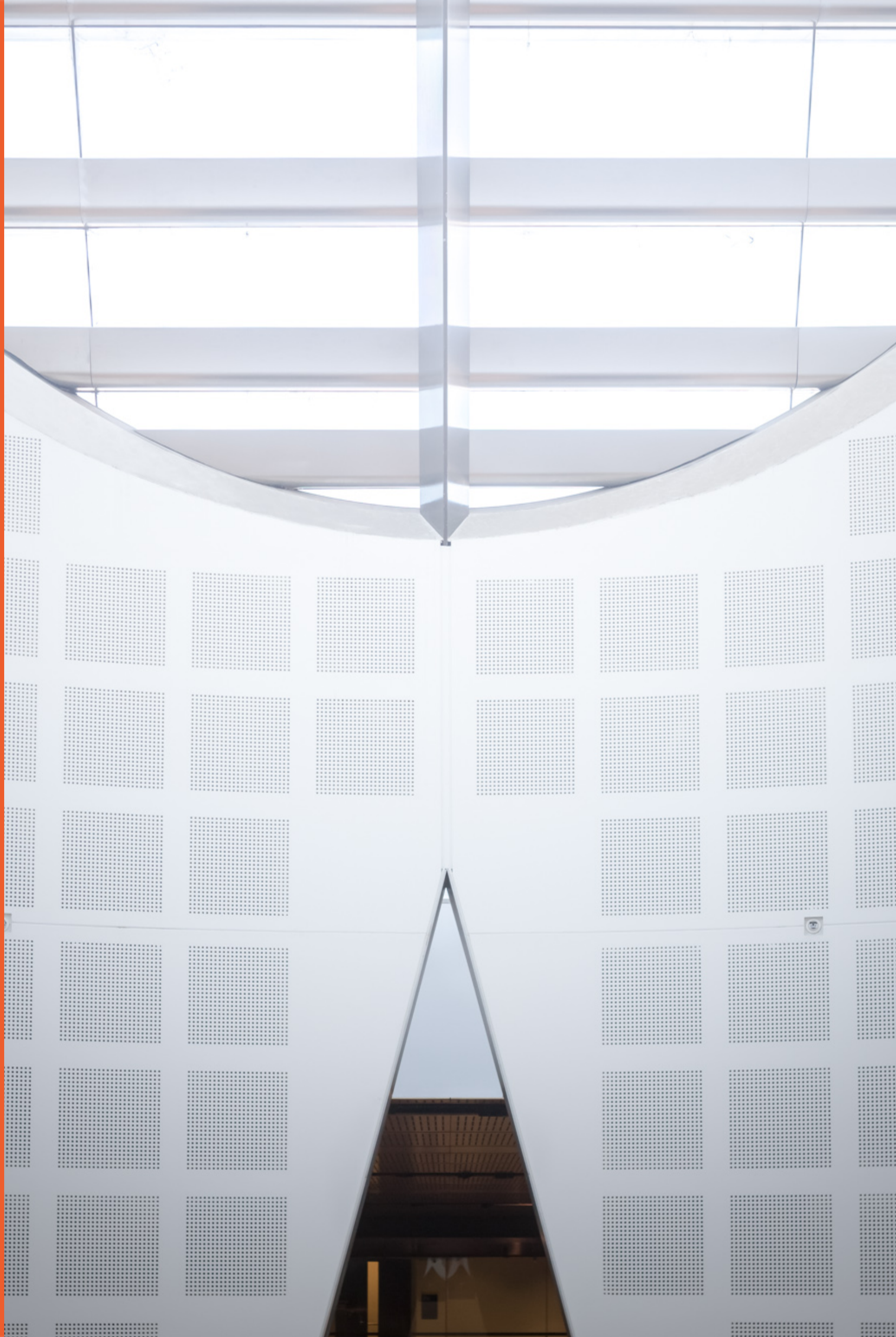
Competitive Fair Scheduling with Predictions

Presented by
Tianming Zhao

Authors:
Tianming Zhao, Chunqiu Xia, Xiaomin Chang,
Chunhao Li, Wei Li, Albert Y. Zomaya

School of Computer Science

THE UNIVERSITY OF SYDNEY



Fair Scheduling with Predictions

Background

Algorithms with predictions (aka learning-augmented algorithms) leverage possibly erroneous predictions to improve decision-making beyond worst-case guarantees.

