Fair Algorithm Design

Yuexing Niu, Agnes Totschnig, Supervisor: Prof. Adrian Vetta

1. Introduction

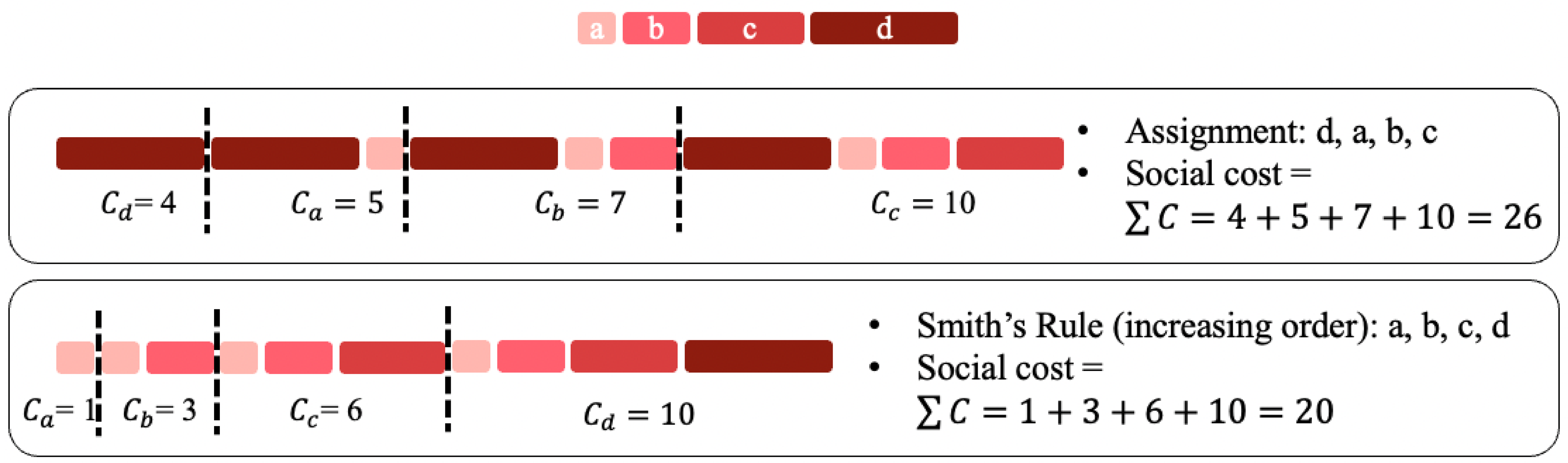
The problem of algorithmic bias has become more and more prominent. Instead of providing impartial and unbiased decisions, current technologies tend to exacerbate the societal problems of inequality. AI court judges, selective advertisement and school admission algorithms are just a few examples of these controversial algorithms that treat people unfairly. Our work focuses on the problem of fair scheduling.

Keywords: Fair Algorithms, Job Scheduling, Multi-Agent Systems, Algorithmic Game Theory, Price of Fairness.

2. The Model

- The model consists of n agents, each wanting to schedule a single task on a shared machine.
- The individual cost c_i of agent i is defined as the completion time of its single task.
- The social cost is defined as the sum of individual costs over all agents: $S = \sum_{\text{all } i} c_i$
- The schedule minimizing the social cost is given by **Smith's Rule**, which schedules the tasks in the order of increasing duration.

Example: Given 4 jobs with processing time: $a < b < c < d$, where $a = 1, b = 2, c = 3, d = 4$.



This example illustrates the intuition behind Smith's rule: every agent has to wait while the first task is being processed, so its duration will be counted n times in the social cost. Hence, it is better to put the shorter tasks first.

3. Fairness

Definition 1 *Let r_i be the expected completion time for an individual task under uniform random assignment. Then the individual fairness is defined by $f_i = \frac{c_i}{r_i}$, where c_i is the completion time under the considered schedule.*

- The overall fairness of a schedule is defined to be the maximum of the individual fairness over all agents.
- The optimal fairness is given by the random assignment, which has value 1. The fairness of the algorithm should be close to 1.

4. The Algorithm

The algorithm partitions the sorted jobs into priority groups, some of which follow Smith's Rule and others assigned from a uniform random distribution.

Lemma 2 *The worst fairness is given by the smallest job in the last priority group.*

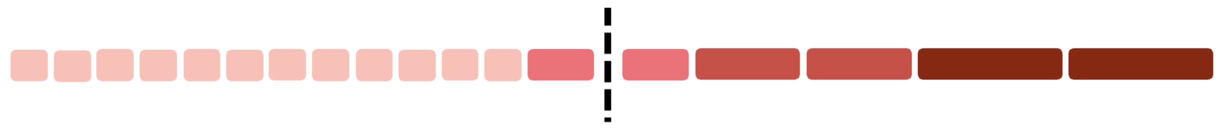
So it suffices to partition the set of jobs into two priority groups: since only the last group affects the overall fairness, all prior groups can be merged into a meta-group following Smith's rule to optimize the social cost.

