On fair division of a homogeneous good

Uriel Feige* Moshe Tennenholtz†

February 4, 2014

Abstract

We consider the problem of dividing a homogeneous divisible good among n players. Each player holds a private non-negative utility function that depends only on the amount of the good that he receives. We define the *fair share* of a player P to be the average utility that a player could receive if all players had the same utility function as P. We present a randomized allocation mechanism in which every player has a dominant strategy for maximizing his expected utility. Every player that follows his dominant strategy is guaranteed to receive an expected utility of at least $n/(2n-1)$ of his fair share. This is best possible in the sense that there is a collection of utility functions with respect to which no allocation mechanism can guarantee a larger fraction of the fair share. In interesting special cases our allocation mechanism does offer a larger fraction of the fair share.

1 Introduction

We consider the problem of dividing a homogeneous good of size n among n players. That is, each player gets some part of the good (some parts might be empty), and the total size of allocated parts does not exceed n. The good is homogeneous in the sense that the utility of a part to a player depends only on the size of the part – any two parts of the same size have the same value. Here are some examples of situations in which there is a homogeneous divisible good. The reader may note that in these examples the utility functions of the players are not linear, and in some cases, not even monotone.

- *Storage space.* Each player has a file to store, and needs storage space exactly equal to the size of the file. Smaller space has no value, and larger space does not offer additional value.
- *Transmission bandwidth.* Each player has a video stream to transmit. Some minimum amount of bandwidth is required to make the transmission

*Department of Computer Science and Applied Mathematics, Weizmann Institute, Rehovot. uriel.feige@weizmann.ac.il. Work done in Microsoft Research, Herzliya.

†Microsoft Research, Herzliya, and Technion, Haifa. moshet@microsoft.com.

feasible. Quality of service can improve with more bandwidth, up to some point beyond which additional bandwidth is not useful.

- *Processing time.* Each player holds a job and desires this job to be finished before some common deadline. The good is the total processing time available on some server. This version was considered in [4]. We note that in this interpretation one may also consider (though we shall not do so in this paper) multiple stage allocation processes, in which a player is allocated some running time, and if his job happens to finish in less time, the remaining time can be given to other jobs.

- *Grant money.* The good may be a certain amount of money that a funding agency wishes to hand out as research grants. Each player submits a research proposal, and the extent of funding (if any) for the proposal will determine which aspects of the proposal will be carried out. Receiving more funding than asked for might be detrimental for the corresponding researcher, as inability to spend the funds within the grant regulations may cause unwanted administrative overhead.

1.1 Overview of our contributions

Consider the special case in which there are two players, P_1 and P_2, that each has a file to store. The homogeneous divisible good is storage space, and the total amount of storage available is 2 gigabytes. How should one divide the storage space among the two players? The *balanced* allocation gives each player 1 gigabyte. It is satisfactory if each player has a file no larger than 1 gigabyte. However, what if each player has a file of size 2 gigabytes? In this case no player gets any utility from 1 gigabyte, and the balanced allocation may be fair, but useless. A more useful allocation would be the *dictator* allocation: give one of the players the full 2 gigabytes and give the remaining player nothing. To make this allocation fair, one can revert to the *random dictator* allocation: choose the player who gets 2 gigabytes at random.

Consider now the case that P_1 has a file of size 1 gigabyte, whereas player P_2 has a file of size 2 gigabytes. In this case P_1 prefers the balanced allocation, whereas P_2 prefers the random dictator allocation. What would be a fair solution? A possible answer is to give equal weight to the preferences of the players: choose the balanced allocation with probability 1/2, and the random dictator allocation with probability 1/2. However, the notion of fairness that we propose in the current work will lead to a different solution: choose the balanced allocation with probability 1/3, and the random dictator allocation with probability 2/3. What is the logic behind such a solution? Observe that the balanced allocation is useless for P_2, whereas the random dictator allocation is partly useful for P_1 (with probability 1/2 P_1 gets sufficient storage space). Hence it makes sense to choose the random dictator allocation (P_2's preference, which also has some utility for P_1) with higher probability than the balanced allocation (P_1's preference, that has no utility for P_2). The probability 2/3 is

chosen to reach an exact balance in this consideration: P_2 gets his preferred allocation rule (random dictatorship) with probability 2/3, whereas P_1 effectively gets his preferred allocation rule (balanced allocation) with probability 2/3 as well (1/3 comes from the balanced allocation rule, and another 1/3 comes from the case in which the random dictator rule made P_1 the dictator).

