Uniform Machine Scheduling with Predictions

Tianming Zhao, Wei Li, Albert Y. Zomaya

School of Computer Science, The University of Sydney

tzha2101@uni.sydney.edu.au, weiwilson.li@sydney.edu.au, albert.zomaya@sydney.edu.au

THE UNIVERSITY OF SYDNEY

Abstract

The recent revival in learning theory has provided us with improved capabilities for accurate predictions. This work contributes to an emerging research agenda of online scheduling with predictions by studying the makespan minimization in uniformly related machine non-clairvoyant scheduling with job size predictions. Our task is to design online algorithms that effectively use predictions and have performance guarantees with varying prediction quality. We first propose a simple algorithm-independent prediction error measurement to quantify prediction quality. To effectively use the predicted job sizes, we design an offline improved 2-relaxed decision procedure approximating the optimal schedule. With this decision procedure, we propose an online $O(\min\{\log\eta, \log m\})$-competitive algorithm that assumes a known prediction error. Finally, we extend this algorithm to construct a robust $O(\min\{\log\eta, \log m\})$-competitive algorithm that does not assume a known error. Both algorithms require only moderate predictions to improve the well-known $\Omega(\log m)$ lower bound, showing the potential of using predictions in managing uncertainty.

Problem Definition

Problem Description:

- There are m uniformly-related parallel machines and n independent jobs.
- Jobs have varying sizes, and machines have varying speeds.
- Jobs are dependency-free, preemptive-restart, and ready at time 0.
- Assign jobs to the machines.

Objective:

- Minimize makespan $C_{\max}$, the time of the last job completes.

Constraint:

- The job size is only known after the job is completed (non-clairvoyant).

The Graham notation of the problem is $Qm \mid online-time-nclv, pmtn-restart \mid C_{\max}$.

ML Oracle and Prediction Error

We assume an ML oracle predicting job sizes is accessible to the algorithm. Let p_j^*, p_j denote the job size and job size prediction for Job J_j, $1 \le j \le n$, respectively. Define the total prediction error η as the max multiplication gap between job size predictions and job sizes:

$$\eta = \max_{1\le j\le n} \eta_j = \max_{1\le j\le n} \max\{\frac{p_j^*}{p_j}, \frac{p_j}{p_j^*}\}$$

Performance Evaluation

We evaluate the performance of the algorithms by the competitive framework and the metrics of consistency and robustness.

Competitive Framework:

An online algorithm A will compare against an optimal offline algorithm A^*. We analyze the competitive ratio, the ratio of the makespans produced by A and A^*.

Consistency:

The competitive ratio under perfect predictions, i.e., under $\eta = 1$.

Robustness:

The competitive ratio under any predictions, i.e., under any $\eta \ge 1$.

Existing Results

- The problem is NP-complete in the strong sense.
- There exists a 2-relaxed decision procedure for the offline version.
- The problem has an $\Omega(\log m)$ lower bound on any deterministic algorithm.
- There exists an $O(\log m)$-competitive algorithm matching the $\Omega(\log m)$ bound (state-of-the-art).

Our Contributions

We present the first learning-augmented algorithm that does not need the knowledge of the prediction error η and achieves the asymptotically optimal consistency, robustness, and a good competitive ratio on η, improving the theoretical bound.

Algorithm Design Process

An improved 2-relaxed procedure:

We present a decision procedure that decides if a valid schedule exists given inputs.

Scheduling with known η:

Given job size predictions and η, we present an $O(\min\{\log\eta, \log m\})$-competitive algorithm via the doubling technique with the improved 2-relaxed procedure.

Scheduling with unknown η :

Building on the previous scheduling algorithm with known η, we present an algorithm achieving the same performance bound but without the knowledge of η.

An Improved 2-relaxed Procedure

Algorithm Overview:

The procedure takes a set of jobs with their sizes and a deadline d. Either it produces a schedule of length at most $2d$, or it confirms that no d-length schedule exists.

High-level Idea:

When a machine is idle, process the largest non-running job that can be completed on the machine by time d. Stop the process at time $2d$.

Improvement:

The improvement (over the existing one) is that it uses slow machines to process jobs even if they can not be completed by the deadline. This approach procedures valid $2d$-length schedule for more problem instances.

Scheduling with Known η

Algorithm Overview:

With job size predictions and the total prediction error η, the algorithm produces a schedule of makespan at most $O(\min\{\log\eta, \log m\}) \cdot C^*_{\max}$, where $C^*_{\max}$ denotes the optimal makespan.

High-level Idea:

Use the ratio of job size prediction and η as an (under-)estimate of the job size. Run the decision procedure repeatedly (doubling technique) with the job size and makespan estimates.

Performance Bound:

- Running the decision procedure incurs $O(1) \cdot C^*_{\max}$ on the makespan (each time).
- There are $O(\min\{\log\eta, \log m\})$ calls to the decision procedure.
- The overall makespan is bounded by $O(\min\{\log\eta, \log m\}) \cdot C^*_{\max}$.

