Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Dynamic Fair Resource Division

Shai Vardi, Alexandros Psomas, Eric Friedman

To cite this article:
Shai Vardi, Alexandros Psomas, Eric Friedman (2022) Dynamic Fair Resource Division. Mathematics of Operations Research 47(2):945-968. https://doi.org/10.1287/moor.2021.1155

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 47, No. 2, May 2022, pp. 945–968
ISSN 0364-765X (print), ISSN 1526-5471 (online)
http://pubsonline.informs.org/journal/moor

Dynamic Fair Resource Division

Shai Vardi,[a] Alexandros Psomas,[b] Eric Friedman[c]

[a] Krannert School of Management, Purdue University, West Lafayette, Indiana 47907; [b] Department of Computer Science, Purdue University, West Lafayette, Indiana 47907; [c] International Computer Science Institute and Electrical Engineering and Computer Sciences Department, Berkeley, California 94704

Contact: svardi@purdue.edu, https://orcid.org/0000-0003-4720-6826 (SV); apsomas@purdue.edu (AP); ejf@icsi.berkeley.edu (EF)

Received: January 17, 2020
Revised: November 18, 2020
Accepted: February 18, 2021
Published Online in Articles in Advance: September 27, 2021

MSC2020 Subject Classification: Primary: 90B35

https://doi.org/10.1287/moor.2021.1155

Abstract. A single homogeneous resource needs to be fairly shared between users that dynamically arrive and depart over time. Although good allocations exist for any fixed number of users, implementing these allocations dynamically is impractical: it typically entails adjustments in the allocation of *every* user in the system whenever a new user arrives. We introduce a dynamic fair resource division problem in which there is a limit on the number of users that can be disrupted when a new user arrives and study the trade-off between fairness and the number of allowed disruptions, using a fairness metric: the *fairness ratio*. We almost completely characterize this trade-off and give an algorithm for obtaining the optimal fairness for any number of allowed disruptions.

Funding: This work was supported by the National Science Foundation [Grants 1161813 and 1216073]. A. Psomas is supported by a Google Research Scholar Award.

Keywords: fair division • dynamic resource allocation

1. Introduction

Fair division has been a central topic in operations research, economics, computer science, and mathematics for decades. There has been a recent surge in interest in fair division because of its applications to resource sharing in data centers and the cloud. Fairness is particularly important when a large organization, such as Microsoft, Google, or a university, wants to allocate a shared resource among its employees or students. Isard et al. [21, p. 1–2] note that Microsoft's "users generally strongly desire some notion of fair sharing of the cluster resources. The most common request is that one user's large job should not monopolize the whole cluster, delaying the completion of everyone else's (small) jobs." There are several accepted notions of fairness in the literature, for example, *envy-freeness* (no agent would like to exchange shares with any other agent) and *equitability* (every agent has the same utility). Arguably the most widely accepted notion is *proportionality*: if there are n agents in the system, each agent is allocated at least a $1/n$ fraction of what the agent would receive if the agent were allocated all of the resources.

Traditionally, research on fair division has focused on static allocations. In practice, however, allocation protocols are usually dynamic and must accommodate users that arrive in and depart from the system over time. A major difficulty in dynamic resource allocation settings stems from the need to balance *fairness*, *efficiency*, and *disruption costs*—the costs to the system from disrupting a running process in order to reallocate some (or all) of the resource currently assigned to it. Any two are straightforward to achieve if the third is not required. As an example, consider a cloud computing platform that allocates a single homogeneous resource, such as RAM, to its users, and the users arrive one by one. If memory reallocation could be done instantly and without cost, it would be trivial to always maintain a fair and efficient allocation: whenever a user arrives, simply redistribute all of the RAM evenly. If fairness is of no concern, one can simply allocate all of the RAM to the first user that arrives, free it up when the user leaves and allocate it to the next user, creating the problem noted by Isard et al. [21]. Finally, without concern for efficiency, the platform could simply allocate each user some infinitesimal amount of its total RAM—a perfectly fair solution—and no reallocation is ever required. Naturally, if there are only a couple of users present, this leads to a substantial underutilization of the platform's resources.

The handful of papers that address the topic of dynamic fair division (e.g., Kash et al. [23], Walsh [33]) have not allowed reallocation. In the real world, however, efficiency is paramount, and it is impossible to achieve in dynamic settings without reallocation. On the other hand, in real systems, every attempt is made to minimize reallocation/preemption: if there are many successive arrivals, much of the time could be spent on reallocation instead of resource consumption (Isard et al. [21], Milojičić et al. [25], Verma et al. [32]). Therefore, most systems impose some limit on the amount of disruptions. For example, Borg, Google's current cluster manager, has the following capability built in: " … a limit can be imposed on the number of task disruptions (reschedules or preemptions) an update causes; any changes that would cause more disruptions are skipped" (Verma et al. [32, p. 2–3]).

In this paper, we consider a single homogeneous, infinitely divisible resource that needs to be shared among users that arrive and depart over time. There is a limit on the number of disruptions to allocations, modeled by a control vector, and each entry represents the amount of allowed disruptions: if the tth entry in the control vector is z (possibly $z = 0$), then up to z users can be disrupted when there are $t - 1$ users in the system and another arrives. No disruptions are allowed upon departure. An allocation for t users is *proportional* if each user receives a $1/t$ fraction of the resource. The *fairness ratio* of an allocation is the ratio between the smallest share and the proportional one. Our goal is to design an algorithm that maximizes the minimum fairness ratio across all allocations given any sequence of arrivals and departures. Two natural control vectors, which are given special attention, are d-uniform control vectors, in which every entry is d, and c-gap control vectors, in which there are at most c consecutive zeros in the control vector. d-uniform control vectors model the case in which a system allows a constant number disruptions per arrival; c-gap control vectors model systems in which disruptions are allowed only rarely: once every $c + 1$ arrivals. We note that these are strict requirements; therefore, our results provide pessimistic bounds for systems that may have more lenient requirements (for example, requiring that there be one disruption for every c arrivals in expectation).

Example 1. Consider a toy system with capacity for three users with the control vector (1, 1, 1): the system is allowed to disrupt at most one existing user whenever a new user arrives. One possible algorithm divides the largest available share equally at each arrival. When the first user arrives, the user is allocated the entire resource; when the second user arrives, the user is given half of the resource, and the first user's resource is halved; when the third user arrives, only one user can be disrupted, and so one user is allocated half of the resource and the two other users are allocated one quarter of the resource each. To gauge the fairness of this algorithm, we compare each of the allocations with the proportional shares: when there are one or two users in the system, the allocations of the algorithm and the proportional shares are identical; when there are three users in the system, the proportional share is one third, but the smallest share is one quarter, $3/4$ of the proportional share. The fairness ratio of this algorithm is, therefore, $3/4$. Intuitively, we should be able to do better that $3/4$: given that we know we are only able to disrupt one user in round 3, it is better to keep one user having a larger share than the other in round 2. Although this is not perfectly fair in round 2, it improves the fairness in round 3, and by balancing the two, we can obtain the optimal fairness. We see in Section 4 that the optimal fairness ratio for this example is $6/7$.

1.1. Our Results

We study the trade-off between the fairness ratio and the number of disruptions in system with a single homogeneous resource with users that arrive and depart dynamically. We almost completely characterize this trade off and design algorithms for obtaining the optimal fairness as a function of the amount of reallocation allowed.

We first describe an optimal algorithm; that is, an algorithm that attains the best possible fairness ratio for any input (control vector). To achieve this, we solve a closely related problem: given a control vector and a fairness ratio, output allocations that guarantee this fairness ratio for this control vector or conclude that no such allocations exist. We design an algorithm for the second problem; our original problem now reduces to computing the optimal fairness ratio of the input control vector.

We show that, if there is a limit on the number of users a system can accommodate (i.e., the control vector has finite size), it is possible to precisely compute the optimal fairness ratio and, hence, to compute optimal allocations. In addition, we show that we can handle systems without an a priori upper bound on the number of users. As long as the set of disruptions can be described succinctly, we can compute the optimal fairness ratio and show that the same algorithm is asymptotically optimal. As we do not wish to restrict the system to supply us with a control vector up front, only to commit to the value of c, we bound the fairness ratio of *all* possible control vectors simultaneously. We compute exact bounds when (at least) $d \geq 1$ disruptions are allowed per arrival and almost matching upper and lower bounds on the fairness ratio of all control vectors that allow one disruption for every $c > 1$ users. A summary of these bounds appears in Table A.1 in the appendix.

We show that a very small number of disruptions suffices to guarantee good fairness: five disruptions per arrival, for example, guarantee a fairness ratio of approximately 0.914. In addition, the fairness ratio decays logarithmically with the number of arrivals one has to wait between disruptions. This implies that, even with very few disruptions, we can guarantee a (surprisingly) high fairness ratio: even if we only require a single disruption for every 20,000 arrivals, each user can still be guaranteed at least $1/10$ of the amount of resource the user would receive if there was no cost to reallocation. In addition, we show that, in many cases, the worst allocations do not occur when there are many users in the system. This means that allowing some flexibility at the beginning—adding a handful of additional disruptions to handle the first few arrivals—can lead to improved fairness ratios and efficiency.

Our results and proofs give several more insights into the limitations that disruption requirements (or alternatively fairness requirements) impose on systems. We show that there is a very small difference in the fairness of systems of different sizes. As an example, the fairness ratio of the algorithm that allows a single disruption per arrival in a system with capacity for 100 users is 0.727, and for a system with one billion users, it is 0.721. In addition, the system does not need to compute all of the allocations in advance: our algorithm can compute the allocations "on the fly" as users arrive and depart. This is especially important for very large or unbounded systems. Finally, we show that—as long as at least one disruption is allowed per arrival—we do not need to sacrifice efficiency for fairness: there always exist allocations that are Pareto efficient (i.e., always allocate the entire resource) and achieve the optimal fairness.

1.2. Related Work

Fair division has been a central topic in operations research (Bertsimas et al. [6], Correa et al. [11], Deng et al. [13], Karsten et al. [22]), economics (Budish [9], Pazner and Schmeidler [28], Steinhaus [30]), management (Boiney [7], Fishburn and Sarin [14], Haitao Cui et al. [19]), mathematics (Alon [2], Brams and Taylor [8]), and computer science (Aziz and Mackenzie [3], Othman et al. [27]).

Walsh [33] is the first to study the problem of online fair cake cutting. He considers the scenario in which users arrive, receive a piece of cake, and depart and shows how several well-known fair division solutions (cut and choose, Dubins–Spanier, etc.) can be adapted to satisfy desirable properties in an online setting with a single (heterogeneous) divisible cake. More recently and closer to the problem studied here, Kash et al. [23] introduce a model of dynamic allocations. However, their model only considers arrivals, and their main algorithm reserves resources for future arrivals; it does not allow the reallocation of resources or users to depart. This leads to allocations that satisfy neither our definition of fairness nor Pareto efficiency as resources are left idle. In our model, users are homogeneous; Li et al. [24] extend our results to heterogeneous users and design an algorithm that has an upper bound on the fairness ratio of $O(\log n)$ when at least one disruption is allowed per arrival.

Guo et al. [18] study the problem of repeatedly allocating a single item between competing users. They give allocation algorithms that do not allow monetary transfers with good competitive ratios with respect to optimal allocation algorithms with payments. Segal-Halevi [29] studied the problem of redividing a two-dimensional resource subject to fairness and "geometric" constraints on the allocations. Isard et al. [21] consider scheduling with locality and fairness constraints. They evaluate different algorithms with and without requiring fairness and with preemption enabled or disabled. They find that requiring fairness and allowing preemption gives the best overall performance regardless of the scheduler implementation that is used. Freeman et al. [15] study the trade-off between strategy-proofness, efficiency, and fairness in a setting with dynamically changing preferences.

Many papers study dynamic resource allocation without the restriction of fairness (Ahmadi et al. [1], Huh et al. [20]). Topaloglu and Powell [31] study the dynamic allocation of indivisible resources to tasks from a multiagent decision-making perspective when the tasks arrive from some known distribution. Ciocan and Farias [10] consider a different model in which there is volatile demand for resources, and the goal is to maximize revenue. Although their model and results are completely disjoint from ours, they too point out that an unattractive solution to the dynamic version is to simply solve "off-line" versions of the allocation problem at hand.

Benade et al. [4] and Zeng and Psomas [34] study a complementary setting in which users are static and resources arrive over time. Finally, the networking community studies the problem of fairly allocating a single homogeneous resource in a queuing model in which each agent's task requires a (given) number of time units to be processed. In these models, even though tasks are processed over time, demands stay fixed, and there are no other dynamics, such as agent arrivals and departures. The well-known fair queuing solution (Demers et al. [12])—which has been also analyzed by economists (Moulin and Stong [26])—allocates one unit per agent in a successive round-robin fashion.

2. The Model

A homogeneous, infinitely divisible resource is shared among users that arrive and depart over time. We normalize the amount of resource to one. Denote an allocation for t users by a vector $X^t \in [0,1]^t$. For convenience, we assume that the vector is sorted in nonincreasing order and denote the jth largest allocation of X^t by X_j^t. An allocation X^t is *feasible* if $\sum_{j=1}^{t} X_j^t \leq 1$. An allocation is *Pareto efficient* if there is no user i whose share of the resource can be improved without strictly decreasing the share of some other user j. Equivalently an allocation X^t is Pareto efficient if $\sum_{j=1}^{t} X_j^t = 1$. The following measure is used to compare allocations.

Definition 1 (Domination). Let $A = (a_1, \ldots, a_t)$ and $B = (b_1, \ldots, b_t)$ be two vectors. We say that *A weakly dominates B*, denoted $A \succcurlyeq B$, if $\forall\, i \in \{1, \ldots, t\}$, $a_i \geq b_i$.

2.1. Control Vectors

Whenever any user's allocation is reduced, we say that the user is *disrupted*. We note that increasing a user's allocation does not count as a disruption. For most of the paper, we assume that the input to our problem is a vector defining how many disruptions are allowed: a *control vector*. We again stress that a large portion of our results do not require that the control vector is given as an input ahead of time.

Definition 2 (Control Vector). A control vector ψ is a vector, for which $\psi[t]$ denotes the number of users that may be disrupted when there are $t-1$ users in the system and another one arrives.

The first entry in any control vector ψ, that is, $\psi[1]$, is redundant as it refers to the empty system; nevertheless, we feel the notation is clearer when it is included. Note that the definition is not restricted to finite control vectors and can be used to define infinite control vectors as well (e.g., $(1, 1, 1, \ldots)$). We sometimes refer to a user whose allocation is reduced as a *donor*.

Example 2. Consider the control vector $\psi = (1, 0, 1)$. When there is a single user in the system and another user arrives, the first user's allocation cannot be disrupted. When there are two users and a third arrives, the allocation of one of the two users currently in the system can be disrupted. Note that the control vectors (0, 0, 1) and (1, 0, 1) impose the same restrictions on the allocation process.

If $\psi[i] = d$ for all i, we say ψ is a *d-uniform control vector*. For example (2, 2, 2, 2) is a 2-uniform control vector. The vector $(2, 2, 2, \ldots)$ is called the infinite d-uniform control vector.

If the maximal number of consecutive zeros in a control vector ψ is c, we call ψ a *c-gap control vector*. We define a *basic c-gap control vector* to be an infinite control vector in which there is exactly one donor every $c + 1$ arrivals; otherwise, the control vector is nonbasic. There are exactly $c + 1$ possible basic c-gap control vectors.

