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

TopMath Mathematik mit Promotion

Technische Universität München · Elitenetzwerk Bayern · Universität Augsburg University

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

"online" = "without full information"

In the online version of an optimization problem, instances arrive "piece by piece". An algorithm has to decide how to treat each piece right away. In general, these solutions are not optimal.

Typically, the performance of an online algorithm ALG is analysed by its worst case behaviour compared to an optimal solution OPT:

Definition: Competitive Ratio (Adversarial Order)

$$\mathsf{cr}_{ALG} := \sup_{\sigma\in\Gamma} \frac{\mathsf{ALG}(\sigma)}{\mathsf{OPT}(\sigma)}$$

Γ: 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

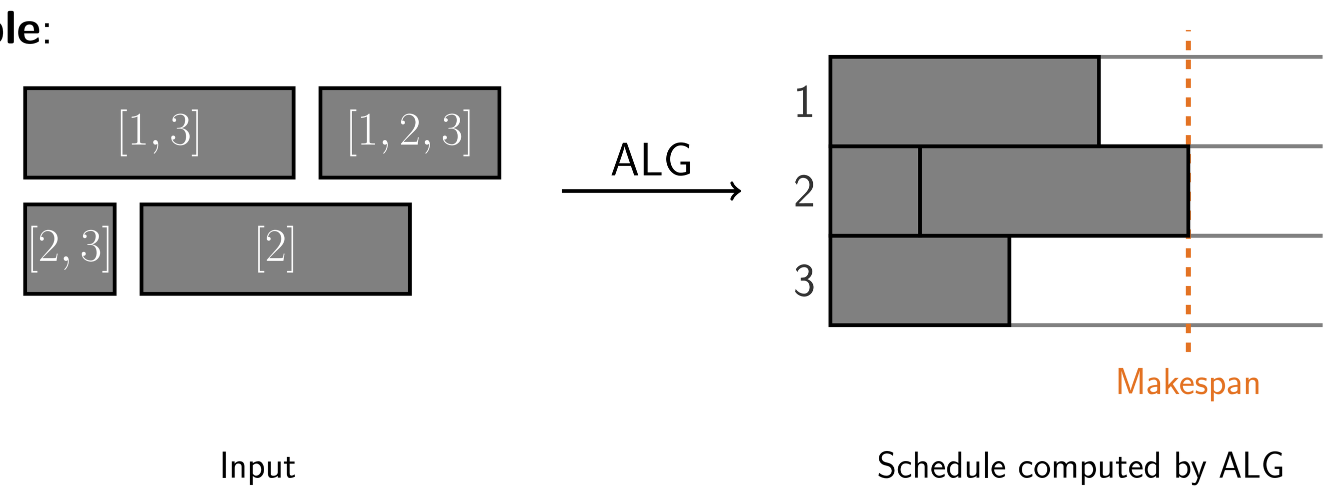
In general: *"Deciding, when and where a number of jobs/tasks is processed on a set of available machines/servers"*

Here: *Makespan minimization under restricted assignment*

Problem: n jobs with specified "length" p_j 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:



Random Order Model

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)

$$\mathsf{rocr}_{\mathsf{ALG}} := \sup_{\sigma\in\Gamma} \frac{\mathbb{E}_\pi[\mathsf{ALG}(\pi(\sigma))]}{\mathsf{OPT}(\sigma)}$$

Γ: set of all input instances

$\mathbb{E}_\pi$: expectation w.r.t. permutation π chosen uniformly at random

Note: this is equivalent to averaging the performance over all possible arrival orders

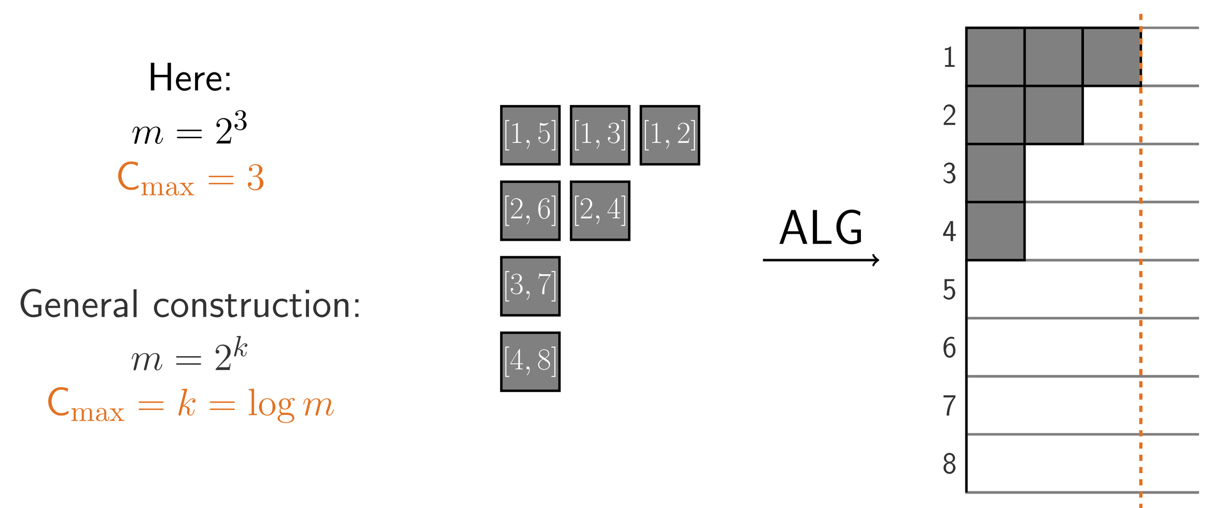
A Tight Result for Adversarial Order

Consider the problem of online makespan minimization under restricted assignment.