The notion of fairness that we introduce in this work is based on similar principles. It applies in general settings with arbitrarily many players, holding arbitrary nonnegative utility functions (in the special case considered above utility functions were limited to be threshold functions). Basically, we define the *fair share* of a player P to be the expected utility that P would get if P could choose the allocation rule that maximizes his expected utility (conditioned on the allocation rule treating all players symmetrically, as was the case with the balanced allocation and the random dictator allocation). Ideally, we would like each player to get his fair share. However, as illustrated in the example above, if different players prefer different allocation rules, there might not be any allocation (or distribution over allocations) that gives every player his fair share. Hence our randomized allocation rules attempt to get as close as possible to the ideal, giving each player expected utility that is ρ times his fair share, with ρ as large as possible. We show that achieving a value of $\rho = \frac{n}{2n-1}$ is always possible (indeed, in the example above we had $n = 2$ and $\rho = 2/3$), and that larger values of ρ are achievable in interesting special cases.

Another aspect that we address in this work is that of allocation mechanisms in which the utility functions of players are private information to the respective players. Given the specification of the allocation mechanism, this defines a game between the players, in which every player may provide input in a strategic fashion to the mechanism, in hope of maximizing his expected utility from the eventual resulting allocation. We design allocation mechanisms in which players have dominant strategies in the resulting game, and moreover, if players play according to their dominant strategies the value of ρ (referred to above) is at least $\frac{n}{2n-1}$.

1.2 Related work

Fair division problems deal with a variety of issues, ranging from divisible to indivisible goods, as well as from dealing with the division of one good to the division of several goods; on another dimension they deal with a variety of users' models, whether they have additive or other utility functions (see [2] for an overview). This paper deals with the most basic type of setting on one side – a single homogeneous divisible good, and with a most elaborated setting on the agents' utility side – private information, general (non-additive) utility. While this type of problem is highly applicable to many central applications, especially to the division of computational goods – time, bandwidth, storage, as illustrated in the multi-billion cloud services industry - to the best of our knowledge it was not addressed in the related impressive fair division literature. One somewhat related discussion appears in [3] in which the authors deal with extension of cake-cutting to some limited form of non-additive valuation. A major issue is

to identify an appropriate fairness criterion for such settings. Indeed, yet again motivated by cloud services division, we assume no impersonal comparison of utilities; customers' preferences create externality for which all customers are jointly responsible (i.e. too high demand for the resources). Hence, we adopt the following criterion of fairness: we aim to bring all agents as close as possible to their perceived unanimity utility (see [8]): an agent's perceived unanimity utility is the maximal utility he can hope for if all agents were like him and the system would have picked the highest equal utility outcome. While Moulin [8] refers to this notion as part of an axiomatic treatment, in which one can require all agents' actual utility to be higher (resp. lower) than their perceived unanimity utilities, we look at it as a way of optimizing fairness: we wish the system to bring all agents to a certain fraction of their perceived unanimity utilities. In the numerical example given in Section 1.1, the perceived unanimity utility of P_1 is 1, and that of P_2 is 1/2. Each agent gets in expectation a 2/3-fraction of his perceived unanimity utility.

In the classical fair division literature there are two canonical approaches for fair division: envy-free mechanisms (studied since [5]; see [13] for a discussion), and mechanisms leading to proportional allocation. The latter is typically applied in settings like cake cutting where the aim is that each of the n agents will obtain at least $\frac{1}{n}$ of the cake as perceived according to his own utility. In many cases proportional division entails by definition the envy freeness property. Our study shares some of the goals of proportional allocation, but adapts them to a different setting. Recall that we aim at dividing a single divisible homogeneous good. In this case, if all agents have linear or concave utilities then split into equal shares provides a trivial satisfactory solution, and if all agents have convex utilities then an allocation of the whole good to a random selected agent seems as the only reasonable option; indeed, this is the reason that with such simple utilities discussed in the literature the case of a single divisible homogeneous good is neglected. However, when agents may have a mixture of arbitrary utility functions, approximating proportional allocation becomes a tricky question. The approach we take attempts to approximate it in the best way possible.