Example 3. The three possible basic 2-gap control vectors are
1. $(0, 1, 0, 0, 1, 0, 0, 1, 0, \ldots)$ also denoted $(0,1,0)^\infty$.
2. $(0, 0, 1, 0, 0, 1, 0, 0, 1, \ldots)$ also denoted $(0,0,1)^\infty$.
3. $(1, 0, 0, 1, 0, 0, 1, 0, 0, \ldots)$ also denoted $(1,0,0)^\infty$.

We use 0^c to denote c consecutive zero entries. For example, $(0^c 1)^\infty$ denotes the vector $(0, 0, \ldots, 0, 1)^\infty$, where zero appears c times.

Definition 3 (Allocation Algorithm). An *allocation algorithm* is an algorithm that receives as input a (possibly infinite) control vector ψ and a (possibly infinite) vector of arrivals and departures ϕ, and each entry corresponds to a single arrival or departure. The algorithm's output is a set of $|\phi|$ allocations (in which $|\phi|$ denotes the length of ϕ). When there are $t-1$ users and another arrives, the algorithm reduces the allocation of at most $\psi[t]$ agents. When a user departs, the algorithm does not reduce the allocation of any user except that of the departing user. The algorithm may augment the resource of any number of users for both arrivals and departures.

The property that only the allocation of the departing user is reduced is sometimes called *population monotonicity*. We allow the allocation algorithm to augment resources freely because augmenting a resource can typically be done without disruption: the user can choose to simply ignore the added resource and not use it. Although we do not restrict the number of augmentations, the allocation algorithms that we describe do not augment the resource of any agent that is already in the system during an arrival. Further, although allocation algorithms are allowed to disrupt any user, all the algorithms that we describe only disrupt the users with the largest allocation.

Definition 4 (History Independence). An allocation algorithm is *history independent* if its allocations only depend on the number of users in the system and not on ϕ. For simplicity, we denote the input to a history-independent algorithm only by the control vector, ψ, and usually denote its output by a set of allocations, $X = \{X^1, X^2, \ldots\}$, where X^t is the allocation when t users are present. Note that $|X| = |\psi|$.

For history-independent algorithms, we sometimes refer to the time period when there are t users in the system as *time t*. For simplicity, we focus on history-independent allocation algorithms for the rest of this section; it is straightforward to extend the definitions to arbitrary allocation algorithms, but this comes at the expense of additional notation. We show that restricting our attention to history-independent algorithms does not come at a loss of generality. If only arrivals are allowed, every allocation algorithm can be viewed as history independent. In Sections 3–5, we describe algorithms and focus on how they handle arrivals. We ignore departures and simply assume that all the algorithms described can be augmented to handle departures such that they are history independent. In Section 6, we show that this is indeed the case, and it is straightforward to augment these algorithms to guarantee history independence.

2.2. Fairness

When there are t users in the system, the proportional share is $1/t$ of the resource. We define the fairness ratio of an allocation X^t to be the ratio between its smallest share and the proportional share. More formally, let X^t denote the allocation vector of a history-independent allocation algorithm $\mathcal{A}$ with control vector ψ when there are t users present (recall that X^t is sorted). Then,

$$\mathrm{FAIRNESS}(\mathcal{A}, \psi, t) = \frac{\mathrm{X}_t^t}{1/t}.$$

Note that this is always a real number in $[0,1]$. We further define $\mathrm{FAIRNESS}(\mathcal{A},\psi) = \inf_{t>0} \mathrm{FAIRNESS}(\mathcal{A},\psi,t)$. Let $\mathcal{H}$ denote the set of all history-independent allocation algorithms. The fairness ratio of a control vector ψ is defined as the supremum $\mathrm{FAIRNESS}(\mathcal{A},\psi)$ over all possible history-independent allocation algorithms:

$$\mathrm{FAIRNESS}(\psi) = \sup_{\mathcal{A}\in\mathcal{H}} \{\mathrm{FAIRNESS}(\mathcal{A},\psi)\}.$$

An *optimal history-independent algorithm* $\mathcal{A}$ is a history-independent allocation algorithm such that, for any ψ, $\mathrm{FAIRNESS}(\mathcal{A},\psi) = \mathrm{FAIRNESS}(\psi)$.

We are interested in the fairness ratio of (possibly infinite) d-uniform and c-gap control vectors. Let $\mathcal{D}$ and $\mathcal{C}$ denote the set of d-uniform and c-gap control vectors, respectively. Then,

$$\mathrm{FAIRNESS}(d - \mathrm{uniform}) = \inf_{\psi\in\mathcal{D}} \{\mathrm{FAIRNESS}(\psi)\}.$$

Similarly, we define the fairness ratio of c-gap control vectors.

$$\mathrm{FAIRNESS}(c - \mathrm{gap}) = \inf_{\psi\in\mathcal{C}} \{\mathrm{FAIRNESS}(\psi)\}.$$

We require one more definition. For some infinite control vectors, it may be the case that the worst fairness ratio occurs when there are only a few users in the system (see, e.g., Figure B.1). In these cases, it may be reasonable for the system designer to allow a few more disruptions earlier on to guarantee that the worst case fairness ratio occurs at the limit. We would, therefore, like to compute the worst fairness ratio for control vectors at the limit, ignoring the outliers early on. To that end, we define $\mathrm{fairness}_{\geq n}(\psi) = \sup_{\mathcal{A}\in\mathcal{H}} \{\inf_{t\geq n} \{\mathrm{FAIRNESS}(\mathcal{A},\psi,t)\}\}$, and $\mathrm{fairness}_{\geq n}(c - \mathrm{gap}) = \inf_{\psi\in\mathcal{C}} \{\mathrm{fairness}_{\geq n}(\psi)\}$. We note that this phenomenon does not occur for d-uniform control vectors: for d-uniform control vectors, the fairness ratio always decreases as the length of the vector increases (Proposition 2). We, therefore, do not need to study this limit behavior separately for d-uniform control vectors.

3. An Optimal Algorithm, Assuming the Desired Fairness Ratio Is Known

Our goal is to design an optimal history-independent allocation algorithm $\mathcal{A}$. We distinguish between two cases: finite and infinite ψ. For both, the main building block is an algorithm for the following problem: given a control vector ψ and a fairness ratio σ, find feasible allocations with fairness ratio σ or decide that the fairness ratio σ is impossible to achieve. We shortly describe an optimal algorithm—the frugally fair algorithm (FFA)—for this related problem, but first let us see how we can use it to design algorithms for the general problem. In both the finite and infinite cases, if we know the optimal fairness ratio for ψ, $\mathrm{FAIRNESS}(\psi)$, we can invoke FFA with ψ and $\mathrm{FAIRNESS}(\psi)$ and generate optimal allocations. We note that FFA can generate the allocations on the fly; it does not need to generate all allocations up front, and as such, it can easily handle infinite control vectors. In Section 4, we show how to compute the optimal fairness ratio for any finite control vector. In Section 5, we compute almost tight bounds for the fairness ratio of infinite control vectors. We note that, with fewer than one disruption per arrival, Pareto optimality is incompatible with any positive fairness ratio.

Proposition 1. *For any ψ, for which $\psi[t] = 0$ for some $t \geq 2$ and any Pareto-optimal allocation algorithm $\mathcal{A}$, it holds that* $\mathrm{FAIRNESS}(\mathcal{A},\psi) = 0$.

Proof. Let $\mathcal{A}$ be a Pareto-efficient allocation algorithm. Let t be the first coordinate for which $\psi[t] = 0$. As the algorithm is Pareto efficient, there is no available resource, and therefore, user t receives nothing, resulting in FAIRNESS$(\mathcal{A}, \psi) = 0$. □

We now describe FFA, and note that we allow FFA to return infeasible allocations (i.e., ones such that $\sum_{i=1}^{t} X_i^t > 1$). It is straightforward to modify FFA to return an error message if there is some time t at which the total resource allocated is greater than one.

Algorithm 1 (Frugally Fair Algorithm)

Input: a control vector ψ and a fairness ratio σ.

When user $1 \le t \le |\psi|$ arrives, allocate $\frac{\sigma}{t}$ of the resource to that user. Reduce the shares of the users with the $\psi[t]$ largest shares to $\frac{\sigma}{t}$ of the resource as well.

As we assume that FFA is history independent, its output can be viewed as a set of allocations, $X = \{X^1, X^2, \ldots,\}$. We show that FFA is optimal in the following sense: for any control vector ψ, if there is any allocation algorithm $\mathcal{A}$ such that FAIRNESS$(\mathcal{A}, \psi) \ge \sigma$, then FFA gives feasible allocations when given ψ and σ as inputs.

Theorem 1. *If there exists an algorithm $\mathcal{A}$ and a (finite or infinite) control vector ψ such that* FAIRNESS$(\mathcal{A}, \psi) \ge \sigma$*, then the allocations produced by* FFA(ψ, σ) *are all feasible.*

By Proposition 1, FFA cannot always be Pareto optimal. However, if there are no zeros in the control vector, it is straightforward to convert FFA to a Pareto optimal algorithm without loss in fairness, for example, by simply adding the leftover resource (if there is any) to the user with the largest share.

Proof of Theorem 1. Let ψ be an input for any allocation algorithm $\mathcal{A}$ and FFA. Let FAIRNESS$(\mathcal{A}, \psi) \ge \sigma$, that is, $\mathcal{A}$ with input ψ produces feasible allocations with fairness ratio σ. We consider FFA when executed with input σ and ψ and show that FFA is feasible as well. We assume without loss of generality that $\mathcal{A}$ never increases the allocation of any user that is already in the system.

Let $\mathcal{A}^t$ and FFA^t be the sorted allocations of $\mathcal{A}$ and FFA, respectively, for t users in the system, in which $\mathcal{A}$ is given ψ as an input and FFA is given ψ and σ as inputs. Let $\mathcal{A}_i^t$ and FFA$_i^t$ denote the ith entry in the allocation vectors $\mathcal{A}^t$ and FFA^t, respectively. As the allocations of $\mathcal{A}$ are always feasible, it suffices to show that $\mathcal{A}^t \succcurlyeq \text{FFA}^t$ for all $t \le |\psi|$. We prove this by induction on the number of users in the system, t.

The base case: By the definition of FFA, FFA$_1^1 = \sigma$. It must clearly hold that $\mathcal{A}_1^1 \ge \sigma$ by the definition of FAIRNESS$(\mathcal{A}, \psi)$.

The inductive step: Assume the statement holds for $t-1$ users. We show it holds for t. There are two cases: $\psi[t] = 0$ and $\psi[t] \ge 1$.

Case $\psi[t] = 0$: Let the allocation of FFA at time $t-1$ be $(\text{FFA}_1^{t-1}, \ldots, \text{FFA}_{t-1}^{t-1})$. Then, $\text{FFA}^t = (\text{FFA}_1^{t-1}, \ldots, \text{FFA}_{t-1}^{t-1}, \frac{\sigma}{t})$. Algorithm $\mathcal{A}$ allocates an $x \ge \frac{\sigma}{t}$ amount of the resource to the incoming user. Let the allocation of $\mathcal{A}$ at time $t-1$ be $(\mathcal{A}_1^{t-1}, \mathcal{A}_2^{t-1}, \ldots, \mathcal{A}_{t-1}^{t-1})$. Then,

$$\mathcal{A}^t = (\mathcal{A}_1^{t-1}, \mathcal{A}_2^{t-1}, \ldots, \mathcal{A}_j^{t-1}, x, \mathcal{A}_{j+1}^{t-1}, \ldots, \mathcal{A}_{t-1}^{t-1}),$$

for some $0 \le j \le t-1$. It is easy to see that $\mathcal{A}^t \succcurlyeq \left(\mathcal{A}_1^{t-1}, \mathcal{A}_2^{t-1}, \ldots, \mathcal{A}_{t-1}^{t-1}, \frac{\sigma}{t}\right)$ for any such value of j. Furthermore, by the induction hypothesis,

$$\left(\mathcal{A}_1^{t-1}, \mathcal{A}_2^{t-1}, \ldots, \mathcal{A}_{t-1}^{t-1}, \frac{\sigma}{t}\right) \succcurlyeq (\text{FFA}_1^{t-1}, \ldots, \text{FFA}_{t-1}^{t-1}, \frac{\sigma}{t}) = \text{FFA}^t.$$

Case $\psi[t] \ge 1$: Starting with the allocation vector at time $t-1$, $\mathcal{A}^{t-1}$, we break the allocation when the tth agent arrives into $|\psi| + 1$ steps. If the algorithm performs $d < \psi[t]$ disruptions, we represent this by choosing an arbitrary donor and not changing the share.

1. In step ℓ, for $\ell = 1, \ldots, |\psi|$, reduce the share of the ℓ th donor to get $\mathcal{A}^{(t-1,\ell)}$.
2. In step $d+1$, allocate x to the incoming user to obtain $\mathcal{A}^t$.

Denote $\mathcal{A}^{(\min,\ell)} = (\mathcal{A}_{\ell+1}^{t-1}, \ldots \mathcal{A}_{t-1}^{t-1}, \frac{\sigma}{t}, \ldots, \frac{\sigma}{t})$, where $\frac{\sigma}{t}$ appears ℓ times. We show that, for all $\ell = 1, \ldots, d$, $\mathcal{A}^{(t-1,\ell)} \succcurlyeq \mathcal{A}^{(\min,\ell)}$. The last step (step $d+1$) is identical to the $\psi[t] = 0$ case, and hence, by combining the two observations, we can conclude that $A^t \succcurlyeq \text{FFA}^t$.

We focus on the $\ell = 1$ case; the other cases are identical. To show that $\hat{\mathcal{A}}^{(t-1,1)} \succcurlyeq \mathcal{A}^{(\min,1)}$, assume algorithm $\mathcal{A}$ reduces the allocation of a user from $\mathcal{A}_j^{t-1}$ to y, and $1 \le j \le t-1$ (possibly $\mathcal{A}_j^{t-1} = y$). There is some k, $j \le k \le t-1$, such that $\mathcal{A}_k^{t-1} \ge y \ge \mathcal{A}_{k+1}^{t-1}$. Then,

$$A^{(t-1,1)} = (\mathcal{A}_1^{t-1}, \mathcal{A}_2^{t-1}, \ldots, \mathcal{A}_{j-1}^{t-1}, \mathcal{A}_{j+1}^{t-1}, \mathcal{A}_{j+2}^{t-1}, \ldots, \mathcal{A}_k^{t-1}, y, \mathcal{A}_{k+1}^{t-1}, \ldots \mathcal{A}_{t-1}^{t-1}).$$

For $i \in [1, j-1]$, $\mathcal{A}_i^{(t-1,1)} \ge \mathcal{A}_i^{(\min,1)}$. For $i \in [j,k]$, $\mathcal{A}_i^{(t-1,1)} = \mathcal{A}_i^{(\min,1)}$. Then, by definition of y, $\mathcal{A}_{k+1}^{(t-1,1)} = y \ge \mathcal{A}_{k+1}^{(t-1,1)} = A_{k+1}^{(\min,1)}$. Similarly, $\hat{A}_i^{t-1} \ge \mathcal{A}^{\min_i}$ for all $i \in [k+1, t-2]$. For the last term, note that, because FAIRNESS$(\mathcal{A}, \psi) \ge \sigma$, $\mathcal{A}_{t-1}^{t-1} \ge \frac{\sigma}{t-1} \ge \frac{\sigma}{t}$ (possibly y is the last share, but then $y \ge \frac{\sigma}{t}$ as FAIRNESS$(\mathcal{A}, \psi) \ge \sigma$). □

We now give a simple result, whose proof is straightforward and is deferred to Appendix D.

