

Scheduling with Costly Job Size Predictions

Ahan Mishra
Cornell University
abm247@cornell.edu

ABSTRACT

An increasing amount of research has gone into studying scheduling algorithms with access to predictions of job sizes in the broader area of learning-augmented algorithms [3]. However, these predictions themselves have costs in real-world applications, and research accounting for prediction costs is limited. We provide, to our knowledge, the first competitive analysis of scheduling algorithms accounting for prediction costs.

1. INTRODUCTION

We study the setting of single-machine scheduling in which pre-emption is allowed, and the objective is to minimize total flow time. As a simplifying assumption, we assume that the predictions received are exact.

1.1 Background

An early work on using predictions in scheduling is [1] which accounts for possible prediction error and quantifies the “price of misprediction” but does not explicitly consider prediction cost as we do. Designing and analyzing algorithms which dynamically decide whether to request predictions is noted as an open question in [2]. The closest work to ours is [6] which accounts for prediction costs by allowing both cheap 1-bit predictions and more expensive predictions. However, that work introduces a more complicated setting of prediction costs and does not provide a competitive analysis. It also separates analysis of two prediction cost models, the *server time cost model* and *external cost model*. In the server time cost model, a prediction occurs on the same server processing the jobs and takes 1 unit of time, leading to a higher flow time objective. In the external cost model, a prediction occurs on an external server directly adds one unit to the objective [6].

1.2 Contributions

We contribute (to our knowledge, the first) results to competitive analysis for both the server time and external cost models. We show that the initial problem has no good algorithm, motivating an assumption on cheap prediction costs. We then provide a super-constant lower bound in the server cost model and constant upper and lower bounds in the external cost model.

2. RESULTS

Formalizing the setting described in Section 1, there are n jobs with arrival times a_i . If an algorithm $\mathcal{A}$ completes the jobs at times $c_1, c_2, \dots, c_n$, then the total flow time achieved by algorithm $\mathcal{A}$ is $F = \sum_{i=1}^n (c_i - a_i)$. The competitive ratio of $\mathcal{A}$ is F/F_{OPT} , where F_{OPT} is the total flow time of the optimal clairvoyant algorithm. This is the algorithm that always processes the job with the shortest remaining processing time (SRPT) [5]. If F/F_{OPT} is not upper-bounded by any constant, $\mathcal{A}$ is not constant-competitive.

PROPOSITION 1. *Without additional assumptions, no deterministic or randomized algorithm is constant-competitive in either the server time cost model or the external cost model.*

It is known that if predictions are not allowed, then the competitive ratio of any deterministic algorithm is $\Omega(n^{1/3})$ and that of any randomized algorithm is $\Omega(\log(n))$ due to Motwani et al. [4]. The job sizes in the construction can be shrunk so that the optimal objective is much less than 1. Then, even one prediction leads to unsustainable cost, reducing to the fully non-clairvoyant case. Proposition 1 and the relevant proof idea involving extremely small job sizes motivate the following assumption.

ASSUMPTION 2. *Let j_i be the job size of job i . Then, $\sum_{i=1}^n j_i \geq n$.*

This assumption should be interpreted as requiring the average prediction cost to be at most the average job size. We find that with Assumption 2, the existence of constant-competitive algorithms depends, perhaps surprisingly, on whether prediction costs are accounted for by the server time cost model or the external cost model. The external cost model admits an algorithmic result, whereas we present hardness results for the server time cost model.

THEOREM 3. *Under the server time cost model with Assumption 2, no deterministic or randomized algorithm is constant-competitive on total flow time. In particular, any deterministic algorithm has a competitive ratio of $\Omega(n^{1/5})$ and any randomized algorithm has a competitive ratio of $\Omega(\log(n))$.*

The key technical challenge in this proof is to construct an adversary that handles prediction queries while still “fooling” the algorithm. We find a way to do this in the deterministic case (see Figure 1), essentially hiding job sizes by immediately ending jobs upon prediction requests and then restarting a new job to take its place.