The challenge we are after is basically the rationing problem, started with the work of Banker [1], in which an infinitely divisible good is to be partitioned based on individual claims. While the techniques used there are deterministic, we appeal to probabilistic rationing methods; the latter have been the subject of work in the context indivisible units [9]. While allocation of resources in proportion to individual claims is the oldest formal rule for distributive justice, going back to Aristotle, allocating goods in say cloud services in a "proportional" manner might be a tricky issue, as users may have quite different preferences' shapes for using the corresponding services. By trying to obtain the perceived unanimity utilities for all agents we overcome this obstacle. Naturally, this cannot be always guaranteed, and following the long tradition on fair algorithms in computational media in other contexts (see e.g. [7]) we appeal to a notion of approximation.

Our work fits the setting of direct revelation mechanisms in mechanism design (starting from Gibbard [6]). In our mechanisms, the dominant strategies of

the players is to reveal upfront (a deterministic function of) their utility functions. Notice that in quite a few fair division contexts, and in particular in ones that fit the single divisible good setup, more specialized action languages are typically used, as they fit metaphors such as cake cutting, in which for example the algorithm may query the agents for various values (see [11]). Our treatment is more in the spirit of mechanism design, and in particular in the spirit of mechanism design without payment [12], and the related approximation approach [10].

A similar approach to fair division in the particular context of scheduling appears in [4]. While less general that the approach taken in this paper, that earlier work considers some additional features that can be associated with the scheduling domain; for example, in that setting each agent is allocated a probability distribution over time slots, and if it finishes earlier than expected, the remaining time can be exploited by others.

2 Terminology and conventions

There are n players $1, \ldots n$ and one divisible homogeneous good G. For simplicity of future notation, the scale in which the amount of good G is measured is normalized so that the total amount of good available is n. An *allocation* A is a partition of the total amount of good available (namely, of n) into $n+1$ parts $A(0), A(1), \ldots, A(n)$, with $\sum_{i=0}^{n} A(i) = n$ and $A(i) \geq 0$ for every $1 \leq i \leq n$. As $A(0) = n - \sum_{i=1}^{n} A(i)$, an allocation can be represented more concisely by the vector $A(1), \ldots, A(n)$. For each $1 \leq i \leq n$, part $A(i)$ is allocated to player i. Part $A(0)$ remains unallocated. Every player i has his own utility function u_i which is a mapping from the domain $[0, n]$ to R^+, where R^+ denotes here nonnegative reals. Hence u_i associates a nonnegative utility $u_i(A(i)) \geq 0$ with the amount $A(i)$ of good that player i receives. A player is indifferent as to how the rest of the good is divided among the remaining players.

An *allocation rule* takes as input a tuple $(u_1, \ldots, u_n)$ of utility functions, and outputs an allocation. A *randomized allocation rule* takes as input a tuple $(u_1, \ldots, u_n)$ of utility functions, and outputs a probability distribution D over allocations, namely, D is a list of allocations $A_1, A_2 \ldots$ (by the conventions below this list will be finite) and associated probabilities $p_1, p_2 \ldots$, with $\sum p_i = 1$ and $p_i \geq 0$ for every i. Thereafter, one particular allocation is selected from D, where the probability of selecting allocation A_i is p_i. We shall use the notation $A \leftarrow D$ to denote an allocation selected at random according to distribution D.

In our work we choose to distinguish between two aspects of utility functions. One is the value that a player associates with the size of the part that he receives. The other is the attitude of the player towards lotteries. The combination of both aspects gives the value that a player associates with a lottery over different sizes of parts. We shall use the term *utility function* so as to capture only the first of these two aspects, and the term *full utility function* so as to capture both aspects (utilities of lotteries). A special case of central importance to our paper is when the value of a lottery is defined as the expected value over all

outcomes of the lottery. This corresponds to the notion of utility functions as axiomatized by von-Neumann and Morgenstern. In this case the utility function completely describes the full utility function, and then the term utility function will be used for short also when lotteries are involved. However, some results in our paper hold even if the full utility function does not satisfy that axioms of von-Neumann and Morgenstern, and for them the distinction between utility functions and full utility functions is helpful. See Section 6.4 for more details.

The following definition establishes some terminology associated with utility functions.