Lemma 1. *The allocations produced by FFA(ψ, σ) as inputs are identical to those produced by FFA(ψ, σ') scaled by $\frac{\sigma}{\sigma'}$.*

4. Finite Control Vectors

If the input control vector ψ is finite, computing the optimal fairness is straightforward. We invoke FFA with ψ and $\sigma = 1$. Typically, some of these allocations are infeasible. By Lemma 1, when FFA is run with the same ψ but different values of σ, the allocations are different, but their relative size remains the same; we can, therefore, compute the smallest amount by which we need to scale the allocations produced by FFA with ψ and $\sigma = 1$ down so that the largest total allocation is one, and this is exactly the optimal fairness ratio. The optimal algorithm for finite control vectors, FiniteFFA, is, therefore, the following: Execute FFA with inputs ψ and $\sigma = 1$ to compute the optimal fairness ratio of ψ and then scale the allocations by this value.

Algorithm 2 (FiniteFFA)

Input: a finite control vector ψ.

Output: a set of allocations X.

1. Execute FFA with input ψ and $\sigma = 1$. Let S^t be the total resource allocated by FFA when there are t users, $1 \le t \le |\psi|$. Set $\sigma^* = (\max_{1 \le t \le |\psi|}\{\mathrm{S}^t\})^{-1}$.

2. Output the allocations of step 1, scaled by σ^*.

Theorem 2. *FiniteFFA is optimal for any finite control vector.*

Proof. Fix ψ. Let S^t denote the total resource allocated at time t when FFA is executed with input ψ and $\sigma = 1$. By Lemma 1, $\mathrm{S}^t = \hat{\sigma}^{-1}\hat{\mathrm{S}}^t$, where $\hat{\mathrm{S}}^t$ is the total resource allocated at time t by FFA with input ψ and $\hat{\sigma}$ for any $\hat{\sigma}$. Therefore, FFA executed with input ψ and $\sigma^* = (\max_{1 \le t \le N}\{\mathrm{S}^t\})^{-1}$ is feasible. Furthermore, FFA executed with input ψ and $\hat{\sigma} > \sigma^*$ creates an infeasible allocation at some time $t \le |\psi|$. The contrapositive of Theorem 1 then implies that there is no allocation algorithm $\mathcal{A}$ that produces feasible allocations with fairness ratio $\hat{\sigma} > \sigma^*$. It follows that FiniteFFA is optimal for ψ. □

It is straightforward to compute the fairness ratio of finite control vectors, using the procedure given in step 1 of FiniteFFA. In Table 1, we give the optimal fairness ratios for some values of d for d-uniform control vectors.

In addition to the values in Table 1, we give a simple lower bound on the fairness ratio of *any* finite control vector. We note that this applies even to adversarial control vectors with which the adversary is completely unrestricted (and can, therefore, choose the all-zero vector). We denote the nth harmonic number by H_n; that is, $H_n = \sum_{i=1}^{n} \frac{1}{i}$.

Theorem 3. *Let ψ be a finite control vector, $|\psi| = n$. Then,* FAIRNESS$(\psi) \ge (H_n)^{-1}$.

Proof. Consider an algorithm that, for any input vector ψ such that $|\psi| = n$, allocates to the tth user a $\frac{1}{t \cdot H_n}$ fraction of the resource upon arrival and never performs any disruptions. The algorithm is always feasible because $\sum_{t=1}^{k} \frac{1}{t \cdot H_n} = \frac{H_k}{H_n} \le 1$ for all $k \le n$, and by definition, it achieves a fairness ratio of ${H_n}^{-1}$. □

Table 1. Optimal fairness ratios for finite d-uniform control vectors for some values of d.

$\lvert\psi\rvert$	d					
	1	2	3	5	10	50
2	1	1	1	1	1	1
3	0.857	1	1	1	1	1
5	0.811	0.909	0.952	1	1	1
10	0.774	0.868	0.913	0.957	1	1
100	0.727	0.827	0.874	0.919	0.958	0.995
1,000	0.722	0.823	0.870	0.915	0.954	0.991
100,000	0.721	0.822	0.869	0.914	0.954	0.990

5. Infinite Control Vectors

We now turn to infinite control vectors. Unfortunately, we cannot use FFA to compute the optimal fairness ratio in finite time in this case. Nevertheless, we would like to allow for systems that can accommodate an arbitrary number of users. In these cases, we can still invoke FFA with the infinite control vector ψ and a fairness ratio σ as input, but we must find another way of computing σ. As long as σ is upper bounded by the worst fairness ratio possible for any number of users, the allocations produced by FFA are feasible. We want to give FFA the tightest σ possible, and toward that end, we precisely compute FAIRNESS($d-$uniform) for any $d \geq 1$ in Section 5.1. In Section 5.2, we show that, for $c \geq 3$, different basic c-gap control vectors lead to different fairness ratios, (see also Figures B.1 and B.3). We provide precise asymptotic bounds for $\text{fairness}_{\geq n}(c-\text{gap})$ and almost-tight bounds for FAIRNESS($c-$gap) for $c \geq 3$. In addition, we show that, for $c \in \{1,2\}$, FAIRNESS($c-$gap) = $\text{fairness}_{n\geq 0}(c-\text{gap})$. This implies that the asymptotic bound holds for all time periods for $c = 1$ and $c = 2$, but not for $c \geq 3$ (see Figures B.2 and B.1). A consequence of our results is that we can obtain this fairness ratio for algorithms that only know c, the maximum gap possible between donors, but do not know the exact control vector a priori.

5.1. Infinite Uniform Control Vectors

Our main result in this section is an exact characterization of FAIRNESS($d-$uniform), the optimal fairness ratio of (possibly) infinite d-uniform control vectors. At a high level, the proof is the following: consider the series of fairness ratios of the control vectors $(d), (d,d), (d,d,d), \ldots$. We show that the series converges and compute the limit. In Table 1, one can begin to see the asymptotic behavior of finite d-uniform control vectors as the number of players grows. Note that, given FAIRNESS($d-$uniform), it is straightforward to find feasible allocations for an infinite d-uniform control vector, ψ, by executing FFA with ψ and FAIRNESS($d-$uniform). All proofs for this section are deferred to Appendix E.

Theorem 4. *For any $d \geq 1$, it holds that*

$$\text{FAIRNESS}(d-\textit{uniform}) = \frac{1}{(d+1)\ln\left(\frac{d+1}{d}\right)}.$$

Proof Sketch. The proof of Theorem 4 is via a precise characterization (as a function of σ) of the allocations output by FFA when the input is the infinite d-uniform control vector and an arbitrary σ. We show that the total amount of resource allocated increases with the number of users in the system and compute the limit. We then use Proposition 2, which shows that the fairness ratio is nonincreasing in the length of the control vector, to show that the worst case fairness ratio indeed occurs for the infinite d-uniform control vector (and not for some finite d-uniform control vector). □

Proposition 2. *For any control vectors ψ and ψ' such that ψ' that are a prefix of ψ,*

$$\text{FAIRNESS}(\psi) \leq \text{FAIRNESS}(\psi').$$

The following corollary to Theorem 4 describes the asymptotic behavior with respect to d of the fairness ratio of infinite d-uniform control vectors.

Corollary 1 (to Theorem 4). FAIRNESS$(d-\textit{uniform}) \xrightarrow[d\to\infty]{} 1$.

5.2. General Infinite Control Vectors

Showing bounds for FAIRNESS($c-$gap) is much more involved than for FAIRNESS($d-$uniform) for several reasons. First, there are infinitely many possible infinite c-gap control vectors as opposed to exactly one for constant $d \geq 1$. For example, the only infinite two-uniform control vector is $(2,2,2,\ldots)$. Furthermore, the fairness ratio of a two-uniform control vector is a lower bound on the fairness ratio of any control vector of identical length in which every element is at least two as one can always simply execute FFA on the two-uniform control vector, ignoring the extra available donors. This is not the case when there are steps in which no donor is allowed; we cannot simply use zero donors every round and obtain any meaningful result. In this section, we consider *binary* control vectors. This is without loss of generality as we would like to compute the worst case fairness ratio (and, similarly to the preceding argument, any allocation algorithm can always treat an element in the control vector that is greater than one as one).

We note that, even if we only considered *basic* c-gap control vectors, the allocations produced by FFA for each of them are very different; furthermore, they are not as "well behaved" as the allocations for the d-uniform control vector with $d \geq 1$. Compare Figures B.1 and B.2. The x-axis is the number of users, and the y-axis is the total resource allocated at step t in the execution of FFA with input ψ and $\sigma = 1$. The total resource allocated is, of course, at least one because the optimal fairness ratio is the inverse of the maximal total allocation created, which is at most one. Figure B.2 shows the first 100 allocations created by FFA when given the infinite d-control vector and $\sigma = 1$ as input. Figure B.1 shows the first 30 allocations created by FFA given the four basic three-gap control vectors and $\sigma = 1$ as input. The complexity in the second setting is due to several reasons: the allocations are not monotone, they are not pointwise comparable, and they are not simple transformations one of the other. Furthermore, the allocations do not necessarily take their maxima at the limit. Still, in both cases, the total allocation converges to a limit as the number of users grows (in the second case, *all* infinite allocations that obey certain natural requirements converge to the *same* limit; this can be seen from the proof of Theorem 6). The horizontal line in both figures denotes this limit. By Proposition 2, the first set of allocations takes its supremum at the limit; however Figure B.1 shows that the second case does not. It might be tempting to think that, as in Figure B.1, there always exists *some* basic c-gap control vector that takes its supremum at the limit. However, there exist values of c for which *every* basic c-gap control vector attains its value at finite number of users (see Figure B.3 for a pictorial example for $c = 10$).

The main results of this section are almost matching upper and lower bounds for the optimal fairness ratio attainable for any (possibly infinite) c-gap control vector (Theorem 5) and an asymptotic bound on the fairness ratio (Theorem 6).

Theorem 5. *For any $c \geq 1$, it holds that*

$$(H_{c+1})^{-1} \geq \text{FAIRNESS}(c - gap) \geq \left(H_{2c+3} - \frac{1}{2}\right)^{-1},$$

where H_n denotes the nth harmonic number.

Theorem 5 gives rise to the following corollary, which describes the asymptotic behavior of the fairness with respect to c.

Corollary 2. $\text{FAIRNESS}(c - gap) \underset{c\to\infty}{\longrightarrow} (\ln c)^{-1}$.

Theorem 6. *For every $c \geq 1$, there exists a number $n_c \leq 2(c+4)^2$ such that*

$$\text{fairness}_{\geq n_c}(c - gap) = (c+1)\left((c+2)\ln(c+2)\right)^{-1}.$$

The proofs of Theorems 5 and 6 involve the following two steps:

1. (Section 5.2.1) Showing that basic control vectors are the worst; that is, for every c-gap control vector ψ, there is a basic c-gap control vector ψ' whose fairness ratio is at least as bad (i.e., $\text{FAIRNESS}(\psi') \leq \text{FAIRNESS}(\psi)$).
2. (Section 5.2.2). Bounding the fairness ratio of *all* basic c-gap control vectors concurrently.

In addition, we show that the fairness ratio of all basic c-gap control vectors ψ, $c \in \{1,2\}$, is the same and compute FAIRNESS(1 − gap) precisely for $c = 1$ and $c = 2$. Computing FAIRNESS(1 − gap) for $c \in \{1,2\}$ is more straightforward than for $c \geq 3$ because when $c = 1$ and $c = 2$, the total allocation is maximized at the limit. The proof of the following theorem appears in Appendix F.1.

Theorem 7. *The following hold*:

1. $\text{FAIRNESS}(1 - gap) = 2(3\ln 3)^{-1}$.
2. $\text{FAIRNESS}(2 - gap) = 3(4\ln 4)^{-1}$.

5.2.1. Reduction to Basic Control Vectors. We want to compute a lower bound on the fairness ratio for *all* c-gap control vectors. We first show that basic control vectors have the worst fairness ratio; hence, in order to provide a lower bound on the fairness ratio, it suffices to analyze basic control vectors.

Proposition 3. *For every (finite or infinite) c-gap control vector ψ, there exists some (infinite) basic c-gap control vector $\hat{\psi}$ such that* $\text{FAIRNESS}(\psi) \geq \text{FAIRNESS}\left(\hat{\psi}\right)$.

Proof Outline. We define a (possibly infinite) series of control vectors $\psi_1, \psi_2, \ldots$, where $\hat{\psi} = \psi_1$ is a basic c-gap control vector such that their fairness ratio is nondecreasing. In other words, for any $i \geq 1$, if all of the allocations of $\text{FFA}(\psi_i, \sigma)$ are feasible, then the allocations of $\text{FFA}(\psi_{i+1}, \sigma)$ are also feasible. We define these vectors inductively. ψ_1 is a basic c-gap control vector that shares the longest prefix with ψ. Simply put, ψ_1 has the same number of zeros before the first (leftmost) one as ψ. For $i > 1$, let β_i be the first coordinate on which ψ_i and ψ differ. It must be that $\psi_i[\beta_i] = 0$ and $\psi[\beta_i] = 1$. We define ψ_{i+1} as follows. For $j \in [1, \beta_i - 1]$, set $\psi_{i+1}[j] = \psi_i[j]$; set $\psi_{i+1}[\beta_i] = 1$, and append $(0^c1)^\infty$ (see Example 4). If ψ is finite, the last control vector in the series is ψ with $(0^c1)^\infty$ appended to it. The series defined this way is finite if ψ is either finite or can be represented as a finite vector with $(0^c1)^\infty$ appended to it. Otherwise, it is infinite.

Given this series of vectors, we show that, for every i and all t such that $t = \beta_i \pmod{c+1}$, if the allocation of FFA(ψ_i, σ) when there are t users in the system is feasible, then so is the allocation of FFA(ψ_{i+1}, σ) for t users. We then show that FFA(ψ_{i+1}, σ) remains feasible for all other values of t ($t \neq \beta_i \pmod{c+1}$). The proposition follows from these two facts as together they cover all time steps. □

Example 4. Let $\psi = (0,1,1,1,0,1,0,0,1)$. We derive the ψ_is as follows:

$$\begin{aligned}
\hat{\psi} = \psi_1 &= (0,1),(0,0,1)^\infty,\\
\psi_2 &= (0,1,1),(0,0,1)^\infty,\\
\psi_3 &= (0,1,1,1),(0,0,1)^\infty\\
\psi_4 &= (0,1,1,1,0,1),(0,0,1)^\infty = \psi(0,0,1)^\infty.
\end{aligned}$$

5.2.2. Bounding the Fairness Ratio of Basic Control Vectors. Having shown that the basic control vectors have the worst fairness ratio, it remains to bound it. The allocations created by FFA on basic c-gap control vectors have a particular structure: they resemble a segment of the harmonic series with some doubled entries (see, e.g., Table F.1). Even though we show that, in order to bound the fairness ratio, it is enough to consider only basic control vectors, each basic control vector has a different fairness ratio. Nevertheless, we provide some upper bounds on the fairness ratio for each c. We give two types of bounds: a general and an asymptotic bound. The asymptotic bound is particularly useful for systems that wish to be able to accommodate an unbounded number of users and in which the amount of time the system has fewer than n_0 users (for some constant n_0) is vanishingly small. In this case, one can set the fairness ratio to be the asymptotic fairness ratio with an arbitrary "quick fix" heuristic for when there are fewer than n_0 users in the system (for example, allowing slightly more disruptions to guarantee the asymptotic fairness ratio). The following lemma bounds the fairness of all basic c-gap control vectors concurrently and is the key ingredient in the proofs of both Theorems 5 and 6. We prove this lemma by characterizing all possible allocations that can be created by FFA(ψ, σ) when ψ is a basic c-gap vector for all possible ψ simultaneously. We bound the allocations when the nth user to arrive has the largest share. Notice that this can be at a time when there is a different number of users in the system for different basic c-gap control vectors. The complete proof appears in Appendix F.3.