Problem

- Single machine non-clairvoyant scheduling to minimize max-stretch.
  - $S_j = \frac{C_j - r_j}{p_j^*}$ (stretch of job J_j)
  - C_j (completion time), r_j (release time), p_j^* (job size, i.e., processing time)
  - max-stretch $= \max_{1 \le j \le n} S_j = \max_{1 \le j \le n} \frac{C_j - r_j}{p_j^*}$
- Max-stretch is a fairness measure.
- Job sizes p_j^* are unknown when the job arrives, but a prediction p_j is known. The predictions might not be accurate, and we use $\eta = \max_{1 \le j \le n} \eta_j = \max_{1 \le j \le n} \max\{p_j^*/p_j, p_j/p_j^*\}$ to represent the total prediction error.

Question: How *to perform fair scheduling with imperfect information?*

Key Results

Improved fairness in scheduling

We propose a family of algorithms for different prediction models. In particular, *Relaxed-Greedy (RG)* achieves an $O(\eta^3\sqrt{P})$ competitive ratio, where *P* is the maximum job size ratio.

Consistency-smoothness trade-off

We show how consistency and smoothness trade-off can be achieved by introducing a parametrized algorithm called RG^x.

Enhanced robustness via resource augmentation

The algorithms we initially have suffered poor robustness. We propose to use resource augmentation to address this issue.

Algorithm Overview

Overview

- Key idea: Reduce the problem to two-job-size case (short vs. long jobs) using predictions. If there are only two job sizes, the jobs of the same type should be processed in a FIFO manner.
- Classify jobs based on job size predictions: job J_j short if $p_j \leq \sqrt{p_{\min} p_{\max}}/\mu$, else long.
- Approximate the stretch using $(C_j - r_j)/\sqrt{p_{\min} p_{\max}}$ for a short job and $(C_j - r_j)/p_{\max}$ for a long job; this is called the *relaxed stretch*.
- Greedily schedule the jobs by always running the one with a minimal *relaxed stretch*.

Analysis

- Classifying the jobs into short and long overestimates the job size by $\eta^2\sqrt{P}$, but the problem is reduced to the two-job-size case.
- The greedy heuristic minimizes the relaxed stretch within a constant factor.
- The optimal relaxed stretch is bounded by η times the optimal max-stretch.
- Therefore, the max-stretch bound of our algorithm is $O(\eta^3\sqrt{P})$ times the optimum.

Consistency-Smoothness Trade-offs

- How can we design algorithms that adapt to different levels of prediction confidence?
- Given a user-defined parameter *x*, use $\alpha = p_{\min}^{x} p_{\max}^{1-x} / \sqrt{3}$ to classify long/short jobs.
- The resulting algorithm RG^x has a competitive ratio of $O(\eta^{2+2x} P^{1-x})$.
- At $x = 0$, RG^0 simplifies to *FIFO* (*P*-competitive), achieving the worst consistency $O(P)$ but optimal smoothness $O(1)$. At $x = 1/2$, $\mathsf{RG}^{1/2}$ is the *Relaxed-Greedy* with the best consistency $O(\sqrt{P})$ but the worst smoothness $O(\eta^3)$.
- RG^x interpolates *FIFO* and *Relaxed-Greedy*.

Bounding Robustness via Resource Augmentation

- All algorithms mentioned suffer poor robustness.
- We propose to use resource augmentation to address this issue.
- Suppose you have a $(1+\epsilon)$-speed machine. Run *RG* in the unit-speed and *Round-Robin (RR)* in ϵ-speed. This yields a $(1+\epsilon)$-speed $O(\min\{\eta^3\sqrt{P}, n/\epsilon\})$-competitive algorithm named *RR-augmented RG*.
- *RR*-augmented *RG* achieves the asymptotically optimal robustness.

Thank you

Contact me: tzha2101@gmail.com (Tianming Zhao)