Definition 1 *A utility function is a function* $u : [0, n] \longrightarrow R^+$ *satisfying* $u(0) = 0$*, but which is not identically 0. We say that a utility function is* monotone *if for all* $0 \leq x < y \leq n$ *we have* $u(x) \leq u(y)$*. We say that a utility function is* concave *if for all* $0 \leq x < y \leq n$ *and* $0 < \lambda < 1$ *we have* $u(\lambda x + (1 - \lambda)y) \geq \lambda u(x) + (1 - \lambda)u(y)$*. We say that a utility function is* convex *if for all* $0 \leq x < y \leq n$ *and* $0 < \lambda < 1$ *we have* $u(\lambda x + (1-\lambda)y) \leq \lambda u(x) + (1-\lambda)u(y)$*. We say that a utility function is* strictly *monotone (concave, convex, respectively) if the inequalities that need to hold in the respective definitions are strict. A utility function is linear if it is of the form* $u(x) = cx$ *for some* $c > 0$*. Linear utility functions are strictly monotone, and non-strictly both concave and convex.*

The convention that a utility function cannot be identically 0 is not essential, but it simplifies the statement of some future claims. For example, it ensures that Definition 4 does not involve division by 0.

Much of the nontrivial content in our work stems from the fact that even though the good is homogeneous and divisible (which is a relatively simple object to handle), the utility functions are arbitrary, and hence more difficult to handle than some standard classes of utility functions such as linear ones or concave ones.

For convenience (see for example the Remark 3), and practically (though not formally) without loss of generality, we assume that there is some small but finite $\epsilon > 0$ such that G can only be divided into parts whose sizes are multiples of ϵ, and every such division is possible. Equivalently, the good is composed of n/ϵ identical *atoms*, and an atom is not divisible. Hence there are only finitely many ways of dividing up the good. Likewise, the distribution implied by a randomized allocation is supported on only finitely many allocations. Utility functions may be defined on the whole interval $[0, n]$, but only their values on multiples of ϵ matter. Whenever applicable, we shall assume that ϵ divides any number that comes up in examples that we give for utility function (and in particular, ϵ divides n, divides 1 and divides $1/2$).

3 Formal definitions for ρ-fair allocation mechanisms

In this section we define and discuss the three aspects (fairness, economic efficiency, and dominant strategies) that are of interest to us in our allocation

mechanisms.

3.1 Fairness

In our notion of fairness, we wish to treat all players as equals. That is, they all have the same entitlement to the good. Later, in Section 6.2 we shall discuss extensions of our definitions to the case of different entitlements for different players.

Our notion of fairness is based on two principles. The example in Section 1.1 can serve as motivation for the use of these principles.

One principle is to define fairness with respect to randomized allocation rules rather than to deterministic ones. That is, instead of associating the notion of fairness with a single allocation, we associate it with a probability distribution over allocations. As such, our notion of fairness is an ex-ante one rather than an ex-post one: it refers to expectations prior to selecting one allocation out of the given distribution, rather than to the allocation that is actually selected.

The other principle is to have fairness refer to the expected utility that a player derives from the randomized allocation, rather than to the expected amount of good that the player receives. For this to make sense, we need to normalize the utility functions of the different players so that they are all measured in the same units. Any such normalization is arbitrary to some extent, especially as we are assuming allocation rules without payments (and hence one cannot measure utility by the amount of money that a player is willing to pay in order to obtain this utility). Despite this, we propose a particular normalization. In the context of fairness we believe that our choice of normalization is well motivated [1]

Consider a utility function u, and a distribution D over allocations to n players. We say that D is u-fair, if every player gets exactly the same expected utility with respect to the utility function u. Namely, $E_{A\leftarrow D}[u(A(i))] = E_{A\leftarrow D}[u(A(j))]$ for all $1 \leq i < j \leq n$. In this case we use u_D as shorthand notation for $E_{A\leftarrow D}[u(A(i))]$.

Proposition 2 *Given utility function u and using notation as above, let t_u denote the supremum of u_D taken over all randomized allocations D that are u-fair. Then there is a particular distribution D supported over n allocations that attains this supremum, namely, $t_u = u_D$ for this particular D.*

Proof. The distribution alluded to in the proposition can be obtained as follows. Find an allocation A maximizing $\sum_{i=1}^{n} u(A(i))$. Such an allocation must exist because there are only finitely many partitions of the good G. Thereafter, choose D to be the uniform distribution over the n allocations that can be derived from this particular allocation by permuting the parts $A(1), \ldots, A(n)$ in a cyclic order. ■

[1] For a plausible proposal of a different normalization, see Section 6.1.

Remark 3 *Proposition 2 is not true if the good G is infinitely divisible (this is one of the reasons for our convention at the end of Section 2). Consider for example the case of $n = 2$ and the utility function for $u(x) = \frac{4}{9}x^2$ for $\frac{1}{2} < x \leq 3/2$, with $u(x) = 0$ for $x \leq 1/2$ and $u(x) = 1$ for $3/2 \leq x \leq 2$. This utility function is not continuous, which is fine, because we do want to allow for discontinuities in utility functions (see the example in Section 1.1). The value of the supremum is $t_u = \frac{5}{9}$, but no randomized allocation attains this supremum, because u is not continuous (to the right) at $x = 1/2$.*

It is not difficult to see that if the utility function u is concave then $t_u = u(1)$, and if u is convex than $t_u = u(n)/n$.