Lemma 2. *When executing FFA with any basic c-gap control vector and* $\sigma = 1$, *if the nth user to arrive has the largest share, the total resource allocated is at most*

$$\max_{j \in \{0,1,\ldots,c+1\}} \sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} + \sum_{i=0}^{n-2} \frac{1}{n + i(c+1) + j}.$$

From Lemma 2, we immediately get the following.

Corollary 3.

$$\text{FAIRNESS}(c - gap) \geq \left(\max_{n \in \mathbb{N}_{>0}, j \in \{0,1,\ldots,c+1\}} \sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} + \sum_{i=0}^{n-2} \frac{1}{n + i(c+1) + j} \right)^{-1},$$

where $\mathbb{N}_{>0}$ *is the set of positive natural numbers.*

We outline how we use Lemma 2 and Corollary 3 to prove Theorems 5 and 6. The complete proofs appear in Appendices F.4 and F.5.

Proof Sketch for Theorem 5. For the upper bound, we consider the $(c+1)$-th allocation of FFA when its input is the basic c-gap control vector $\psi = (10^c)^\infty$ (and $\sigma = 1$). For the lower bound, we bound each of the terms in Corollary 3 separately. We show that the left term is upper bounded by $H_{2c+3} - 1$, and the right term is upper bounded by $\frac{1}{2}$ for all $c > 1$. □

Proof Sketch for Theorem 6. We use standard bounds on the harmonic numbers to upper bound the first and second terms of the expression in Lemma 2. The sum of these is a function of n, c, and j. We consider the series generated by fixing c and j and varying n and show that this series is increasing in n for sufficiently large values of n. We do this by showing that the derivative is strictly positive for these values of n. Finally, we compute the limit of the expression and show that it is $\frac{c+2}{c+1}\ln(c+2)$. We give a matching lower bound. □

6. Accommodating Departures

Thus far, we assume that our algorithms are history independent. More specifically, as all of our algorithms rely on FFA, we assume that FFA is history independent. In Section 3, we define FFA only with respect to arrivals. We now show how to augment FFA to handle departures such that FFA is history independent.

Theorem 8. *Let ψ be a control vector and σ be a nonnegative real number. Denote the allocations of FFA(ψ,σ) by X^1, $X^2,\ldots$. There exists an algorithm FFA′ that takes as input ψ, σ, and a vector of arrivals and departures such that, for any vector of arrivals and departures ϕ, the allocation specified by FFA′(ψ,σ,ϕ) is X^t whenever there are t agents in the system.*

Proof. It suffices to show that, when an arbitrary user from X^{t+1} departs, it is possible to distribute that user's share in a way that the sorted allocation is X^t. If $\psi[t+1]=0$, this is straightforward: if the user with the jth largest allocation leaves, we simply augment the smallest allocation to X_j^t. Note that, under FFA, the smallest share is given to the t + 1-th arrival. Thus, augmenting it to X_j^t indeed makes the allocation equal to X_t. Without loss of generality, we consider $\psi[t+1]=1$; it is straightforward to extend the proof to $\psi[t+1]>1$. The two allocations we consider are, therefore, $X^t=(a_1,a_2,\ldots,a_t)$ and $X^{t+1}=\left(a_2,a_3,\ldots,a_t,\frac{\sigma}{t+1},\frac{\sigma}{t+1}\right)$.

Denote $S^t=\sum_{i=1}^t X_i^t$. First, assume that one of the last two users (i.e., one with a $\frac{\sigma}{t+1}$ share), departs. Because both X^t and X^{t+1} are feasible, $a_1-2\frac{\sigma}{t+1}$ is equal to the difference S^t-S^{t+1}. Therefore, there must be a way to combine the departing user's share with the other $\frac{\sigma}{t+1}$ share and the unallocated amount $1-S^{t+1}$ to get a_1.

If the departing user is some other user $j\in[2,t-1]$ (whose allocation in X^t is a_j), we do the following: allocate the difference $a_j-\frac{\sigma}{t+1}$ (which is positive because the allocation is sorted) to one of the last two users. The amount left to distribute is exactly $\frac{\sigma}{t+1}$; we've already shown this is sufficient to increase the share of the other of the last two users to a_1. □

We note that the fairness ratio when departures are allowed cannot be better than when they are not despite the fact that we could ostensibly use up the freed resource to increase the allocations as we are not guaranteed any departures. We elaborate on this in Section 7.

7. Conclusion and Future Directions

In this paper, we introduce and study the trade-off between fairness and the amount of allowed disruptions in a dynamic fair division setting. We show that we can achieve good fairness while allowing only a very small number of disruptions as a function of the total arrivals. Our results give insights into how to design shared-resource systems that need to maintain a high degree of fairness while still allowing for smooth operation.

There are various directions in which our research can be extended. Our model is defined so that our results still hold for more in a way that gives a design for the setting in which there is a hard constraint on the number of disruptions per arrival/arrivals between disruptions. For example, one could define the control vector such that $\psi[t]=1$ means that we are allowed to disrupt a user upon the tth arrival independently of the number of departures. As a result, our bounds are worst case bounds for systems whose demands are special cases, for example, ones with a constraint on the total number of disruptions (equivalently, the average number of disruptions per period). Algorithms geared toward these specialized systems might give stronger bounds. In addition, the fairness ratio is a very strong notion of fairness; it requires that the smallest share received by *any* user at *any* time be sufficiently large. Most users would arguably not mind being allocated slightly fewer resources at some time periods if they are compensated at other times. This is especially true if jobs have deadlines; a user might be willing to accept a slightly more "unfair" allocation at some time period if it means the user's job finishes earlier overall. In addition, one may wish to take into consideration the system requirements. In a software company, for example, if the system requires a few more seconds to complete the execution of an important program, and 100 agents join, the system might prefer to delay the agents' entry into the system for a short while (here, "short" needs to be defined to allay the concerns raised by Isard et al. [21]). Under our definition, this system would have a fairness ratio of zero. This example brings us to another direction for future research: balancing fairness and social welfare (the sum of the agents' utilities). Bertsimas et al. [5] study system efficiency loss under "fair" allocations compared with ones that maximize the sum of player utilities in static settings. It would be interesting to quantify the efficiency loss caused by fairness requirements in dynamic settings such as ours as well.

Li et al. [24] extend our results to heterogeneous users; however, many open questions remain, such as bounding the fairness ratio for c-gap control vectors, $c > 1$ in this setting. Another important extension is removing the assumption that there is a single resource as many real-world systems typically consist of multiple resources (e.g., CPU, RAM, disk space, and I/O resources), which users require in different proportions. We briefly address this question in Friedman et al. [16], but the algorithms and bounds therein are far from optimal. Finally, the resource in our model is infinitely divisible, and users have additive valuations and utility that is linear in the amount of resource allocated to them. Although these are often standard assumptions in the fair division literature, they are clearly simplifications, and it would be interesting to extend our results to models in which one or both of these assumptions does not hold.

Acknowledgment

The authors thank the reviewers at *Mathematics of Operation Research* for their exceptional reviews and suggestions and reviewers of the conference and previous versions of this paper for their useful feedback.

Appendix A. Summary of Results

We summarize our main results in Table A.1.

Table A.1. Results for single resource dynamic fair division. H_n denotes the n-th harmonic number.

Control vectors	Bound on the fairness ratio
$d-$ uniform, $d \geq 1$	$\left((d+1)\ln\left(\frac{d+1}{d}\right)\right)^{-1}$ (tight)
$1-$ gap	$2(3\ln 3)^{-1}$ (tight)
$2-$ gap	$3(4\ln 4)^{-1}$ (tight)
$c-$ gap, $c \geq 3$	$(H_{c+1})^{-1} \geq FAIRNESS(c-\text{gap}) \geq \left(H_{2c+3}-\frac{1}{2}\right)^{-1}$ $(c+1)((c+2)\ln(c+2))^{-1}$ (asymptotic bound, tight)

Appendix B. Figures

Figure B.1. (Color online) Allocations of FFA for the basic 3-gap control vectors with $\sigma = 1$.

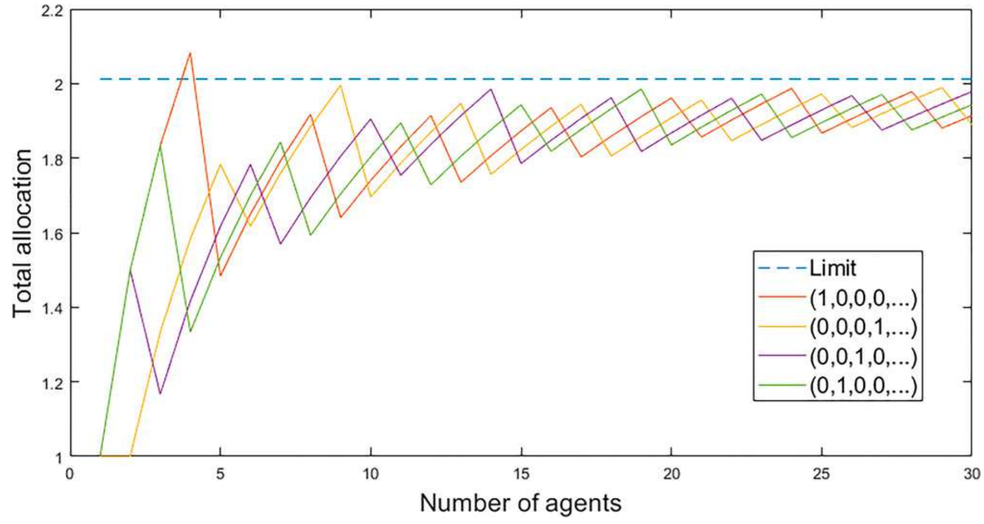


Figure B.2. (Color online) Allocations of FFA for 1-uniform control vectors and $\sigma = 1$.

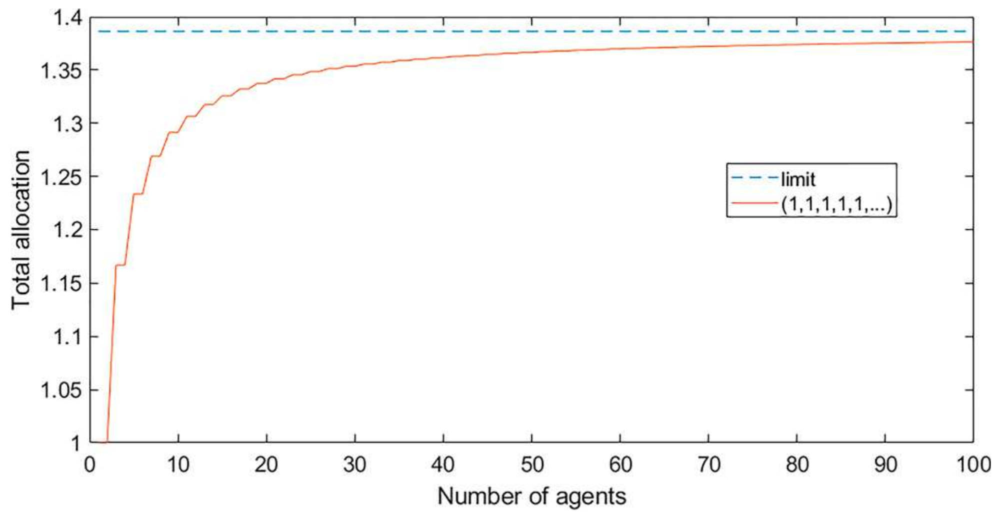
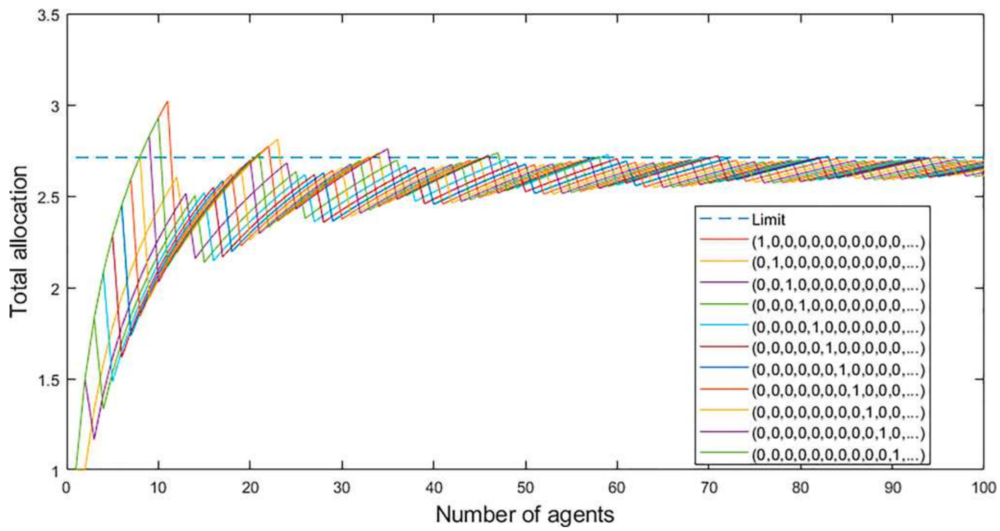


Figure B.3. (Color online) Allocations of FFA for the basic 10-gap control vectors with $\sigma = 1$.



Appendix C. Number Theory

We use the following well-known bounds on the *nth* harmonic number H_n (Guo and Qi [17]):

$$\frac{1}{2n+1} \le H_n - \ln n - \gamma \le \frac{1}{2n}, \tag{C.1}$$

where $\gamma = 0.57721\ldots$ is the Euler–Mascheroni constant. Using these, it is straightforward to derive the following:

Lemma C.1. *For any natural numbers $b > a > 1$,*

$$\ln(b) - \ln(a-1) - \frac{1}{2a-2} \le \sum_{x=a}^{b} \frac{1}{x} \le \ln(b) - \ln(a-1) + \frac{1}{2b} - \frac{1}{2a-2}.$$

Lemma C.2. *The function*

$$f(n) = \sum_{i=n}^{(n-1)(c+2)+c+1} \frac{1}{i}$$

is monotone decreasing in n for integer values of n and integer c > 0.

Proof.

$$\begin{aligned} f(n) - f(n+1) &= \sum_{i=n}^{(n-1)(c+2)+c+1} \frac{1}{i} - \sum_{i=n+1}^{n(c+2)+c+1} \frac{1}{i} \\ &= \frac{1}{n} - \sum_{i=n(c+2)}^{n(c+2)+c+1} \frac{1}{i} \\ &> \frac{1}{n} - (c+2)\frac{1}{n(c+2)} = 0. \quad \square \end{aligned}$$

Lemma C.3. *For integer $c > 1$ and integer $n \geq 2$,*

$$\sum_{i=0}^{n-2} \frac{1}{n + i(c+1)} \leq 1/2.$$

Proof. It suffices to prove the lemma for $c = 2$ as the sum decreases as c increases.

$$\begin{aligned} \sum_{i=0}^{n-2} \frac{1}{n+3i} &= \frac{1}{n} + \sum_{i=1}^{n-2} \frac{1}{n+3i} \\ &\leq \frac{1}{n} + \int_0^{n-2} \frac{1}{n+3x} dx \\ &= \frac{1}{n} + \frac{1}{3}\ln(4 - 6/n) \\ &\leq \frac{1}{n} + \frac{1}{3}\ln 4 \\ &\leq 1/2, \end{aligned}$$

for $n \geq 27$. It is easy to computationally verify the result holds for smaller n. □

Appendix D. Supplementary Material for Section 3

Lemma 1 (Reproduced Here for Clarity). *The allocations produced by FFA(ψ,σ) as inputs are identical to those produced by FFA(ψ,σ'), scaled by $\frac{\sigma}{\sigma'}$.*

Proof. Assume that the tie-breaking rule in both cases is identical; that is, if there are several users that have a maximal share and a donor needs to be selected from them, the algorithm chooses the same donor for both inputs.

