Dynamic Weighted Fairness with Minimal Disruptions

Sungjin Im
University of California, Merced, USA
sim3@ucmerced.edu

Kamesh Munagala
Duke University, USA
kamesh@cs.duke.edu

Benjamin Moseley
Carnegie Mellon University, USA
moseleyb@andrew.cmu.edu

Kirk Pruhs
University of Pittsburgh, USA
kirk@cs.pitt.edu

ABSTRACT

In this paper, we consider the following dynamic fair allocation problem: Given a sequence of job arrivals and departures, the goal is to maintain an approximately fair allocation of the resource against a target fair allocation policy, while minimizing the total number of disruptions, which is the number of times the allocation of any job is changed. We consider a rich class of fair allocation policies that significantly generalize those considered in previous work.

We first consider the models where jobs only arrive, or jobs only depart. We present tight upper and lower bounds for the number of disruptions required to maintain a constant approximate fair allocation every time step. In particular, for the canonical case where jobs have weights and the resource allocation is proportional to the job’s weight, we show that maintaining a constant approximate fair allocation requires $\Theta(\log^* n)$ disruptions per job, almost matching the bounds in prior work for the unit weight case. For the more general setting where the allocation policy only decreases the allocation to a job when new jobs arrive, we show that maintaining a constant approximate fair allocation requires $\Theta(\log n)$ disruptions per job. We then consider the model where jobs can both arrive and depart. We first show strong lower bounds on the number of disruptions required to maintain constant approximate fairness for arbitrary instances. In contrast we then show that there is an algorithm that can maintain constant approximate fairness with $O(1)$ expected disruptions per job if the weights of the jobs are independent of the jobs arrival and departure order. We finally show how our results can be extended to the setting with multiple resources.

1 INTRODUCTION

The formal study of fair resource allocation has advanced rapidly in recent years, motivated by applications to computer systems.

The basic theory of fair resource allocation has its roots in Economics and in scheduling results in computer science. However, modern applications such as data center scheduling have motivated considering new desiderata in fair resource allocation.

In this paper, we consider a dynamic model for resource allocation, a topic that has received significant attention in recent literature. In this model, which is again motivated by computing systems, each of $n$ jobs (or agents) may potentially arrive or depart from the system, so at every time step $t$ we are presented with a set of alive jobs $N^t \subseteq N$. The set $N^t \setminus N^{t-1}$ is the set of jobs that arrive at time $t$, and the set $N^{t-1} \setminus N^t$ is the set that departs at time $t$. We have a single divisible resource.

There is some underlying fair share policy $I(j, t)$ which specifies the ideal fair share of the resource for job $j$ at time $t$. At every time step $t$, an allocation policy/algorithm $A$ must determine $A(j, t)$, its allocation of the resource to a job $j \in N^t$. The policy $A$ must be online in that it can not rely on knowledge of the future. Ideally one would like $A$ to be perfectly fair, that is it is always the case that $A(j, t) = I(j, t)$. However, a perfectly fair allocation policy would generally lead to a disruption, which is a change in the resource allocation of a job, of every job when any job arrives or departs (which is exactly when a job’s fair share changes in most natural fair share policies). These disruptions can have significant overheads as they involve reassigning resources and changing the job states. Due to the overhead, limiting the number of disruptions is a key design factor to most systems. Therefore, we investigate the minimum number of disruptions required to achieve approximate fairness.

Definition 1. For $c \geq 1$, an allocation policy $A$ is $c$-approximate if it always guarantees that $A(j, t) \geq I(j, t)/c$.

1.1 Weighted Fairness

Previous work [1–3] considered the case of uniform fairness, where $I(j, t) = \frac{1}{|N|^t}$. However, there are many situations where the appropriate notion of fairness is something other than a uniform sharing of the resource(s). One natural/common example is weighted fairness. In this setting each job $j$ has weight $w_j$. Weights typically correspond to priorities that could be based on criteria such as willingness to pay for the resource, importance of the job, and so on. Furthermore, weights also arise naturally in fair allocation contexts where there are multiple resources that could be complements or substitutes, and the utility (or rate) of a job is a function of the resources of each type allocated to the job.

In weighted fair share policies, a job’s fair share is proportional to its weight, that is, $I(j, t) = \frac{w_j}{\sum_{k \in N^t} w_k}$. Uniform fairness is a special case of weighted fairness, where the weight of every job is 1.
2 OUR RESULTS

In this paper, we study these natural questions: (1) What is the optimal bound on the number of disruptions per job for $O(1)$-approximate allocation policies with weighted fairness?; and (2) What is the optimal bound on the number of disruptions per job for $O(1)$-approximate allocation with more general fair share policies?

We answer the above questions by presenting tight results in increasingly complex models of fairness. Our main (and surprising) result is that in the model where jobs only arrive, it is indeed possible to achieve constant approximation to fairness with nearly constant number of disruptions per job. When jobs can both arrive and depart, we show that to achieve constant approximate fairness an algorithm will have to disrupt a large number of jobs per arrival/departure for some instances. In contrast we show that there is an algorithm that can maintain constant approximate expected fairness with $O(1)$ expected disruptions per job if the weights of the jobs are independent of the jobs arrival and departure order.