Given the value of t_u as in Proposition 2, the normalized version of u is $\hat{u} = u/t_u$. We call this the *unanimity normalization*. An expected utility of 1 is the maximum that a player P with normalized utility function $\hat{u}$ can hope to achieve in any fair distribution over allocations, if all players had the same utility function as P. We think of an expected normalized utility of 1 as the *fair share* of a player. A randomized allocation that gives every player the same expected normalized utility (regardless of whether the expected normalized utility given to each player equals 1 or not) gives each player exactly the same fraction of the utility of his perceived fair share, and hence is arguably fair. The above discussion is summarized (in an equivalent way) in Definitions 4 and 5.

Definition 4 *Given a utility function u and a number of players n, the normalized version of u is $\hat{u}(x) = u(x)/t_u$, where $t_u = \frac{1}{n} max_A \sum_{i=1}^{n} u(A(i))$, where A ranges over all valid allocations (satisfying $\sum_{i=1}^{n} A(i) \leq n$ and $A(i) \geq 0$ for $1 \leq i \leq n$).*

Definition 5 *A distribution D over allocations is* fair *with respect to utility functions $(u_1, \ldots, u_n)$ if and only if every player gets exactly the same normalized expected utility, where the normalization is as in Definition 4. Namely, for all $1 \leq i < j \leq n$, $E_{A \leftarrow D}[\hat{u}_i(A(i))] = E_{A \leftarrow D}[\hat{u}_j(A(j))]$. A randomized allocation rule is* fair *if given a tuple $(u_1, \ldots, u_n)$ it produces a fair randomized allocation.*

Definition 5 is related to earlier definitions given in [8] and [4]. See Section 1.2 for more details.

3.2 Economic efficiency

Though our notion of fairness was guided by aspects of economic efficiency, the fairness notion per se does not offer any level of economic efficiency.

In the current manuscript the utility functions (even the normalized ones) for players are on a subjective scale for each player. A common way of incorporating economic efficiency considerations in such cases is by requiring Pareto optimality. This is not the approach that we will follow in this manuscript, for reasons discussed in Section 6.3. In this Section we only note that if utility functions are strictly monotone then an allocation is Pareto-optimal if and only if it is *complete*.

Definition 6 *An allocation is* complete *if all the good is allocated.*

Instead of Pareto optimality, we shall consider a different notion of economic efficiency that we shall refer to as *individual efficiency*, or *safety*. It comes together with a parameter $\rho > 0$.

Definition 7 *A randomized allocation is ρ-safe if every player is guaranteed expected utility of at least ρ, where the utility of player i is measured according to the player's normalized utility function $\hat{u}_i$ (as in Definition 4).*

In general, our goal would be to obtain randomized allocations that are fair (in the sense of Definition 5) and safe for a value of ρ as large as possible. A natural target value is $\rho = 1$, and this value is indeed achievable in several special cases.

Proposition 8 *If all utility functions are concave then there is a fair allocation that is 1-safe and complete. If all utility functions are convex then there is a fair randomized allocation that is 1-safe and complete. If all players have the same normalized utility function then there is a fair randomized allocation that is 1-safe.*

Proof. For concave utility functions, $\hat{u}(1) = 1$. Hence the allocation $A(i) = 1$ for all i is fair, 1-safe and complete.

For convex utility functions, $\hat{u}(n) = n$. Hence the randomized allocation $A(i) = n$ for a random player i and $A(j) = 0$ for $j \neq i$ is fair, 1-safe and complete.

If all players have the same normalized utility function, then the randomized allocation in the proof of Proposition 2 is fair and 1-safe. It is complete if $\hat{u}$ is monotone[2], but not necessarily complete if $\hat{u}$ is not monotone. ■

3.3 Allocation mechanisms

For allocation rules, we assumed that the utility functions of players are given. In this section we assume that the utility functions are private information of the respective players. We consider *allocation mechanisms* that operate in two stages. In the first stage each player i provides a report r_i of his choice to the mechanism. In the second stage the mechanism M computes a randomized allocation $D = M(r_1, \ldots, r_n)$ as a function of the reports, and selects one allocation $A \leftarrow D$ at random according to D.