Scheduling with Unknown η

Algorithm Overview:

With job size predictions only, the algorithm produces a schedule of makespan at most $O(\min\{\log\eta, \log m\}) \cdot C^*_{\max}$, where $C^*_{\max}$ denotes the optimal makespan.

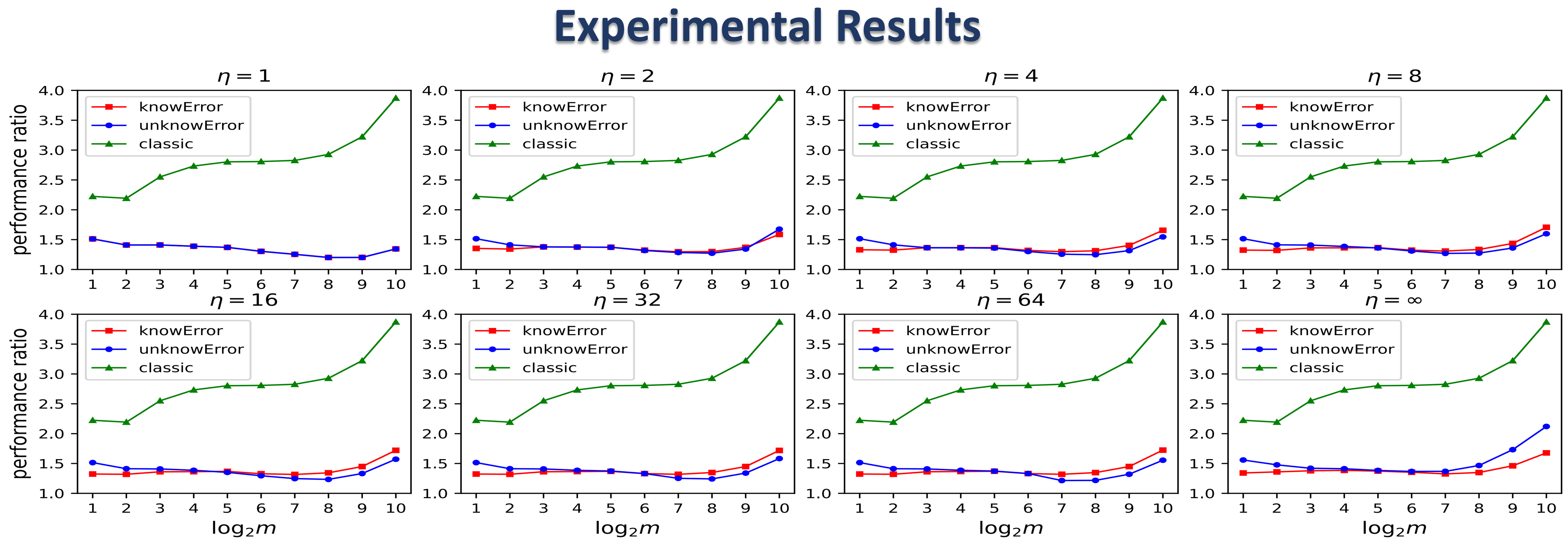
High-level Idea:

View the scheduling with known η as a "decision procedure". Estimate η online via doubling technique: run scheduling with known η, and force the algorithm to stop when detecting η is underestimated. If η is too large, switch to the classic $O(\log m)$-competitive algorithm.

Performance Bound:

- The makespan is bounded by $O(\log m) \cdot C^*_{\max}$ when η is unbounded due to switching.
- There are $O(\log\eta)$ calls to scheduling with known η when η is relatively small.
- Running scheduling with known η costs either $O(1) \cdot C^*_{\max}$ if it is forced to stop or $O(\min\{\log\eta, \log m\}) \cdot C^*_{\max}$ if it succeeds.
- The overall makespan is bounded by $O(\min\{\log\eta, \log m\}) \cdot C^*_{\max}$.

Experimental Results



Our algorithms consistently outperform state-of-the-art even under arbitrarily bad predictions. The performance ratio increases sublinearly as $\log m$ increases, verifying the theoretical results. Both algorithms stay close to the offline optimum in all instances.

References

- Purohit, M.; Svitkina, Z.; and Kumar, R. 2018. Improving Online Algorithms via ML Predictions. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Garey, M. R.; and Johnson, D. S. 2009. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman and Company.
- Shmoys, D. B.; Wein, J.; and Williamson, D. P. 1995. Scheduling Parallel Machines On-Line. *SIAM Journal on Computing*, 24(6): 1313–1331.

Acknowledgements

- Dr. Wei Li acknowledges the support of the Australian Research Council (ARC) through the Discovery Early Career Researcher Award (DE210100263).
- Professor Zomaya and Dr. Wei Li acknowledge the support of an ARC Discovery Project (DP200103494).