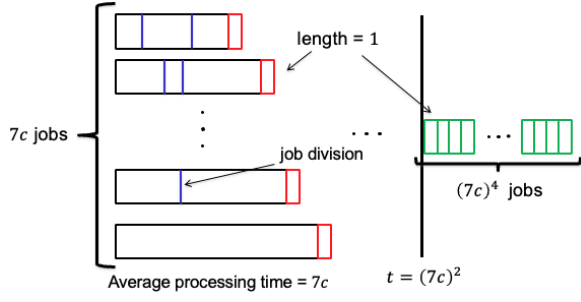


Figure 1: Modification of the adversary strategy from Motwani et al. demonstrating there is no c -competitive deterministic algorithm for this problem. This reduction works for any c , showing there is no constant-competitive algorithm. A “job division” occurs at each time a prediction is completed, with a 1-time-unit delay. For example, if a job is predicted at $t = 4$, it is declared to have 1 more unit of processing time and another job is released at $t = 5$.

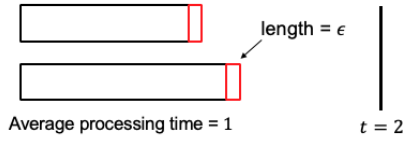


Figure 2: Adversary strategy for a lower bound of $4/3$ among deterministic algorithms. Two jobs are released at $t = 0$ and are declared to have ϵ more units of processing time at $t = 2$ (if their size is not predicted). This can be done for any $\epsilon > 0$. This all assumes that there is no prediction, but if there is any prediction, then the total cost must be at least 4 (compared to the optimal of at most 3).

Algorithm 1 AllPredict

```

1: Initialize an empty priority queue  $Q$ 
2: while system is running do
3:   if a new job  $j$  arrives then
4:     Query the prediction oracle  $O$  to get the job time:
        $p_j \leftarrow O(j)$ 
5:     Insert  $j$  into  $Q$  with priority  $p_j$ 
6:   end if
       Run the SRPT Algorithm [5] using queue  $Q$ .
7: end while

```

THEOREM 4. *Under the external cost model with Assumption 2, Algorithm 1 is 2-competitive on total flow time. No deterministic algorithm has a competitive ratio strictly better than $\frac{4}{3}$. No randomized algorithm has a competitive ratio strictly better than $\frac{9}{8}$.*

The lower bounds of Theorem 4 can be shown using adversary strategies with only 2 jobs. That for deterministic algorithms is presented in Figure 2. The proof for the randomized case involves one job of size $\frac{2}{3}$ and another of size $\frac{4}{3}$. We expect better lower bounds can be achieved with more sophisticated examples. For the proof of 2-competitiveness, the objective achieved by Algorithm 2 is $F_{\text{OPT}} + n$, where F_{OPT} is the total flow time for SRPT on the instance. Furthermore, because the average job size is at least 1 (see Assumption 2), $F_{\text{OPT}} \geq n$, so we have $\frac{F_{\text{OPT}} + n}{F_{\text{OPT}}} \leq 2$, as desired.

3. FUTURE WORK

A significant simplifying assumption made was that predictions are exact. Similar questions could be asked in cases where predictions are associated with some error, perhaps with a parameter for the total absolute additive error. This would enable analysis of traditional properties relevant to learning-augmented algorithms such as robustness and consistency.

Another goal of future work is to design and analyze algorithms for this problem on real-world instances and datasets.

4. REFERENCES

- [1] M. Mitzenmacher. Scheduling with Predictions and the Price of Misprediction. In T. Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [2] M. Mitzenmacher and R. Shahout. Queueing, predictions, and llms: Challenges and open problems, 2025.
- [3] M. Mitzenmacher and S. Vassilvitskii. Algorithms with predictions. *Commun. ACM*, 65(7):33–35, June 2022.
- [4] R. Motwani, S. Phillips, and E. Torng. Nonclairvoyant scheduling. *Theoretical Computer Science*, 130(1):17–47, 1994.
- [5] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.
- [6] R. Shahout and M. Mitzenmacher. Skippredict: When to invest in predictions for scheduling. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 105663–105701. Curran Associates, Inc., 2024.