Denote the allocations produced by FFA when given ψ and σ as inputs by $X^1, \ldots, X^{|\psi|}$ and when given ψ and σ' as inputs by $Z^1, \ldots, Z^{|\psi|}$. The proof is by induction. For the base, $X_1^1 = \sigma, Z_1^1 = \sigma'$; hence, $X_1^1 = \sigma' \frac{\sigma}{\sigma'}$. For the inductive step, assume that the claim holds for X^{t-1} and Z^{t-1}. The claim immediately holds for X^t and Z^t for any user for which the allocation does not change. The users for whom the allocations *do* change as well as the arriving user are allocated $\frac{\sigma}{t}$ and $\frac{\sigma'}{t}$ in X and Z, respectively. This completes the proof. □

Appendix E. Supplementary Material for Section 5.1

E.1. Proof of Proposition 2

Proposition 2 (Reproduced Here for Clarity). *For any control vectors ψ and ψ' such that ψ' that is a prefix of ψ,*

$$\text{FAIRNESS}(\psi) \leq \text{FAIRNESS}(\psi').$$

Proof. Denote $\sigma = \text{FAIRNESS}(\psi)$ and $\sigma' = \text{FAIRNESS}(\psi')$. Let $S(\psi,\sigma)^t$ denote the total resource allocated by FFA when it is executed with ψ and σ when there are t users present. Toward a contradiction, assume that $\sigma > \sigma'$. As $\sigma' = \text{FAIRNESS}(\psi')$ and ψ' is finite, there exists some τ such that $S(\psi',\sigma')^\tau = 1$. Note that the first $|\psi'|$ allocations when executing FFA with (i) ψ and σ' and (ii) ψ' and σ' are identical. Therefore, by Lemma 1, $S(\psi,\sigma)^t = \sigma' \frac{\sigma}{\sigma'} > 1$. This is a contradiction to $\sigma = \text{FAIRNESS}(\psi)$. □

E.2. Proof of Theorem 4

For the proof of Theorem 4, we require the following lemma.

Lemma E.1. *Let* X^t *denote the allocation of* FFA *when there are* t *users in the system when* FFA's *inputs are the infinite* d*-uniform control vector and* σ*. For all* $t = \ell \cdot (d+1) + i$*, integer* $\ell \geq 0$*, integer* $i \in [0, d]$*,*

$$X^t = \Bigg(\underbrace{\frac{\sigma}{t-\ell}, \ldots, \frac{\sigma}{t-\ell}}_{i\text{ terms}}, \underbrace{\frac{\sigma}{t-\ell+1}, \ldots, \frac{\sigma}{t-\ell+1}}_{d+1\text{ terms}}, \ldots, \underbrace{\frac{\sigma}{t-1}, \ldots, \frac{\sigma}{t-1}}_{d+1\text{ terms}}, \underbrace{\frac{\sigma}{t}, \ldots, \frac{\sigma}{t}}_{d+1\text{ terms}}\Bigg).$$

Proof. The proof is by induction on ℓ. For the base case, $\ell = 0$, that is, $t = i$, $i \in [0, d]$, there are at least as many donors as users, and so the allocation is

$$X^i = \Bigg(\underbrace{\frac{\sigma}{t-\ell}, \ldots, \frac{\sigma}{t-\ell}}_{i\text{ terms}}\Bigg).$$

For the inductive step, assume the statement holds for some ℓ and all $i \in [0, d]$. Specifically, for $i = d$, we have

$$\begin{aligned} X^{\ell(d+1)+d} &= \Bigg(\underbrace{\frac{\sigma}{t-\ell}, \ldots, \frac{\sigma}{t-\ell}}_{d\text{ terms}}, \underbrace{\frac{\sigma}{t-\ell+1}, \ldots, \frac{\sigma}{t-\ell+1}}_{d+1\text{ terms}}, \ldots, \underbrace{\frac{\sigma}{t-1}, \ldots, \frac{\sigma}{t-1}}_{d+1\text{ terms}}, \underbrace{\frac{\sigma}{t}, \ldots, \frac{\sigma}{t}}_{d+1\text{ terms}}\Bigg) \\ &= \Bigg(\underbrace{\frac{\sigma}{\ell d+d}, \ldots, \frac{\sigma}{\ell d+d}}_{d\text{ terms}}, \underbrace{\frac{\sigma}{\ell d+d+1}, \ldots, \frac{\sigma}{\ell d+d+1}}_{d+1\text{ terms}}, \ldots, \underbrace{\frac{\sigma}{\ell(d+1)+d}, \ldots, \frac{\sigma}{\ell(d+1)+d}}_{d+1\text{ terms}}\Bigg). \end{aligned}$$

When the next user arrives, FiniteFFA disrupts the d users with the largest shares, so the next allocations are

$$\begin{aligned} X^{(\ell+1)(d+1)} &= \Bigg(\underbrace{\frac{\sigma}{\ell d+d+1}, \ldots, \frac{\sigma}{\ell d+d+1}}_{d+1\text{ terms}}, \ldots, \underbrace{\frac{\sigma}{\ell(d+1)+d+1}, \ldots, \frac{\sigma}{\ell(d+1)+d+1}}_{d+1\text{ terms}}\Bigg) \\ &= \Bigg(\underbrace{\frac{\sigma}{(\ell+1)(d+1)-\ell}, \ldots, \frac{\sigma}{(\ell+1)(d+1)-\ell}}_{d+1\text{ terms}}, \ldots, \underbrace{\frac{\sigma}{(\ell+1)(d+1)}, \ldots, \frac{\sigma}{(\ell+1)(d+1)}}_{d+1\text{ terms}}\Bigg), \end{aligned}$$

as required. It is now straightforward to confirm that the characterization holds for $i \geq 1$ because d of the first $d+1$ terms are disrupted, and $d+1$ new terms are added at the end. □

Proof of Theorem 4. Let X^t denote allocation of FFA on input the infinite d-uniform control vector and σ when there are t users in the system. Denote the total amount of resource allocated by $S^t = \sum_{i=1}^t X_i^t$. We first show that S^t increases with t. Let $t = \ell \cdot (d+1) + i$ for some ℓ and i. Then, from Lemma E.1,

$$S^t = i \cdot \frac{\sigma}{t-\ell} + (d+1) \cdot \sum_{k=1}^{\ell} \frac{\sigma}{t-\ell+k} = i \cdot \frac{\sigma}{\ell d+i} + (d+1) \cdot \sum_{k=1}^{\ell} \frac{\sigma}{\ell d+i+k}.$$

Let $t' = \ell \cdot (d+1) + i + 1$.

$$\begin{aligned} S^{t'} - S^t &= \left(\frac{\sigma(i+1)}{\ell d+i+1} + (d+1) \cdot \sum_{k=1}^{\ell} \frac{\sigma}{\ell d+i+1+k}\right) - \left(i \cdot \frac{\sigma}{\ell d+i} + (d+1) \cdot \sum_{k=1}^{\ell} \frac{\sigma}{\ell d+i+k}\right) \\ &= \frac{(d+1)\sigma}{\ell d+i+1+\ell} - i \cdot \frac{\sigma}{\ell d+i} - \frac{\sigma((d+1)-(i+1))}{\ell d+i+1} \\ &= \sigma \frac{d-i}{(\ell d+i)(\ell d+i+1)(\ell d+i+\ell+1)}, \end{aligned}$$

which is nonnegative for all $i \leq d$. Therefore, $S^{(\ell+1)(d+1)} \geq S^{\ell(d+1)+i}$ for all $i \in [0, d]$. The same argument shows that $S^{\ell(d+1)+1} \geq S^{\ell(d+1)}$. Thus, it suffices to bound S^t at the limit. Because the sequence S^t is monotone, having a convergent subsequence is equivalent to the sequence being convergent, and further, the limits are the same. Hence, we analyze $t = \ell(d+1)$ without loss of generality.

$$\begin{aligned} \lim_{\ell\to\infty} S^{\ell(d+1)} &= \lim_{\ell\to\infty} \sum_{k=1}^{\ell} \frac{\sigma(d+1)}{\ell d+k} \\ &= \sigma(d+1) \cdot \lim_{\ell\to\infty} \sum_{k=\ell d+1}^{\ell(d+1)} \frac{1}{k} \\ &= \sigma(d+1) \cdot \lim_{\ell\to\infty} (H_{\ell(d+1)} - H_{\ell d}) \\ &= \sigma(d+1) \cdot \lim_{\ell\to\infty} (\ln(\ell(d+1)) - \ln(\ell d)) \\ &= \sigma(d+1) \cdot \ln\left(\frac{d+1}{d}\right), \end{aligned} \tag{E.1}$$

where Equality (E.1) is due to the well-known limit $\lim_{n\to\infty}(H_n - \ln n) = \gamma$, where γ is the Euler–Mascheroni constant. As S^t is increasing in t, we set $\lim_{\ell\to\infty} S^{\ell(d+1)} = 1$. From Proposition 2, this characterizes the infimum of the fairness ratio of all d-uniform control vectors. □

E.3. Proof of Corollary 1

Corollary 1 (to Theorem 4, Reproduced Here for Clarity). $\text{FAIRNESS}(d-\mathit{uniform}) \to_{d\to\infty} 1$.

Proof. From Theorem 4, it holds that $\text{FAIRNESS}(d-\mathit{uniform}) = \frac{1}{(d+1)\ln\left(\frac{d+1}{d}\right)}$ for any $d \geq 1$. The Laurent series expansion of $\frac{1}{(d+1)\ln\left(\frac{d+1}{d}\right)}$ at $d=\infty$ is $1 - \frac{1}{2d} + O\left(\frac{1}{d^2}\right)$. The corollary follows. □

Appendix F. Supplementary Material for Section 5.2

F.1. Proof of Theorem 7

Theorem 7 (Reproduced Here for Clarity). *The following hold:*

1. $\text{FAIRNESS}(1-gap) = 2(3\ln 3)^{-1}$,
2. $\text{FAIRNESS}(2-gap) = 3(4\ln 4)^{-1}$.

There are two basic one-gap control vectors: $(1,0,1,0,\dots)$ and $(0,1,0,1,\dots)$. There are three basic two-gap control vectors (Example 3). We prove the bound in Theorem 7 for each basic control vector separately. The theorem then follows from Proposition 3. Here, we only present the proof for one of the two basic one-gap control vectors. We omit the proof for the other control vector as it is virtually identical. Furthermore, we only present the proof for $c = 1$; the proof for $c = 2$ is by similar (slightly more involved) case analysis.

Proposition F.1. *The allocations created by FFA, with inputs basic one-gap control vector $\psi^1 = (0,1,\dots)$ and $\sigma = 1$ (from step 6 onward) are*

1. *At time $t = 0 \pmod 6$:* $\frac{1}{(t/3)+1}, \frac{1}{(t/3)+2}, \frac{1}{(t/3)+2}, \frac{1}{(t/3)+3}, \dots, \frac{1}{t-2}, \frac{1}{t-2}, \frac{1}{t-1}, \frac{1}{t}, \frac{1}{t}$.
2. *At time $t = 2 \pmod 6$:* $\frac{1}{((t+1)/3)+1}, \frac{1}{((t+1)/3)+2}, \frac{1}{((t+1)/3)+2}, \dots, \frac{1}{t}, \frac{1}{t}$.
3. *At time $t = 4 \pmod 6$:* $\frac{1}{((t+2)/3)}, \frac{1}{((t+2)/3)+1}, \frac{1}{((t+2)/3)+2}, \frac{1}{((t+2)/3)+2}, \dots, \frac{1}{t}, \frac{1}{t}$.

On odd time steps t, add $\frac{1}{t}$ to the previous step (note the denominators have "changed" relative to the new time t).

Proof. The proof is by induction on the time t. The base case (the first six time steps) appears in Table F.1. The move from even to odd steps is immediate as there is no donor. Because only the user with the highest utility has the user's allocation decreased, it is easy to verify the transition from odd to even steps by renaming the denominators. For example, in the inductive step, for the case of $t = 0 \pmod 6$, we know that $t-1 = 5 \pmod 6$, and by the inductive hypothesis, the allocation on step $t-1$ is equal to $\frac{1}{(t/3)}, \frac{1}{(t/3)+1}, \frac{1}{(t/3)+2}, \frac{1}{(t/3)+2}, \frac{1}{(t/3)+3}, \frac{1}{(t/3)+4}, \frac{1}{(t/3)+4}, \dots, \frac{1}{t-2}, \frac{1}{t-2}, \frac{1}{t-1}$ (we get this by plugging in $t = t-2$ in the "$t = 4 \pmod 6$" allocation and adding $\frac{1}{t-1}$ at the end). At step t, we have a disruption; therefore, the leading term $\frac{1}{(t/3)}$ is disrupted, and we have two additional $\frac{1}{t}$ terms. The remaining cases are virtually identical. □

In order to compute the optimal fairness ratio of $\psi^1 = (0,1,0,1,\dots)$, we bound the sum of allocations at odd steps because each odd step uses strictly more resources than the previous even step.

Denote by $S(\psi^1,t)$ the total resource allocated by $\text{FFA}(\psi^1,1)$ when there are t users in the system. We bound $1 - S(\psi^1,t)$, noting, for all $t > 0$, $1 - S(\psi^1,t) < 1$. On odd steps, we take $\frac{1}{i}$ from the bank (the unallocated resource). On all even steps except the second and fourth, we return some resource to the bank. (Note that, to be able to reach step 5, we

Table F.1. Allocations for the first six steps of FFA with basic one-gap control vectors and some σ as input.

Step	Allocation for $(0,1,\dots)$	**Sum**	**Allocation for** $(1,0,\dots)$	**Sum**
1	σ	σ	σ	σ
2	$\frac{\sigma}{2}, \frac{\sigma}{2}$	σ	$\sigma, \frac{\sigma}{2}$	$\frac{90\sigma}{60}$
3	$\frac{\sigma}{2}, \frac{\sigma}{2}, \frac{\sigma}{3}$	$\frac{80\sigma}{60}$	$\frac{\sigma}{2}, \frac{\sigma}{3}, \frac{\sigma}{3}$	$\frac{70\sigma}{60}$
4	$\frac{\sigma}{2}, \frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{4}$	$\frac{80\sigma}{60}$	$\frac{\sigma}{2}, \frac{\sigma}{3}, \frac{\sigma}{3}, \frac{\sigma}{4}$	$\frac{85\sigma}{60}$
5	$\frac{\sigma}{2}, \frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{4}, \frac{\sigma}{5}$	$\frac{92\sigma}{60}$	$\frac{\sigma}{3}, \frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{5}, \frac{\sigma}{5}$	$\frac{79\sigma}{60}$
6	$\frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{4}, \frac{\sigma}{5}, \frac{\sigma}{6}, \frac{\sigma}{6}$	$\frac{82\sigma}{60}$	$\frac{\sigma}{3}, \frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{5}, \frac{\sigma}{5}, \frac{\sigma}{6}$	$\frac{89\sigma}{60}$

need $\frac{23}{15}$ in the bank; this immediately implies that $\sigma \le \frac{15}{23}$.) First, we show that the resource allocated monotonically increases when we look at it through a slightly wider lens.

Lemma F.1. *Denote by* $S(\psi^1, t)$ *the total resource allocated by* $FFA(\psi^1, 1)$ *when there are* t *users in the system. For any* $t = 0 \pmod 6$, $S(\psi^1, t+5) = \max_{i \in \{0,\dots,5\}} S(\psi^1, t+i)$. *Furthermore,* $S(\psi^1, t+5)$ *is monotone nondecreasing in* t *for* $t = 0 \pmod 6$.