The only actions available to players are to choose a report. In the special case that the report of each player is his utility function, namely, $r_i = u_i$, an allocation mechanism is the same as a randomized allocation rule. However, allocation mechanisms differ from allocation rules in two respects. One is that players might be strategic and not report their true utility functions. The

[2]More exactly, it can be assumed without loss of generality to be complete, because if u is monotone then the allocation A in the proof of Proposition 2 can be taken to be complete.

other is that the reporting language need not be that of utility functions – it could be that reports take some other form. This last aspect is essential to our randomized allocation mechanisms. The utility function of a player in the sense used in this paper (recall the discussion prior to Definition 1) does not suffice in order to describe the player's true preferences. A complete description of the preferences of a player would be what we referred to as a *full utility function* that includes also his preferences over lotteries, and not just his preferences over allocations. This might seem to require the reporting language to be more complex than that of stating a utility function. However, in our case the opposite holds, and our mechanisms use a reporting language that in general requires less communication than reporting a utility function. Reporting a utility function might require a player to communicate n/ϵ real numbers (recall that utility functions need be defined only on integer multiples of ϵ), which might be a lot of communication if ϵ is very small, and if high precision is needed in order to specify each real number[3]. It will turn out that the allocation mechanisms that we design will require from each player a report that can be encoded using only $n \log \frac{n}{\epsilon}$ bits, regardless of the complexity of the underlying utility function.

Let us now return to the issue of strategic behavior. Every player i is assumed to know his own utility function u_i, the value of n and ϵ, and the function M. Given n, ϵ, M, a strategy $S_{i,n,\epsilon,M}$ for player i is a function that maps utility functions to reports. We shall omit i, n, M from this notation, and then we have that $r_i = S_i(u_i)$. More generally, one may consider also mixed strategies that output a distribution over reports, but this will not be needed in the current work.

In the current work we consider the benefits that an allocation mechanism can offer to players that are *expectation maximizers*[4]. The goal of an expectation maximizing player i is to maximize his own expected utility $E[u_i(A(i))]$, where expectation is computed relative to the randomized allocation that is generated by the mechanism. Observe that in general the report r_i can influence this randomized allocation.

The solution concept that we shall use is that of dominant strategies. (A more accurate term would be *strategies that are dominant for expectation maximizers*, but we shall use the shorter term *dominant strategies* with the hope that no ambiguity arises because of it.)

Definition 9 *Given an allocation mechanism M and values for n and ϵ, a report r_i is a* dominant report *for player i that has utility function u_i, if regardless of the reports of other players, r_i maximizes the expected utility of player i. Namely, for every tuple $R = (r_1, \ldots, r_n)$ with r_i at its ith coordinate, and for every tuple R' that differs from R only on its ith coordinate,*

[3]Observe that players might need to know n and ϵ in order to bound the size of their report. Without loss of generality we assume throughout that n and ϵ are publicly known. For example, they can be announced by the mechanism prior to soliciting reports from the players.

[4]We remark here that our mechanisms retain many of their desirable properties even if players are not expectation maximizers, but we defer discussion of this aspect to Section 6.4.

$E_{A \leftarrow M(R)}[u_i(A(i))] \geq E_{A \leftarrow M(R')}[u_i(A(i))]$. *A strategy S_i is a* dominant strategy *for player i if for every possibly utility function u_i it holds that $r_i = S_i(u_i)$ is a dominant report.*

We extend the notion of safety introduced in Definition 7 from the context of allocations to the context of allocation mechanisms. Our allocation mechanisms will be required to be safe only to players that follow their dominant strategies.

Definition 10 *An allocation mechanism is ρ-safe if every player has a dominant strategy, and moreover, whenever a player follows his dominant strategy, his expected normalized utility is at least ρ, regardless of reports of other players.*

We now incorporate both fairness (as in Definition 5) and safety, to obtain our definition of a ρ-fair allocation mechanism.

Definition 11 *An allocation mechanism is ρ-fair (at least ρ-fair, respectively) if every player has a dominant strategy, and moreover, whenever all players follow their dominant strategies, the randomized allocation output by the mechanism is fair and ρ-safe (at least ρ-safe, respectively). More generally, if only a subset of the players follow their dominant strategies, the fairness and ρ-safety properties hold with respect to the members of this subset.*

A ρ-fair mechanism is ρ-safe. However, a ρ-safe mechanism is not necessarily ρ-fair, because though all players get expected utility at least ρ (assuming they all play their dominant strategies), some players may get higher utility than others. Sacrificing fairness in this way may have potential benefits: it may increase total welfare, and it may circumvent the conflict underlying Proposition 34.

