction of the sub-instances. For the formal proof we refer to the Bachelor’s thesis [3].

Random Order Model

**Definition:** Competitive Ratio (Random Order)

$$\gamma_{\text{opt}}(\sigma) = \frac{\text{ALG}(\tilde{\pi}(\sigma))}{\text{OPT}(\tilde{\pi}(\sigma))}\mathbb{E}_{\tilde{\pi}}$$

\(\tilde{\pi}\): expectation w.r.t. permutation $\tilde{\pi}$ chosen uniformly at random

**Note:** This is equivalent to averaging the performance over all possible arrival orders.

Random Order Competitive Ratio

**Idea** to overcome drawback of adversarial order competitive ratio: draw arrival order of instances uniformly at random from all permutations

**Definition:** Competitive Ratio (Random Order)

$$\gamma_{\text{opt}}(\sigma) = \frac{\text{ALG}(\tilde{\pi}(\sigma))}{\text{OPT}(\tilde{\pi}(\sigma))}\mathbb{E}_{\tilde{\pi}}$$

\(\tilde{\pi}\): expectation w.r.t. permutation $\tilde{\pi}$ chosen uniformly at random

**A Tight Result for Adversarial Order**

**Theorem:** Upper Bound [2]

- Placing each job on the currently least loaded feasible machine is $\lceil \log m \rceil + 1$-competitive.

**Proof/Construction:**

- W.l.o.g. ALG places each job on the machine with lower index. Otherwise, relabel machines IDs.
- Placing each job on the machine with higher index yields an optimal makespan of 1.

Further Steps

- In the proof, the following non-trivial technical issues arise:
  - Lower bound should also hold for randomised algorithms.
  - Algorithm might use the first sub-instances for "training" and would then be able to handle a sub-instance arriving in original order better.

These difficulties can be overcome by an elaborate randomised construction of the sub-instances. For the formal proof we refer to the Bachelor’s thesis [3].

Extending the Lower Bound to Random Order

**Consider** the problem of online makespan minimization under restricted assignment.

**Main Challenge:** Construction of the adversarial order lower bound relies on the jobs arriving one by one; each job must be placed on one of a given subset of $m$ machines. The construction works analogously for other values of $m$.

**Theorem:** Lower Bound [1]

- $\gamma_{\text{opt}}(\sigma) = \Omega(\log \log m)$ for any possibly randomised algorithm $\text{ALG}$ where $m$ is the number of machines in an instance.

**Proof/Construction:**

- $\gamma_{\text{opt}}(\sigma) = \Omega(\log \log m)$

Further Steps

- In the proof, the following non-trivial technical issues arise:
  - Lower bound should also hold for randomised algorithms.
  - Algorithm might use the first sub-instances for "training" and would then be able to handle a sub-instance arriving in original order better.

These difficulties can be overcome by an elaborate randomised construction of the sub-instances. For the formal proof we refer to the Bachelor’s thesis [3].

Summary

Currently known upper and lower bounds for online makespan minimization with restricted assignment – the new result is marked in orange:

<table>
<thead>
<tr>
<th></th>
<th>Adversarial Order</th>
<th>Random Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitive Ratio</td>
<td>$\lceil \log m \rceil + 1$</td>
<td>$\lceil \log \log m \rceil + 1$</td>
</tr>
</tbody>
</table>

References

This is joint work with Jannik Matsuschke, Nicole Megow, Thomas Kesselheim, Kevin Schewior and Andreas Tönnis.

