Dynamic Fair Division

15-300, Fall 2017

David Zeng

1 Project Website

The website for this project is davidzeng-x.github.io

2 Project Description

My faculty advisor for this project will be Professor Ariel Procaccia. In addition, I will be mentored by his postdoc Alex Psomas.

Fair division studies how we can allocate finite resources to multiple agents while optimizing some metric of fairness. For dynamic fair division, we consider the case when these agents can arrive and depart, but the goal is still to allocate the resource at each step.

One model for dynamic fair division assumes that the resource is homogeneous, and can be thought of the interval $[0, 1]$. A resource is homogenous if the value an agent gets from a section of the interval is proportionate to the size of the section. For example, any agent receiving the subinterval $[0, 1/2]$ would value it at $1/2$. For simplicity, the model assumes that at each step t, exactly one new agent arrives. The agent is then allocated some portion of the resource. An allocation for t agents is called σ_t-fair if each agent values their section at least $\frac{\sigma}{t}$. Finally, an allocation algorithm is called σ-fair if it guarantees that for any step t, the allocation will be σ_t-fair. The goal is to maximize σ, also known as the fairness ratio.

In order to ensure algorithms are actually dynamic and don't just find a new optimal allocation for each time step t, the model restricts reallocation. In particular, an algorithm is d-disruptive if at each time step t, it disrupts (lowers the allocation) of at most d agents. In addition, to study algorithms that disrupt less than one agent per arrival, we also consider algorithms that take in a control vector as a parameter, which specifies at each time step i, that at most d_i disruptions are allowed. A natural question in this area is "How do restrictions on the number of disruptions affect the optimal fairness ratio?"

The above model has been studied by Alex Psomas, who proved the optimal fairness ratio for different versions of the above model.

My project will be to investigate some variations on the above model. The first proposed variation is to instead set a fixed number of disruptions that can be spent at any time step. Formally, the goal is to prove optimal fairness ratios when given parameters d, n, which specify that d disruptions can be spent over n steps. A second proposed variation is to prove various fairness metrics when instead of a single divisible resource, there are many indivisible items. For this variation, it might be useful to investigate other fairness metrics such as envy or proportionality. Because not all variations will be approachable, one challenge is to find the models that seem the most promising.

Dynamic fair division is a fairly new research area. However, dynamic fair division models some important real world resources. For example, it could model a shared computing resource where it is impractical to constantly reallocate resources between users. This commonly occurs in shared data centers and cloud computing. In addition, dynamic fair division is related to problems in fields such as scheduling, so results and techniques found here could be useful in other fields.

3 Project Goals

100% Goal: I would like to investigate both proposed variations. For at least one variation, I hope to find a nontrivial allocation algorithm where I can prove some properties about its performance with regard to some metric (such as fairness ratio or envy-freeness). Another way to satisfy this goal would be to prove a nontrivial lower bound for the performance of any allocation algorithm on some metric (such as fairness ratio).

75% Goal: If I'm unable make substantial progress for the general version of the two variations, I will try to prove something for a restricted version of the problem. I will likely be investigating restricted versions anyway on my way to reaching the 100% goal.

125% Goal: The goal here will be to improve on the algorithm from the 100% goal. One way to improve it would be to find an algorithm that is provably optimal with regard to some metric.

4 Milestones

End of Semester: The first milestone is to do the necessary background reading for the project. This includes reading Alex's paper in detail and focusing on the techniques used in the mathematical sections. In addition, I will identify and read related papers that will be useful to this project.

January 31st: Investigate the fixed disruptions variation. Gain intuition by playing with small examples and running simulations. Prove lower bounds or find allocation algorithms for restricted versions of the fixed disruptions variation.

February 14th: Make progress on the fixed disruptions variation. Regardless of results, start thinking about the individual items variation by coming up with some specific models, especially focusing on choosing a good fairness metric to work with. In order to do this, I will learn more about other fairness metrics like envy-freeness, proportionality and figure out what happens when they get applied to our problem.

February 28th: Gain some intuition on the individual items variation by working with small examples, or trying to find allocation algorithms for restricted versions of the problem. Try to improve on any allocation algorithm I've found already.

March 21st: If I have made substantial progress with the initial models chosen or if seem stuck, I will look into some new variations, guided by my better understanding of the variations I have worked on. Possible variations include assuming the individual items have generalized utility functions or applying different restrictions on disruptions.

April 4th: For whatever algorithms I have come up with, if I have not already done so yet, investigate the opposite direction by investigating provable lower bounds for what allocation algorithms can achieve.

April 18th: Start to formalize my results. Focus on resolving any mathematical details in my proofs that I may have overlooked when working on the proof at a high level. I can also try to prove some related problems. For example, proving a particular problem is actually NP-hard would be interesting.

May 2nd: Finish formalizing the results I've gotten and write up the results for the paper and presentation.

5 Literature Search

The main paper that will be useful to this project is the paper authored by Alex Psomas [1]. In addition, it will be useful to look through the citations of that paper, as well as other papers on dynamic fair division.

6 Resources Needed

Since the project falls under computer science theory, it is unlikely I will need any special resources for this project. It is possible it will be useful to use a computational or programming approach to gain insight into the problem, in which case it will be useful to have programming related software.

References

[1] E. Friedman, Christos-Alexandros Psomas, and Shai Vardi. Controlled dynamic fair division. In *EC*, 2017. 5