4 Results

In this section we state our main results. An overview of the proof technique appears in Section 4.1. Full proofs appear in Section 5.

Our first proposition relates to our normalization of utility functions but disregards fairness issues. It limits the total utility that can be derived from an allocation.

Proposition 12 *Let $(\hat{u}_1, \ldots, \hat{u}_n)$ be arbitrary normalized utility functions and let A be an arbitrary allocation. Then $\sum_{i=1}^{n} \hat{u}_i(A(i)) \leq 3n - 1$.*

Our next proposition shows that there are no ρ-fair allocation rules for $\rho > \frac{n}{2n-1}$.

Proposition 13 *There is a choice of normalized utility functions $(\hat{u}_1, \ldots, \hat{u}_n)$ such that for every distribution D over allocations, there is at least one player i for which $E_{A \leftarrow D}[\hat{u}_i(A(i))] \leq \frac{n}{2n-1}$. Namely, no randomized allocation rule can guarantee every player expected normalized utility higher than $n/(2n-1)$.*

Theorem 14 shows that there are allocation rules that match the bounds of Proposition 13. Moreover, there are such allocation mechanisms. The allocation mechanism referred to in Theorems 14, 15 and 17 are all based on a new framework that we refer to as *bin matching*. See Section 4.1 for more details.

Theorem 14 *There is a ρ-fair allocation mechanism with $\rho = n/(2n-1)$.*

Theorem 14 offers no guarantees in terms of Pareto optimality of the allocations that it generates. This is partly remedied in Theorem 15, though at the price of replacing fairness by safety. (Recall that complete allocations are those that allocate all the good. They are necessarily Pareto optimal if utility functions are strictly monotone. See Section 6.3 for more information on Pareto optimality.)

Theorem 15 *If utility functions are required to be monotone, then there is ρ-safe allocation mechanism with $\rho = n/(2n-1)$, that furthermore is supported only on complete allocations.*

Though the value of ρ in Theorems 14 and 15 cannot be improved in the worst case (due to Proposition 13), it can be improved in important special cases.

Definition 16 *For $1 \leq t \leq n$, a utility function is* t-concave *if there is no optimal unanimity allocation (allocation A maximizing t_u in Definition 4) that involves parts larger than t. In particular a strictly concave utility function is 1-concave, and every utility function is n-concave.*

Theorem 17 *There is a randomized allocation mechanism (that shall be called* minbin*) that is ρ-safe for $\rho = n/(2n-1)$ and has the following additional guarantees if players follow their dominant strategies:*

1. *If all utility functions are strictly concave then $\rho = \frac{n}{n+1}$.*
2. *Let t be the smallest value that satisfies both the following conditions: all utility functions are t-concave and t divides n. Then $\rho \geq \frac{n}{n+\lceil t \rceil + 1}$.*

The minbin allocation mechanism underlying the proof of Theorem 17 is supported on allocations that need not be complete. When utility functions are monotone, it is possible to modify the minbin mechanism so that it becomes complete, at the cost of slightly hurting the value of ρ (by a factor no worse than $\frac{n-1}{n}$) in its safety guarantees. See Proposition 19 for example.

4.1 Techniques

The proofs of Propositions 12 and 13 are quite short and do not involve particularly interesting techniques. They can be found in Section 5.1.

The allocation mechanisms that we design (in Theorems 14, 15 and 17) are all based on a new framework that we refer to as *bin matching*, as it combines

aspects of bin packing and of matchings. In this framework, the allocation does not require a player to report his utility function. Instead, the player reports an allocation. Namely, the report R_i of player i is a vector $(r_{i1}, \ldots, r_{in})$ with $\sum_j r_{ij} \leq n$ and $r_{ij} \geq 0$ for every $1 \leq j \leq n$. Every r_{ij} is referred to as an *item*. Hence from each of the n players the mechanism receives n items of nonnegative sizes summing up to at most n. Our mechanism will pack all n^2 items in B bins, where a packing is *legal* if every item is placed in exactly one bin, no two items of the same player are placed in the same bin, and furthermore, the total size of items placed in a bin does not exceed n. Observe that every such bin corresponds to an allocation: if every player with an item in this bin gets a part of the good of size equal to the size of his respective item, then every player gets a nonnegative part (some players might get an empty part), and the total size of allocated parts does not exceed n. The distribution D associated with the allocation mechanism is the uniform distribution over these B allocations. namely, one of the B bins is chosen at random, and the corresponding allocation is given to the players.