Proof. Let $t = 0 \pmod 6$. It is easy to verify the following using Proposition F.1:

1. At the $t+1$-th arrival, we add $\frac{1}{t+1}$ to $S(\psi^1, t)$.
2. At the $t+2$-th arrival, we add $\frac{2}{t+2}$ and subtract $\frac{1}{t/3+1}$ from $S(\psi^1, t+1)$.
3. At the $t+3$-th arrival, we add $\frac{1}{t+3}$ to $S(\psi^1, t+2)$.
4. At the $t+4$-th arrival, we add $\frac{2}{t+4}$ and subtract $\frac{1}{t/3+2}$ from $S(\psi^1, t+3)$.
5. At the $t+5$-th arrival, we add $\frac{1}{t+5}$ to $S(\psi^1, t+4)$.

Clearly, $S(\psi^1, t+5) \ge S(\psi^1, t+4)$, $S(\psi^1, t+3) \ge S(\psi^1, t+2)$, and $S(\psi^1, t+1) \ge S(\psi^1, t)$. It is easy to verify using simple calculus that $S(\psi^1, t+5) = S(\psi^1, t+4) + \frac{1}{t+5} = S(\psi^1, t+3) + \frac{2}{t+4} - \frac{1}{t/3+2} + \frac{1}{t+5}$ is larger than $S(\psi^1, t+3)$ for all t. Also, $S(\psi^1, t+5) = S(\psi^1, t+1) + \frac{1}{t+3} + \frac{2}{t+2} - \frac{1}{t/3+1} + \frac{2}{t+4} - \frac{1}{t/3+2} + \frac{1}{t+5} \ge S(\psi^1, t+1)$. Therefore, $X(\psi^1, t+5)$ is the allocation with the maximum resource out of the six allocations $S(\psi^1, t+i) : i \in \{0, \dots, 5\}$.

Furthermore, $S(\psi^1, t+5) = S(\psi^1, t-1) + \frac{1}{t+1} + \frac{2}{t} - \frac{1}{t/3} + \frac{1}{t+3} + \frac{2}{t+2} - \frac{1}{t/3+1} + \frac{2}{t+4} - \frac{1}{t/3+2} + \frac{1}{t+5}$ is greater than $S(\psi^1, t-1)$ for all $t \ge 2$. Because $t - 1 = 5 \pmod 6$, this implies that $S(\psi^1, t)$ for $t = 5 \pmod 6$ is monotone nondecreasing. □

We now complete the proof of Theorem 7 (for the control vector $(0, 1, 0, 1, 0, \dots)$).

Proof of Theorem 7. From Lemma F.1, the amount of resource allocated increases with t; therefore, it suffices to bound the total allocation when t goes to infinity. Similarly, it suffices to analyze the case when the allocation is larger than all previous allocations, $t = 5 \pmod 6$. At time $t = 0 \pmod 6$, the total resource allocated is

$$
\begin{aligned}
&2\sum_{i=1}^{t/3} \frac{1}{t/3 + 2i} + \sum_{i=1}^{t/3} \frac{1}{t/3 + 2i - 1} \\
&= \frac{1}{2}\left(4\sum_{i=1}^{t/3} \frac{1}{t/3 + 2i} + 2\sum_{i=1}^{t/3} \frac{1}{t/3 + 2i - 1}\right) \\
&\le \frac{1}{2}\left(3\sum_{i=1}^{t/3} \frac{1}{t/3 + 2i} + 3\sum_{i=1}^{t/3} \frac{1}{t/3 + 2i - 1}\right) \\
&= \frac{3}{2} \sum_{i=t/3+1}^{t} \frac{1}{i} \\
&\le \frac{3\ln(3)}{2} - \frac{1}{t},
\end{aligned}
$$

where the last inequality is due to Lemma C.1. As $S(\psi^1, t) = S(\psi^1, t-1) + \frac{2}{t} - \frac{1}{t/3}$ for $t = 0 (\bmod 6)$, it holds that the allocations at times $t = 5 (\bmod 6)$ are at most $\frac{3\ln(3)}{2} - \frac{1}{t} - \frac{2}{t} + \frac{1}{t/3} = \frac{3\ln(3)}{2}$.

The proof for the matching lower bound is similar, using the other inequality of Lemma C.1, and is omitted. □

F.2. Proof of Proposition 3

Proposition 3 (Reproduced Here for Clarity). *For every (finite or infinite) c-gap control vector* ψ, *there exists some (infinite) basic c-gap control vector* $\hat{\psi}$ *such that* $\text{FAIRNESS}(\psi) \ge \text{FAIRNESS}(\hat{\psi})$.

Proof. We define a (possibly infinite) series of control vectors $\psi_1, \psi_2, \dots$, where $\hat{\psi} = \psi_1$ is a basic c-gap control vector such that their fairness ratio is nondecreasing. In other words, for any $i \ge 1$, if all of the allocations of $\text{FFA}(\psi_i, \sigma)$ are feasible, then the allocations of $\text{FFA}(\psi_{i+1}, \sigma)$ are also feasible. We define these vectors inductively. ψ_1 is a basic c-gap control vector that shares the longest prefix with ψ. For $i > 1$, let β_i be the first (leftmost) coordinate on which ψ_i and ψ differ. For $j \in [1, \beta_i - 1]$, set $\psi_{i+1}[j] = \psi_i[j]$; set $\psi_{i+1}[\beta_i] = \psi[\beta_i]$, and append $(0^c1)^\infty$. If ψ is finite, we append $(0^c1)^\infty$ to it as well. The series is defined this way if ψ is finite or can be represented as a finite vector with $(0^c1)^\infty$ appended to it. Otherwise, it is infinite.

We show that the fairness ratio of $\psi_1, \psi_2, \dots$ is nonincreasing. Note that, if ψ is finite, there is no k for which $\psi_k = \psi$, but there is a k such that $\psi_k = \psi(0^c1)^\infty$: the last control vector in the series. By Proposition 2, $\text{FAIRNESS}(\psi) \ge \text{FAIRNESS}(\psi_k)$.

In Lemma F.2, we show that, for every i and all steps t such that $t = \beta_i \pmod{c+1}$, if $\text{FFA}(\psi_i, \sigma)$ is feasible, then so is $\text{FFA}(\psi_{i+1}, \sigma)$. In Lemma F.4, we show that $\text{FFA}(\psi_{i+1}, \sigma)$ remains feasible for all other time steps ($t \ne \beta_i \pmod{c+1}$). The proposition follows from these two lemmas as together they cover all time steps. □

Fix σ and let $\hat{\psi} = \psi_1, \psi_2, \dots$, be as in the proof of Proposition 3. Fix σ and denote the set of allocations of $\text{FFA}(\psi, \sigma)$ by $\mathcal{X}(\psi)$. Notice that $\mathcal{X}(\psi_i)$ and $\mathcal{X}(\psi_{i+1})$, the allocation sets of $\text{FFA}(\psi_i, \sigma)$ and $\text{FFA}(\psi_{i+1}, \sigma)$, are identical up to step $\beta_i - 1$.

Then, on step β_i, necessarily $\psi_i[\beta_i]=0$, $\psi_{i+1}[\beta_i]=1$. Let $\mathcal{X}(\psi,t)$ denote the allocation produced by FFA with inputs ψ and σ at time t (when there are $t-1$ agents in the system and one more arrives). Let $|\mathcal{X}(\psi,t)|$ denote its magnitude (i.e., how much resource is allocated at this time).

Lemma F.2. *For all* $i\geq 1$ *and all* $t=\beta_i \pmod{c+1}$,

$$\mathcal{X}(\psi_i,t)\succcurlyeq\mathcal{X}(\psi_{i+1},t).$$

Proof. Recall that β_i is the leftmost coordinate on which ψ_i and ψ differ and, hence, the leftmost coordinate on which ψ_i and ψ_{i+1} differ. If $t<\beta_i$, the proposition trivially holds because the two allocations are identical up until step β_i-1. We prove the claim for $t=\beta_i+j(c+1)$ by induction on j. For the base case, $j=0$, denote $\mathcal{X}(\psi_i,\beta_i-1)=\mathcal{X}(\psi_{i+1},\beta_i-1)=(a_1,a_2,\dots,a_{\beta_i-1})$.

Because $\psi_i[\beta_i]=0$ and $\psi_{i+1}[\beta_i]=1$, we have

$$\mathcal{X}(\psi_i,\beta_i)=(a_1,a_2,\dots,a_{\beta_i-2},a_{\beta_i-1},\sigma/\beta_i),$$
$$\mathcal{X}(\psi_{i+1},\beta_i)=(a_2,a_3,\dots,a_{\beta_i-1},\sigma/\beta_i,\sigma/\beta_i).$$

As $\mathcal{X}(\psi_i,\beta_i-1)$ is sorted (hence, $a_\ell\geq a_{\ell+1}$ for all ℓ), and $a_{\beta_i-1}=\frac{\sigma}{\beta_i-1}>\frac{\sigma}{\beta_i}$, this completes the proof of the base case.

For the inductive step, let $\mathcal{X}(\psi_i,t)=(a_1,a_2,\dots,a_t)$ and $\mathcal{X}(\psi_{i+1},t)=(b_1,b_2,\dots,b_t)$. By the inductive hypothesis $\mathcal{X}(\psi_i,t)\succcurlyeq\mathcal{X}(\psi_{i+1},t)$, that is, $a_i\geq b_i$, for all $i=1,\dots,t$. We show that $\mathcal{X}(\psi_i,t+c+1)\succcurlyeq\mathcal{X}(\psi_{i+1},t+c+1)$.

Because $t=\beta_i+j(c+1)$, we know that $\psi_i[t]=0$, $\psi_i[t+c+1]=0$, and $\psi_{i+1}[t]=1$. Furthermore, $\psi_{i+1}[t']=0$, for $t<t'<t+c+1$. Let q_i be the smallest value such that $\psi_i[t+q_i]=1$.

$$\mathcal{X}(\psi_i,t)=(a_1,\dots,a_{t-1},a_t),$$
$$\mathcal{X}(\psi_i,t+q_i)=\left(a_2,\dots,a_t,\frac{\sigma}{t+1},\dots,\frac{\sigma}{t+q_i-1},\frac{\sigma}{t+q_i},\frac{\sigma}{t+q_i}\right),$$
$$\mathcal{X}(\psi_i,t+c+1)=\left(a_2,\dots,a_t,\frac{\sigma}{t+1},\dots,\frac{\sigma}{t+q_i-1},\frac{\sigma}{t+q_i},\frac{\sigma}{t+q_i},\frac{\sigma}{t+q_i+1},\dots,\frac{\sigma}{t+c+1}\right).$$

And

$$\mathcal{X}(\psi_{i+1},t)=(b_1,\dots,b_{t-1},b_t),$$
$$\mathcal{X}(\psi_{i+1},t+c+1)=\left(b_2,\dots,b_t,\frac{\sigma}{t+1},\dots,\frac{\sigma}{t+c},\frac{\sigma}{t+c+1},\frac{\sigma}{t+c+1}\right).$$

For the first $t-1$ terms, $a_i\geq b_i$ by the inductive hypothesis. The next q_i terms are identical ($\frac{\sigma}{t+1}$ through $\frac{\sigma}{t+q_i}$) as well as the very last term. □

Proving Lemma F.4—that, if FFA(ψ_i,σ) is feasible, then FFA(ψ_{i+1},σ) is feasible, for steps $t\neq\beta_i \pmod{c+1}$—is more involved. First, we show—in Lemma F.3—that there exists some T_{ψ_i} for which, for every $t\geq T_{\psi_i}$, $\mathcal{X}(\psi_{i+1},t)$ is identical to $\mathcal{X}(\psi_i,t)$. Therefore, we only need to consider allocations prior to that T_{ψ_i}.

Lemma F.3. *For all* $i\geq 1$, *there exists some (minimal)* T_{ψ_i} *such that, for all* $t>T_{\psi_i}$, $\mathcal{X}(\psi_{i+1},t)=\mathcal{X}(\psi_i,t)$.

Proof. Consider what the allocation of a basic c-gap control vector looks like. Every $c+1$ arrivals, a disruption is allowed, so the "bulk" of the allocation is c single terms of the form $\left(\frac{\sigma}{j},\frac{\sigma}{j+1},\dots,\frac{\sigma}{j+c}\right)$, followed by two terms $\left(\frac{\sigma}{j+c+1},\frac{\sigma}{j+c+1}\right)$, followed by c singles, followed by a double, and so on. The leading terms (until the first double term) could be any number of singles between zero and $c+1$. (There could be exactly $c+1$ single terms in the following case: the first term of the allocation in the previous step was a double term (followed by c singles) and the next step had a disruption.) Similarly, the last terms could be any number of singles between zero and $c+1$. Knowing the specific basic c-gap control vector essentially only determines the steps in which the double terms appear.

Let p be number of zeros between β_i and the previous one. In Example 4, for ψ_4, $p=1$. Then, $\mathcal{X}(\psi_{i+1},\beta_i)=(a_1,\dots,a_m,\frac{\sigma}{\beta_i-p-1},\frac{\sigma}{\beta_i-p-1},\frac{\sigma}{\beta_i-p},\frac{\sigma}{\beta_i-p+1},\dots,\frac{\sigma}{\beta_i-1},\frac{\sigma}{\beta_i},\frac{\sigma}{\beta_i})$. T_{ψ_i} is exactly the smallest time in which the first term of $\mathcal{X}(\psi_{i+1},T_{\psi_i})$ is $\frac{\sigma}{\beta_i-p-1}$, and the second term is $\frac{\sigma}{\beta_i-p}$, that is, only the first one of the $\frac{\sigma}{\beta_i-p-1}$ terms in $\mathcal{X}(\psi_{i+1},\beta_i)$ has been disrupted. So we'd like for the first $m+1$ terms of the β_i terms to be disrupted. Solving for m, we get that $m=\beta_i-p-4$ (we subtract the two double terms $\frac{\sigma}{\beta_i}$ and $\frac{\sigma}{\beta_i-p-1}$ and the p single terms); hence, $T_{\psi_i}=\beta_i+(c+1)(\beta_i-p-3)$. □

In order to show that FFA(ψ_{i+1},σ) is feasible, we want to show that the total resource allocated in $\mathcal{X}(\psi_{i+1},t)$ for all $t\in[\beta_i,T_{\psi_i}]$, $t\neq\beta_i \pmod{c+1}$ is at most one. Equivalently, if it suffices to show that, for all steps when a donor is used, there is enough unallocated resource to handle c consecutive nondisruptive steps.

Lemma F.4. *For* $i\geq 1$, *let* T_{ψ_i} *be as in Lemma F.3. For all* $t=\beta_i \pmod{c+1}$, $t\in[\beta_i,T_{\psi_i}]$,

$$1-|\mathcal{X}(\psi,t)|\geq\sum_{j=1}^{c}\frac{\sigma}{t+j}.$$

Proof. Let $t = \beta_i + \ell(c+1)$. Let q_i be the smallest positive integer such that $\psi_i^N[\beta_i + q_i] = 1$. Clearly, $q_i \le c$ as ψ has at most c consecutive zeros. Note further that, because $\psi_i^N[\beta_i + q_i] = 1$ and because $\beta_i \ge 2$, it must hold that $\beta_i + q_i \ge c + 2$.