Theorem: Lower Bound [1]

$\mathsf{cr}_{\mathsf{ALG}} = \Omega(\log m)$ for any possibly randomised algorithm ALG where m is the number of machines in an instance.

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.

The construction works analogously for other values of m.

Theorem: Upper Bound [2]

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

Extending the Lower Bound to Random Order

Consider the problem of online makespan minimization under restricted assignment.

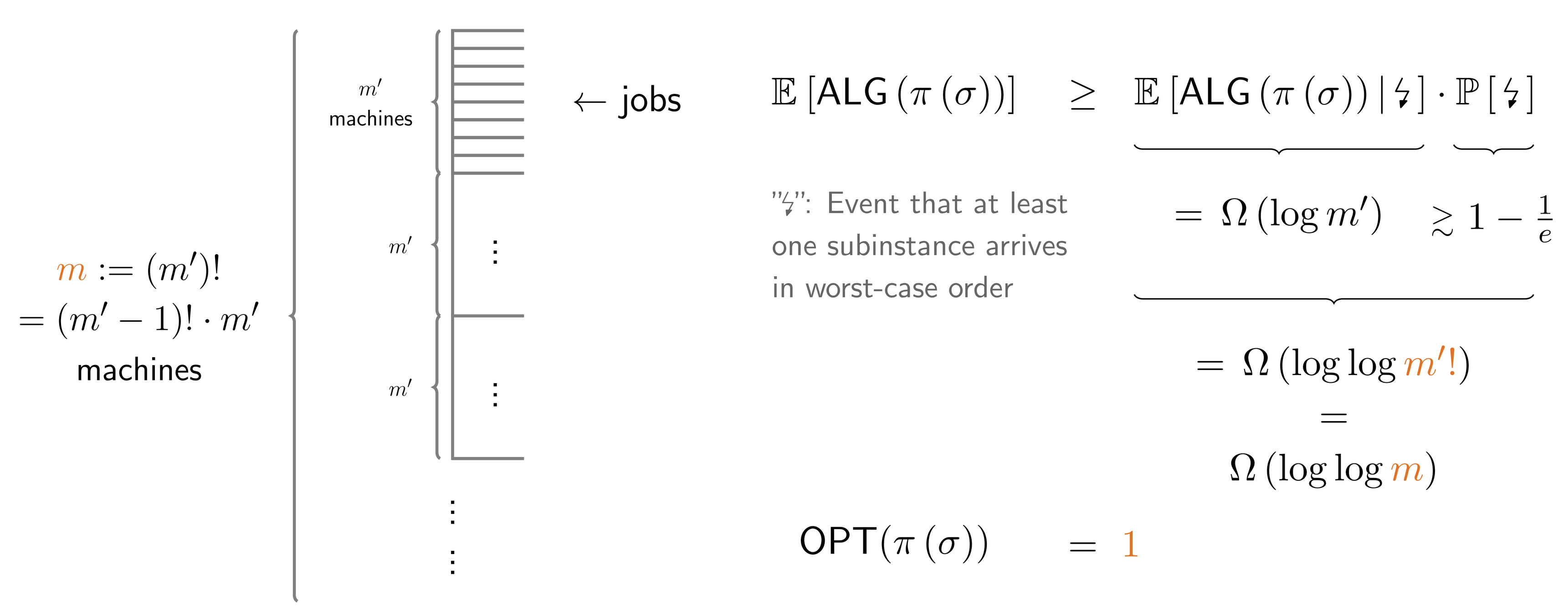
Main Challenge Construction of the adversarial order lower bound relies on the jobs arriving in a specific order! This cannot be guaranteed in the random order model.

Main Idea Concatenate multiple instances to a large instance. This increases the probability of at least one sub-instance arriving in desired order. A permutation of the full instance induces independent permutations of the sub-instances. Constant probability of at least one sub-instance arriving in original order allows to apply the adversarial order construction.

Theorem: Lower Bound (Random Order) [3]

$\mathsf{rocr}_{\mathsf{ALG}} = \Omega(\log\log m)$ for any possibly randomised algorithm ALG where m is the number of machines in an instance.

Proof/Construction



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].

References

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

[1] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, On-line load balancing with applications to machine scheduling and virtual circuit routing, in STOC 1993, pp. 623–631.

[2] Y. Azar, J. S. Naor, and R. Rom, The competitiveness of on-line assignments, in SODA 1992, pp. 203–210.

[3] B. M. Plank, Online Scheduling Problems in the Random Order Model, Bachelor's Thesis, Technische Universität München, 2017

Summary

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

	Adversarial Order Competitive Ratio	Random Order Competitive Ratio
Lower Bound	$\Omega(\log m)$	$\Omega(\log\log m)$
Upper Bound	$\lceil\log m\rceil + 1$	$\lceil\log m\rceil + 1$