2.1 Weighted Fairness with Only Arrivals

We first consider weighted fairness in the arrival-only model, where $N^t \subseteq N^T$ for all times $t$. The same results will apply to the symmetric departure-only model where $N^T \subseteq N^t - 1$ for all times $t$. (Imagine maintaining a fair allocation of some resource among a batch of jobs as jobs finish and depart.) We show the following:

Theorem 3. Consider weighted fair share policies in the arrival only model. There is an $O(1)$-approximate allocation policy that will cause at most $O(\log^* n)$ disruptions for each job, where $n$ is the total number of arriving jobs. Further, this result is tight.

Our allocation policy groups jobs into groups with exponentially increasing weights, and then treats each group as a single job. It then applies a monotone transform to the weight of each job, and uses this transformed weight instead of the original weight to perform the weighted fair allocation. The transformation must both (a) be sufficiently invariant to keep the number of disruptions low; and (b) sufficiently faithful to the original weight of the jobs to achieve $O(1)$-approximation. In fact, it is a priori not even clear that such a transform even exists, and showing its existence is one of our primary technical contributions.

2.2 Arrivals and Departures

We next consider the case where jobs can both arrive and depart. We show that it is no longer possible to always achieve both $O(1)$-approximation and a near linear number of disruptions.

Theorem 4. Consider weighted fair share policies with both job arrivals and departures. For every $c$-approximate deterministic algorithm $A$, there is an instance that causes $A$ to make $\Omega(n^{1+1/(4c+1)})$ disruptions.

In contrast, we prove a positive result if job weights are independent of the jobs arrival and departure order.

Theorem 5. Consider weighted fair share policies with both job arrivals and departures. Assume that the weights $w_1, \ldots, w_n$ of the jobs are arbitrary, but the assignment of these weights to the $n$ jobs is uniformly random. In this setting there is an $4$-approximate randomized algorithm $A$, for which the expected number of disruptions per arrival and per departure is at most $5$.

2.3 Monotone Fairness

One natural property that many/most fair share policies have is monotonicity, that is, the arrival of a job can not increase another job’s fair share, and the departure of a job can not decrease another job’s fair share. More formally:

Definition 1. A fair resource share policy $I$ is monotone if it satisfies the following conditions: Suppose job $j$ arrives at time $t$, then $I(j', t) \leq I(j', t - 1)$ for every $j' \in N^t \setminus \{j\}$. Similarly, if job $j$ departs at time $t$, then $I(j', t) \geq I(j', t - 1)$ for every $j' \in N^t \setminus \{j\}$.

We show that while more disruptions may be needed to approximate fairness for an arbitrary monotone fairness policy than for weighted fair share policies, it is still possible to achieve an almost linear number of disruptions.

Theorem 6. Consider general monotone share policies in the arrival-only model. There is an $O(1)$-approximate deterministic algorithm $A$ such that the number of disruptions per job is $O(n \log n)$ and this bound is tight.

2.4 Multidimensional Resources

Our analysis easily extend to some canonical settings where there are $D$ divisible resources each with unit supply, and the rate of a job is a function of the resources allocated to it.

In Cobb-Douglas utilities, job $j$ has a substitutability vector $\alpha_{jd}, d \in [D]$ with $\sum_{d=1}^{D} \alpha_{jd} = 1$. Given allocation $x_{jd}$ in dimension $d$, the rate of execution is: $y_j = \prod_{d=1}^{D} x_{jd}^{\alpha_{jd}}$. A fair share policy maximizes the product of the rates. In weighted Dominant Resource Fairness (weighted DRF), job $j$ has weight $w_j$ and resource requirement $r_{jd}$ in resource $d \in [D]$. If the job executes at rate $y_{jd}$, it consumes an amount $r_{jd}y_{jd}$ of resource $d$. The weighted DRF allocation sets $y_{jd}$ so that: (1) $\sum_{d} r_{jd}y_{jd} \leq 1$ for all dimensions $d$; and (2) $w_jy_{jd} \max_{d} r_{jd}$ is the same for all jobs, i.e., the weighted share of the dominant resource consumed is equalized. In the two models, we show $O(1)$-approximate allocations with $O(nD \log^* n)$ and $O(nD \log n)$ disruptions, respectively.

Acknowledgment. Im is supported in part by NSF grants CCF-1409130, 1617653, and 1844939. Mosley is supported in part by a Google Research Award and NSF grants CCF-1617724, 1733873 and 1725543. Munagala is supported in part by NSF grants CCF-1408784, 1637397, and IIS-1447554, ONR award N00014-19-1-2268, and awards from Adobe and Facebook. Pruhs is supported in part by NSF grants CCF-1421508, 1535755, and an IBM Faculty Award.

REFERENCES

[1] Eric Friedman, Christos-Alexandros Psoymos, and Shai Vardi. Dynamic fair division with minimal disruptions. In Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC ’15, pages 697–713. ACM, 2015.

[2] Eric Friedman, Christos-Alexandros Psoymos, and Shai Vardi. Controlled dynamic fair division. In Proceedings of the 2017 ACM Conference on Economics and Computation, EC ’17, pages 461–478. ACM, 2017.

[3] Ian Kash, Ariel D. Procaccia, and Nisarg Shah. No agent left behind: Dynamic fair division of multiple resources. In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS ’13, pages 351–358. International Foundation for Autonomous Agents and Multiagent Systems, 2013.