Observe that, if $\mathcal{X}(\psi_i^N, \beta_i) = (a_1, a_2, \ldots, a_{\beta_i-2}, a_{\beta_i-1}, \sigma/\beta_i)$, then

$$\mathcal{X}(\psi_i^N, \beta_i + q_i) = \left(a_2, \ldots, a_{\beta_i-2}, a_{\beta_i-1}, \frac{\sigma}{\beta_i}, \frac{\sigma}{\beta_i+1}, \ldots, \frac{\sigma}{\beta_i+q_i-1}, \frac{\sigma}{\beta_i+q_i}, \frac{\sigma}{\beta_i+q_i}\right),$$
$$\mathcal{X}(\psi_{i+1}^N, \beta_i) = \left(a_2, \ldots, a_{\beta_i-2}, a_{\beta_i-1}, \frac{\sigma}{\beta_i}, \frac{\sigma}{\beta_i}\right).$$

After $c+1$ more steps, the leading term a_2 appears in neither $\mathcal{X}(\psi_i^N, \beta_i + q_i + c + 1)$ nor $\mathcal{X}(\psi_{i+1}^N, \beta_i + c + 1)$. The first allocation has additional terms $\frac{1}{\beta_i+q_i+1}$ through $\frac{1}{\beta_i+q_i+c}$ and two terms $\frac{1}{\beta_i+q_i+c+1}$. The latter allocation has additional terms $\frac{1}{\beta_i+1}$ through $\frac{1}{\beta_i+c}$ and two terms $\frac{1}{\beta_i+c+1}$. A similar pattern appears $c+1$ steps later, and so on. For ease of notation, denote the total resource allocated by $\mathcal{X}(\psi, t)$ (i.e., $|\mathcal{X}(\psi, t)|$) by $\mathcal{S}(\psi, t)$. At step t, we have (for some k)

$$\mathcal{S}(\psi_{i+1}^N, t) = a_k + \ldots + a_{\beta_i-1} + \underbrace{\left(\frac{1}{\beta_i+1} + \frac{1}{\beta_i+2} + \ldots + \frac{1}{t}\right)}_{\text{singleterms'' plusoneofthetwo doubleterms''}} + \underbrace{\left(\frac{1}{\beta_i} + \frac{1}{\beta_i+c+1} + \ldots + \frac{1}{t}\right)}_{\text{remaining doubleterms''}}$$
$$\mathcal{S}(\psi_i^N, t+q_i) = a_k + \ldots + a_{\beta_i-1} + \sum_{j=\beta_i+1}^{t+q_i} \frac{1}{j} + \left(\frac{1}{\beta_i+q_i} + \frac{1}{\beta_i+c+1+q_i} + \ldots + \frac{1}{t+q_i}\right).$$

Their difference (which is also the difference of their "banks") is

$$\mathcal{S}(\psi_i^N, t+q_i) - \mathcal{S}(\psi_{i+1}^N, t) = \sum_{j=t+1}^{t+q_i} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right)$$
$$1 - \mathcal{S}(\psi_{i+1}^N, t) = 1 - \mathcal{S}(\psi_i^N, t+q_i) + \sum_{j=t+1}^{t+q_i} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right).$$

Because ψ_i^N is feasible, we know that $1 - \mathcal{S}(\psi_i^N, t+q_i) \ge \sum_{j=1}^{c} \frac{1}{t+q_i+j}$. Our goal is to show that $1 - \mathcal{S}(\psi_{i+1}^N, t) \ge \sum_{j=1}^{c} \frac{1}{t+j}$. To that end, we prove the following:

$$\sum_{j=1}^{c} \frac{1}{t+q_i+j} + \sum_{j=t+1}^{t+q_i} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right) \ge \sum_{j=1}^{c} \frac{1}{t+j} = \sum_{j=t+1}^{t+c} \frac{1}{j}.$$

This is equivalent to the following two inequalities, and we prove the latter.

$$\sum_{j=t+q_i+1}^{t+q_i+c} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right) \ge \sum_{j=t+q_i+1}^{t+c} \frac{1}{j}$$
$$\sum_{j=t+c+1}^{t+c+q_i} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right) \ge 0.$$

Let

$$f(\beta_i, \ell, c, q_i) = \sum_{j=t+c+1}^{t+c+q_i} \frac{1}{j} + \sum_{j=0}^{\ell}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right),$$

where $t = \beta_i + \ell(c+1)$. We know that $c \ge q_i \ge 1$, $t + q_i \ge c + 2$, and $\beta_i + \ell(c+1) \le T_{\psi_i} = \beta_i + (c+1)(\beta_i - p - 3)$, where p is the number of zeros between β_i and the previous one in ψ_N. Therefore, $p + q_i = c$, which implies that $\ell \le \beta_i + q_i - c - 3$. This is the choice of parameters for which we would like to show nonnegativity of $f(\beta_i, \ell, c, q_i)$.

As we note earlier, showing that $f(\beta_i, \ell, c, q_i)$ is nonnegative is a very delicate task. For example, using a more coarse upper bound for ℓ, such as $\ell \le \beta_i - 3$, $f(\beta_i, \ell, c, q_i)$ might be negative ($\beta_i = 9$, $\ell = 6$, $c = 4$, and $q_i = 1$). We first show that the function is nondecreasing in ℓ, in Lemma F.5. Then, because f decreases as ℓ increases, it suffices to show nonnegativity for the maximum value of ℓ, $\beta_i + q_i - c - 3$. This implies that $t = T_{\psi_i} = \beta_i + \ell(c+1) = \beta_i + (\beta_i + q_i - c - 3)(c+1) = (c+2)\beta_i + (c+1)(q_i - c - 3)$. We overload notation and redefine f as

$$f(\beta_i, c, q_i) = \sum_{j=T_{\psi_i}+c+1}^{T_{\psi_i}+c+q_i} \frac{1}{j} + \sum_{j=0}^{\beta_i+q_i-c-3}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right).$$

We show in Lemma F.6 that $f(\beta_i, c, q_i)$ is nonincreasing in c. It, therefore, remains to show that $f(\beta_i, c, q_i)$ is nonnegative in the limit. We can rewrite the function as

$$
\begin{aligned}
f(\beta_i, c, q_i) &= \sum_{j=T_{\psi_i}+c+1}^{T_{\psi_i}+c+q_i} \frac{1}{j} + \sum_{j=0}^{\beta_i+q_i-c-3}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right)\\
&= \sum_{j=T_{\psi_i}+c+1}^{T_{\psi_i}+c+q_i} \frac{1}{j} + \frac{1}{c+1}\sum_{j=0}^{\beta_i+q_i-c-3}\left(\frac{1}{\frac{\beta_i+q_i}{c+1}+j} - \frac{1}{\frac{\beta_i}{c+1}+j}\right)\\
&= \sum_{j=T_{\psi_i}+c+1}^{T_{\psi_i}+c+q_i} \frac{1}{j} + \frac{1}{c+1}\left(\sum_{j=\frac{\beta_i+q_i}{c+1}}^{\frac{T_{\psi_i}+q_i}{c+1}} \frac{1}{j} - \sum_{j=\frac{\beta_i}{c+1}}^{\frac{T_{\psi_i}}{c+1}} \frac{1}{j}\right).
\end{aligned}
$$

Applying the approximations for the harmonic number from Appendix C gives

$$
f(\beta_i, c, q_i) \geq \ln\left(\frac{T_{\psi_i}+c+q_i}{T_{\psi_i}+c}\right) - \frac{1}{2(T_{\psi_i}+c)} + \frac{1}{c+1}\ln\left(\frac{T_{\psi_i}+q_i}{\beta_i+q_i-c-1}\right) - \frac{1}{c+1}\cdot\frac{1}{2(\beta_i+q_i-c-1)} - \frac{1}{c+1}\ln\left(\frac{T_{\psi_i}}{\beta_i-c-1}\right).
$$

Taking the limit at β_i goes to infinity (recall that $T_{\psi_i} = (c+2)\beta_i + (c+1)(q_i - c - 3)$) gives

$$
\lim_{\beta_i\to\infty} f(\beta_i, c, q_i) = 0 - 0 + \frac{1}{c+1}\ln(c+2) - 0 - \frac{1}{c+1}\ln(c+2) = 0.
$$

This completes the proof of Lemma F.4. □

Lemma F.5. $f(\beta_i, \ell+1, c, q_i) - f(\beta_i, \ell, c, q_i) \leq 0.$

Proof.

$$
f(\beta_i, \ell+1, c, q_i) - f(\beta_i, \ell, c, q_i) = \sum_{j=t+2(c+1)}^{t+c+1+c+q_i} \frac{1}{j} - \sum_{j=t+c+1}^{t+c+q_i} \frac{1}{j} + \frac{1}{\beta_i+q_i+(\ell+1)(c+1)} - \frac{1}{\beta_i+(\ell+1)(c+1)}.
$$

Every positive term in the first sum is smaller than the corresponding negative term in the second sum (and they have the same number of terms). The second to last positive term is smaller than the last (negative) term. □

Lemma F.6. $f(\beta_i + c + 1, c, q_i) - f(\beta_i, c, q_i) \leq 0.$

Proof. Increasing β_i by $c + 1$ increases T_{ψ_i} by $(c+2)(c+1)$.

$$
\begin{aligned}
f(\beta_i + c+1, c, q_i) - f(\beta_i, c, q_i) = &\sum_{j=T_{\psi_i}+(c+2)(c+1)+c+1}^{T_{\psi_i}+(c+2)(c+1)+c+q_i} \frac{1}{j} - \sum_{j=T_{\psi_i}+c+1}^{T_{\psi_i}+c+q_i} \frac{1}{j} + \sum_{j=0}^{\beta_i+q_i-2}\left(\frac{1}{\beta_i+q_i+(j+1)(c+1)} - \frac{1}{\beta_i+(j+1)(c+1)}\right)\\
&- \sum_{j=0}^{\beta_i+q_i-c-3}\left(\frac{1}{\beta_i+q_i+j(c+1)} - \frac{1}{\beta_i+j(c+1)}\right).
\end{aligned}
$$

This can be rewritten as

$$
\begin{aligned}
f(\beta_i + c+1, c, q_i) - f(\beta_i, c, q_i) &= \left(\frac{1}{T_{\psi_i}+(c+1)(c+2)+c+1} + \cdots + \frac{1}{T_{\psi_i}+(c+1)(c+2)+c+q_i}\right)\\
&\quad - \left(\frac{1}{T_{\psi_i}+c+1} + \frac{1}{T_{\psi_i}+c+2} + \cdots + \frac{1}{T_{\psi_i}+c+q_i}\right)\\
&\quad + \left(\frac{1}{\beta_i+q_i+c+1} + \frac{1}{\beta_i+q_i+2(c+1)} + \cdots + \frac{1}{T_{\psi_i}+q_i+(c+1)(c+2)}\right)\\
&\quad - \left(\frac{1}{\beta_i+c+1} + \frac{1}{\beta_i+2(c+1)} + \cdots + \frac{1}{T_{\psi_i}+(c+1)(c+2)}\right)\\
&\quad - \left(\frac{1}{\beta_i+q_i} + \frac{1}{\beta_i+q_i+c+1} + \cdots + \frac{1}{T_{\psi_i}+q_i}\right) + \left(\frac{1}{\beta_i} + \frac{1}{\beta_i+c+1} + \frac{1}{\beta_i+2(c+1)} + \cdots + \frac{1}{T_{\psi_i}}\right).\\
&= \sum_{j=1}^{q_i}\left(\frac{1}{T_{\psi_i}+(c+1)(c+2)+c+j} - \frac{1}{T_{\psi_i}+c+j}\right) + \left(\sum_{j=1}^{c+2}\frac{1}{T_{\psi_i}+q_i+j(c+1)}\right) - \left(\sum_{j=1}^{c+2}\frac{1}{T_{\psi_i}+j(c+1)}\right) - \frac{1}{\beta_i+q_i} + \frac{1}{\beta_i}\\
&= -\sum_{j=1}^{q_i}\frac{(c+1)(c+2)}{\left(T_{\psi_i}+(c+1)(c+2)+c+j\right)\left(T_{\psi_i}+c+j\right)} - \sum_{j=1}^{c+2}\frac{q_i}{\left(T_{\psi_i}+(c+1)j+q_i\right)\left(T_{\psi_i}+(c+1)j\right)} + \frac{q_i}{\beta_i(\beta_i+q_i)}\\
&\leq -\frac{(c+1)(c+2)q_i}{\left(T_{\psi_i}+(c+1)(c+2)+c+q_i\right)\left(T_{\psi_i}+c+q_i\right)} - \frac{(c+2)q_i}{\left(T_{\psi_i}+(c+1)(c+2)+q_i\right)\left(T_{\psi_i}+(c+1)(c+2)\right)} + \frac{q_i}{\beta_i(\beta_i+q_i)}.
\end{aligned}
$$

To verify this by computer to be nonpositive, we execute the following code in Mathematica:

```
f[t_, c_, q_] =
q/(t (t + q)) - (c + 2) q/(((c + 2) t + (c + 1) (q - c - 3) + (c + 1) (c + 2))
* ((c + 2) t + (c + 1) (q - c - 3) + (c + 1) (c + 2) + q))
- q (c + 1) (c + 2)/(((c + 2) t + (c + 1) (q - c - 3) + c + q)
* ((c + 2) t + (c + 1) (q - c - 3) + c + q + (c + 1) (c + 2)));
Reduce [{f[t, c, q] <= 0, q <= c, q >= 1, t + q >= c + 2}, t]
```

which gives the following result, thus verifying nonpositivity.

```
(c = = 1 && q = = 1 && t >= 2) || (c > 1 && 1 <= q <= c && t >= 2 + c - q)
```

This completes the proof of Lemma F.6. □

F.3. Proof of Lemma 2

Lemma 2 (Reproduced Here for Clarity). *When executing FFA with any basic c-gap control vector and $\sigma = 1$, if the nth user to arrive has the largest share, the total resource allocated is at most*

$$\max_{j\in\{0,1,\ldots,c+1\}} \sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} + \sum_{i=0}^{n-2} \frac{1}{n+i(c+1)+j}.$$

In order to prove Lemma 2, it is convenient to introduce some additional notation. Elements of an allocation that appear once are called *singletons*, and those that appear twice are *doubles*. We use the following to make our notation more compact. Instead of individually analyzing each basic control vector, we define a set of allocations $Z^c = \{Z^c_{t,j}\}$ that simultaneously upper bounds all allocations created by all basic c-gap control vectors for any fixed $c \geq 1$.

Definition F.1. For any $c \geq 1$,

- $Z^c_{1,1} = (\sigma)$.
- For $j \in \{2,\ldots,c+1\}$,

$$Z^c_{1,j} = \left(\sigma, \frac{\sigma}{2}, \ldots, \frac{\sigma}{j}\right).$$

- For $n>1$ and $j=0$, let $n' = (n-1)(c+2)-c$:

$$Z^c_{n,0} = \left(\underbrace{\frac{\sigma}{n}, \frac{\sigma}{n}}_{\text{one double}}, \underbrace{\frac{\sigma}{t+1}, \ldots, \frac{\sigma}{n+c}}_{c\text{ singletons}}, \underbrace{\frac{\sigma}{n+c+1}, \frac{\sigma}{n+c+1}}_{\text{one double}}, \ldots, \underbrace{\frac{\sigma}{n'}, \frac{\sigma}{n'}}_{\text{one double}}, \underbrace{\frac{\sigma}{n'+1}, \ldots, \frac{\sigma}{n'+c}}_{c\text{ singletons}}\right).$$

- For $n>1$ and $j \in \{1,\ldots,c+1\}$, let $n' = (n-1)(c+2)+j-c$.

$$Z^c_{n,j} = \left(\underbrace{\frac{\sigma}{n}, \ldots, \frac{\sigma}{n+j-1}}_{j\text{ singletons}}, \underbrace{\frac{\sigma}{n+j}, \frac{\sigma}{n+j}}_{\text{one double}}, \underbrace{\frac{\sigma}{n+j+1}, \ldots, \frac{\sigma}{n+j+c}}_{c\text{ singletons}}, \ldots, \underbrace{\frac{\sigma}{n'}, \frac{\sigma}{n'}}_{\text{one double}}, \underbrace{\frac{\sigma}{n'+1}, \ldots, \frac{\sigma}{n'+c}}_{c\text{ singletons}}\right).$$

Example F.1. $Z^c_{n,j}$ for some possible values of c,n,j:

1. $Z^2_{3,2} = \left(\frac{\sigma}{3}, \frac{\sigma}{4}, \frac{\sigma}{5}, \frac{\sigma}{5}, \frac{\sigma}{6}, \frac{\sigma}{7}, \frac{\sigma}{8}, \frac{\sigma}{8}, \frac{\sigma}{9}, \frac{\sigma}{10}\right)$.
2. $Z^3_{4,0} = \left(\frac{\sigma}{4}, \frac{\sigma}{4}, \frac{\sigma}{5}, \frac{\sigma}{6}, \frac{\sigma}{7}, \frac{\sigma}{8}, \frac{\sigma}{8}, \frac{\sigma}{9}, \frac{\sigma}{10}, \frac{\sigma}{11}, \frac{\sigma}{12}, \frac{\sigma}{12}, \frac{\sigma}{13}, \frac{\sigma}{14}, \frac{\sigma}{15}\right)$.