Lemma 3 *The smallest job in the second section should be at least as large as the biggest job in the first section.*

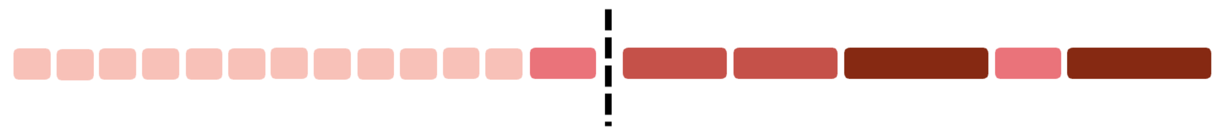
```
FairScheduling (set of n jobs, ϵ):
```

- Sort the jobs in increasing processing time.

- Partition the jobs into two groups, where the siye of the first priority group is ϵ.

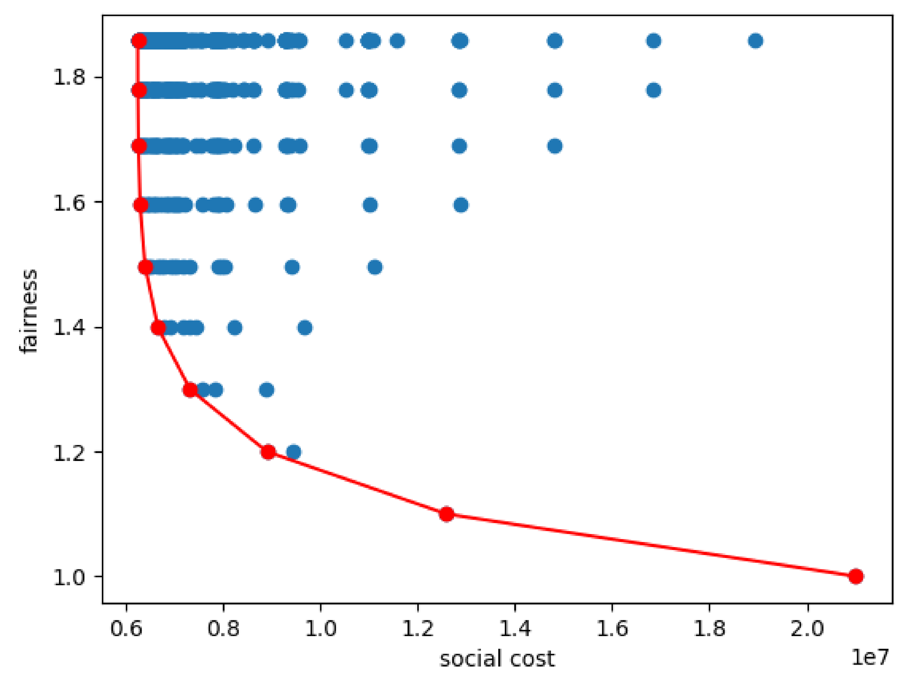


- Keep the first priority group according to Smith's Rule and reorder the second group according to a uniform random distribution.



5. Pareto Optimality

The algorithm induces a class of Pareto optimal schedules: for these algorithms, fairness and social cost cannot be simultaneously improved. The graphic below shows their performance on the Pareto frontier in red.



A simple swapping argument shows that having a task in the first section with longer processing time than some task in the second section violates Pareto optimality.

6. Results

It remains to decide on the right trade-off between fairness and social cost by choosing an appropriate value for ϵ, which measures the proportion of the first priority group.

Theorem 4 *The algorithm yields a fairness of $1 + \epsilon$, i.e. each task finishes within a factor of $(1 + \epsilon)$ of their expected completion time in the uniform random assignment.*

As ϵ tends to zero, the schedule resembles the fully random assignment with optimal fairness of 1, but the social cost increases:

Theorem 5 *The social cost of the algorithm is within a factor of $\frac{1}{2}(1 + \frac{1}{\epsilon})$ of the social cost of Smith's rule.*

This bound is tight and essentially cannot be improved, as the matching worst case example is given by instances with only one large task and many smaller tasks of unitary size.

7. Future Research

Our future work will focus on applying these concepts to the generalized model with multiple machines. We are hoping to combine our current approach with a PTAS to achieve similar performance guarantees. Moreover, we will study the applications of our results to related areas of algorithmic game theory, such as network routing, and other classical problems of applied computer science, such as task scheduling in processors and operating systems.

8. References

[1] Dimitris Bertsimas, Vivek Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59:17–31, 02 2011.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

9. Contact Information

Yuexing Niu
`yuexing.niu@mail.mcgill.ca`

Agnes Totschnig
`agnes.totschnig@mail.mcgill.ca`