We now show that Z^c is the set of all possible allocations in the round just before a donor is used.

Proposition F.2. *For any $c \geq 1$, the set of allocations $Z^c = \{Z^c_{n,j} : n \geq 1, 0 \leq j \leq c+1\}$ is precisely the set of all possible allocations of FFA in the round before a donor is used when the inputs to FFA are a basic c-gap control vector and σ.*

Proof. For $n = 1, j \in [0,\ldots,c+1]$, the corresponding basic c-gap control vector is the one with a prefix of exactly j zeros followed by one.

For $n > 1$, first notice that, even though there are $c + 1$ basic c-gap control vectors, there are $c + 2$ possible allocations in which $\frac{\sigma}{n}$ is the largest share because one of the basic c-gap control vectors gives an allocation with two terms equal to $\frac{\sigma}{n}$. At some step, the first of the two is the largest share (and equal to the second largest share), and after $c + 1$ arrivals it gets disrupted, resulting in an allocation that, again, has the largest share equal to $\frac{\sigma}{n}$. Thus, it is easy to confirm that our allocations $Z^c_{n,j}$ start with the correct pattern.

It remains to show that we have the correct number of total terms; that is, an allocation that starts with j singletons has a total of $n' + c = (n-1)(c+2) + j$ terms. Toward this statement, let ℓ be the number of times the pattern "one double followed by c singletons" appears (in $Z^2_{3,2}$ in Example F.1, $\ell = 2$). This pattern has $c + 2$ terms; therefore,

$n' + c = j + \ell(c+2) \Rightarrow n' = \ell(c+2) + j - c$. A different way to count the total number of terms $n' + c$ is to notice that the denominator of the singleton right before one double increases by $c+1$ every time the pattern "one double followed by c singletons" appears. Therefore, the last denominator $n' + c$ is equal to $n + j - 1 + \ell(c+1)$, which (together with the previous equality) implies that $\ell = n - 1$, and thus, $n' = (n-1)(c+2) + j - c$. □

Proof of Lemma 2. The allocation just before a donor is used is necessarily greater than the previous c allocations as there was no donor for the previous c rounds. Therefore, by Proposition F.2, the maximal allocation of $Z^c_{n,j}$ is an upper bound on the maximal allocation of FFA with a basic c-gap control vector. The total allocation of $Z^c_{n,j}$ is

$$\sum_{i=n}^{(n-1)(c+2)+j} \frac{\sigma}{i} + \sum_{i=0}^{n-2} \frac{\sigma}{n + i(c+1) + j},$$

by straightforward summation over the allocation vector of Definition F.1, in which the first is a sum of all the values appearing in the allocation vector and the second part is a sum of the values that have duplicates. Therefore, the total resource allocated when the nth user has not yet been a donor (over the choice of all basic c-gap control vectors) is at most

$$\max_{j \in \{0,1,\ldots c+1\}} \sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} + \sum_{i=0}^{n-2} \frac{1}{n + i(c+1) + j}.$$

The first term is maximized at $j = c+1$, and the second is maximized at $j = 0$. □

F.4. Proof of Theorem 5

Theorem 5 (Reproduced Here for Clarity). *For any $c \geq 1$, it holds that*

$$(H_{c+1})^{-1} \geq \text{FAIRNESS}(c - gap) \geq \left(H_{2c+3} - \frac{1}{2}\right)^{-1},$$

where H_n denotes the nth harmonic number.

Proof of Theorem 5. For the upper bound, consider the basic c-gap control vector $\psi = (10^c)^\infty$. When $c+1$ agents are in the system, the total allocation when executing FFA with ψ and σ is $\sigma + \frac{\sigma}{2} + \cdots + \frac{\sigma}{c+1}$. This is simply the first $c+1$ elements of the harmonic progression multiplied by σ. Therefore, $\sigma H_{c+1} \leq 1$, and so $(H_{c+1})^{-1}$ provides an upper bound on the fairness ratio.

For the lower bound, we bound each of the terms in Corollary 3 separately. The term on the left, $\sum_{i=n}^{(n-1)(c+2)+c+1} \frac{1}{i}$, is decreasing in n. This is restated as Lemma C.2 and proved in Appendix C. Plugging in $n = 2$, we, therefore, have that this term is upper bounded by $\sum_{i=2}^{2c+3} \frac{1}{i} = H_{2c+3} - 1$.

The term on the right, $\sum_{i=0}^{n-2} \frac{1}{n+i(c+1)}$, is upper bounded by $\frac{1}{2}$ for all $c > 1$ (Lemma C.3). Combined, we get that the term inside the brackets is at most $H_{2c+3} - \frac{1}{2}$. Hence, $\left(H_{2c+3} - \frac{1}{2}\right)^{-1}$ is a lower bound on FAIRNESS$(c - \text{gap})$. □

F.5. Proof of Theorem 6

Theorem 6 (Reproduced Here for Clarity). *For every $c \geq 1$ there exist a number $n_c \leq 2(c+4)^2$ such that*

$$\text{fairness}_{\geq n_c}(c - gap) = (c+1)((c+2)\ln(c+2))^{-1}.$$

Proof of Theorem 6. We use Lemma C.1 to bound the left- and right-hand sides of the expression in Lemma 2. For the first term,

$$\sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} \leq \ln\left(\frac{(n-1)(c+2)+j}{n-1}\right) + \frac{1}{2((n-1)(c+2)+j)} - \frac{1}{2n-1}.$$

For the second term,

$$\sum_{i=0}^{n-2} \frac{1}{n + i(c+1) + j} = \frac{1}{c+1} \sum_{i=0}^{n-2} \frac{1}{\frac{n+j}{c+1} + i} = \frac{1}{c+1} \sum_{i=\frac{n+j}{c+1}}^{\frac{n+j}{c+1}+n-2} \frac{1}{i}$$

$$\leq \frac{1}{c+1}\left(\ln\left(\frac{\frac{n+j}{c+1} + n - 2}{\frac{n+j}{c+1} - 1}\right) + \frac{1}{2(\frac{n+j}{c+1} + n - 2)} - \frac{1}{2(\frac{n+j}{c+1}) - 1}\right)$$

$$= \frac{1}{c+1}\ln\left(\frac{n+j+(n-2)(c+1)}{n+j-c-1}\right) + \frac{1}{2(n+j+(c+1)(n-2))} - \frac{1}{2(n+j)-c-1}.$$

Together, this gives that, when executing FFA with any basic c-gap control vector and $\sigma = 1$, as long as the nth user to arrive has never had the user's allocation reduced, the total resource allocated is at most

$$\ln\left(\frac{(n-1)(c+2)+j}{n-1}\right) + \frac{1}{2((n-1)(c+2)+j)} - \frac{1}{2n-1} + \frac{1}{c+1}\ln\left(\frac{n+j+(n-2)(c+1)}{n+j-c-1}\right) + \frac{1}{2(n+j+(c+1)(n-2))} - \frac{1}{2(n+j)-c-1}. \tag{F.1}$$

Taking the derivative of (F.1) with respect to n gives

$$\begin{aligned}
&-\frac{j}{(c+2)n^2+(j-2c-4)n+2+c-j} - \frac{c-j}{(n+j-c-1)(cn-2c+j+2n-2)} \\
&-\frac{c+2}{2(cn-2c+j+2n-2)^2} - \frac{2(c+2)}{(2(c+2)(n-1)+j)^2} + \frac{2}{(2n-1)^2} + \frac{2}{(2n-c+2j-1)^2} \\
&\geq -\frac{j}{(c+2)n^2-2(c+2)n} - \frac{c-j}{(c+2)n^2-n(c+1)(c+4)} \\
&-\frac{1}{2(c+2)(n-2)^2} - \frac{1}{2(c+2)(n-1)^2} + \frac{4}{(2n+c)^2} \\
&\geq -\frac{c}{(c+2)n^2-n(c+4)^2} - \frac{1}{(c+2)(n-2)^2} + \frac{4}{(2n+c)^2} \\
&\geq \frac{1}{n^2+n(c+1)} - \frac{c+1}{(c+2)n^2-n(c+4)^2} \\
&= \frac{c+2}{(c+2)n^2+n(c+4)^2} - \frac{c+1}{(c+2)n^2-n(c+4)^2}.
\end{aligned} \tag{F.2}$$

For every $n > n_c = 2(c+4)^2$, Expression (F.2) is strictly positive. Therefore, it holds that, for every c, there exists a n_c such that, for every $n > n_c$ and every $j \in \{0, 1, \ldots, c+1\}$,

$$\sum_{i=n}^{(n-1)(c+2)+j} \frac{1}{i} + \sum_{i=0}^{n-2} \frac{1}{n+i(c+1)+j}$$

is upper bounded by an expression that is strictly increasing in n. Fixing c, the limit of this expression with respect to n is $\frac{c+2}{c+1}\ln(c+2)$ for every j. Hence, for $n > n_c$, for every j, it holds that this expression is less than $\frac{c+2}{c+1}\ln(c+2)$.

The lower bound on the fairness (i.e., showing that this is tight) is straightforward by using (the other direction of) Lemma C.1. □

References

[1] Ahmadi RH, Dasu S, Tang CS (1992) The dynamic line allocation problem. *Management Sci.* 38(9):1341–1353.
[2] Alon N (1987) Splitting necklaces. *Adv. Math.* 63(3):247–253.
[3] Aziz H, Mackenzie S (2016) A discrete and bounded envy-free cake cutting protocol for four agents. *Proc. 48th Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 454–464.
[4] Benade G, Kazachkov AM, Procaccia AD, Psomas CA (2018) How to make envy vanish over time. *Proc. 19th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 593–610.
[5] Bertsimas D, Farias VF, Trichakis N (2011) The price of fairness. *Oper. Res.* 59(1):17–31.
[6] Bertsimas D, Farias VF, Trichakis N (2013) Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Oper. Res.* 61(1):73–87.
[7] Boiney LG (1995) When efficient is insufficient: Fairness in decisions affecting a group. *Management Sci.* 41(9):1523–1537.
[8] Brams SJ, Taylor AD (1995) An envy-free cake division protocol. *Amer. Math. Monthly* 102(1):9–18.
[9] Budish E (2011) The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J. Political Econom.* 119(6):1061–1103.
[10] Ciocan DF, Farias V (2012) Model predictive control for dynamic resource allocation. *Math. Oper. Res.* 37(3):501–525.
[11] Correa JR, Schulz AS, Stier-Moses NE (2007) Fast, fair, and efficient flows in networks. *Oper. Res.* 55(2):215–225.
[12] Demers A, Keshav S, Shenker S (1989) Analysis and simulation of a fair queueing algorithm. *Comput. Comm. Rev.* 19:1–12.
[13] Deng X, Qi Q, Saberi A (2012) Algorithmic solutions for envy-free cake cutting. *Oper. Res.* 60(6):1461–1476.
[14] Fishburn PC, Sarin RK (1994) Fairness and social risk I: Unaggregated analyses. *Management Sci.* 40(9):1174–1188.
[15] Freeman R, Zahedi SM, Conitzer V, Lee BC (2018) Dynamic proportional sharing: A game-theoretic approach. *Proc. ACM Measurement Anal. Comput. Systems* (Association for Computing Machinery, New York), 1–36.
[16] Friedman E, Psomas CA, Vardi S (2017) Controlled dynamic fair division. *Proc. 18th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 461–478.
[17] Guo BN, Qi F (2011) Sharp bounds for harmonic numbers. *Appl. Math. Comput.* 218(3):991–995.
[18] Guo M, Conitzer V, Reeves DM (2009) Competitive repeated allocation without payments. *Internat. Workshop Internet Network Econom.* (Springer, Berlin, Heidelberg), 244–255.
[19] Haitao Cui T, Raju JS, Zhang ZJ (2007) Fairness and channel coordination. *Management Sci.* 53(8):1303–1314.

[20] Huh WT, Liu N, Truong VA (2013) Multiresource allocation scheduling in dynamic environments. *Manufacturing Service Oper. Management* 15(2):280–291.
[21] Isard M, Prabhakaran V, Currey J, Wieder U, Talwar K, Goldberg A (2009) Quincy: Fair scheduling for distributed computing clusters. *Proc. ACM SIGOPS 22nd Sympos. Oper. Systems Principles* (Association for Computing Machinery, New York), 261–276.
[22] Karsten F, Slikker M, van Houtum GJ (2015) Resource pooling and cost allocation among independent service providers. *Oper. Res.* 63(2): 476–488.
[23] Kash IA, Procaccia AD, Shah N (2014) No agent left behind: Dynamic fair division of multiple resources. *J. Artificial Intelligence Res.* 51: 579–603.
[24] Li B, Li W, Li Y (2018) Dynamic fair division problem with general valuations. *Proc. 27th Internat. Joint Conf. Artificial Intelligence* (International Joint Conferences on Artificial Intelligence Organization), 375–381.
[25] Milojičić DS, Douglis F, Paindaveine Y, Wheeler R, Zhou S (2000) Process migration. *ACM Comput. Surveys* 32(3):241–299.
[26] Moulin H, Stong R (2002) Fair queuing and other probabilistic allocation methods. *Math. Oper. Res.* 27(1):1–30.
[27] Othman A, Papadimitriou C, Rubinstein A (2014) The complexity of fairness through equilibrium. *Proc. 15th ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 209–226.
[28] Pazner EA, Schmeidler D (1978) Egalitarian equivalent allocations: A new concept of economic equity. *Quart. J. Econom.* 92(4):671–687.
[29] Segal-Halevi E (2018) Redividing the cake. *Proc. 27th Internat. Joint Conf. Artificial Intelligence* (International Joint Conferences on Artificial Intelligence Organization), 498–504.
[30] Steinhaus H (1948) The problem of fair division. *Econometrica* 16(1):101–104.
[31] Topaloglu H, Powell WB (2005) A distributed decision-making structure for dynamic resource allocation using nonlinear functional approximations. *Oper. Res.* 53(2):281–297.
[32] Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J (2015) Large-scale cluster management at Google with Borg. *Proc. 10th Eur. Conf. Comput. Systems*, 18.
[33] Walsh T (2011) Online cake cutting. Brafman RI, Roberts FS, Tsoukiàs A, eds. *Algorithmic Decision Theory* (Springer, Berlin, Heidelberg), 292–305.
[34] Zeng D, Psomas A (2020) Fairness-efficiency tradeoffs in dynamic fair division. *Proc. 21st ACM Conf. Econom. Comput.* (Association for Computing Machinery, New York), 911–912.