Suppose that B depends only on n but not on the reports of the players. In this case, each player can partition the good into n parts of his choice, and is guaranteed to get every part with probability $1/B$. As B is fixed a-priori, a player will maximize his expected utility by partitioning the good into parts whose sum of utilities (according to his own utility function) is maximized. Hence a dominant report for the player is to choose a partition exactly as in the allocation underlying Definition 4. One can readily see that under this dominant report, the expected utility of the player is exactly n/B. Hence such an allocation mechanism is ρ-fair and ρ-safe, for $\rho = n/B$.

To maximize the value of ρ in this approach, one would like B to be as small as possible. This leads us to ask what is the minimum number of bins that surely suffices for a legal packing, regardless of the reports of the players. We call this problem the *bin matching* problem, and denote the minimum number of bins by $B(n)$. Theorem 14 will follow from showing that $B(n) = 2n - 1$. See details in Section 5.2.

Observe that the fact that $B > n$ implies that packing the n^2 items into B bins necessarily involves bins that are not fully packed (the total size of items in them is strictly less than n). Hence the bin matching mechanism does not provide a distribution over *complete* allocations (in the sense of Definition 6). The proof of Theorem 15 extends the bin matching mechanism so that the parts not allocated by it do get allocated in the extended mechanism. The difficulty is to do so while maintaining the property that players have dominant strategies. Our solution involves picking one player P at random, solving the bin matching problem while ignoring P's report, allocating the good to the other players according to the solution of the bin matching problem, and if not all the good was allocated, giving the remaining part to P. The proof that such a mechanism remains ρ-safe for $\rho = \frac{n}{2n-1}$ (even for P) appears in Section 5.3.

The proof of Theorem 17 extends the bin matching mechanism in a different way – if the reports are favorable, it uses less than $2n - 1$ bins, and hence offers players a higher value of ρ. Again, the difficulty is to implement this

principle while maintaining the property that players have dominant strategies. Our *minbin* allocation mechanism does this as follows. For the given reports by all players, pack the items legally in the minimum possible number of bins. Let this number be n_{opt}. In addition, for every player i, let n_i be the number of bins that would suffice for every bin matching instance in which the report of player i is arbitrary and the reports of all other players remain unchanged. Observe that necessarily $n_{opt} \leq n_i \leq 2n-1$ (where the last inequality follows from Theorem 18). Now pick at random one bin out of n_{opt}, and give each player i its item from that bin (if there is such an item) with probability n_{opt}/n_i. Hence each player i gets a fraction of $\frac{n}{n_{opt}}\frac{n_{opt}}{n_i} = \frac{n}{n_i} \geq \frac{n}{2n-1}$ of his fair share. The minimal values for n_{opt} and n_i might not be computable in polynomial time. Nevertheless, we show that even a relaxed version of the min bin allocation mechanism in which one uses polynomial time computable upper bounds on these values suffices in order to prove the bounds claimed in Theorem 17. See Section 5.4 for the full proof.

5 Proofs

5.1 Negative results

Proof of Proposition 12.

Proof. Consider an arbitrary allocation and for every player i, let p_i be the size of the part that player i receives. Then $\sum p_i \leq n$. Definition 4 can be used to upper bound the normalized utility $\hat{u}_i(p_i)$ that a part of size p_i can give player i. This is done as follows. Let $q_i = \lfloor \frac{n}{p_i} \rfloor$ be the maximum number of parts of size p_i into which the good of size n can be partitioned (possibly, leaving a leftover part smaller than p_i). Definition 4 implies that the sum of normalized utilities of any n parts (some of which may be empty) of total size at most n cannot exceed n. Hence if $q_i < n$, we have that $\hat{u}_i(p_i) \leq n/q_i < 2p_i$, and if $q_i \geq n$ then $\hat{u}_i(p_i) \leq 1$. Consequently, the sum of normalized utilities of all players cannot exceed $\sum_{i=1}^{n} \max[1, 2p_i]$. This sum is maximized when $p_1 = n$ and $p_i = 0$ for all $i > 1$, giving $3n-1$. ■

We remark that the bounds in Proposition 12 can be slightly improved, but such improvements are omitted for the sake of simplicity.

Proof of Proposition 13.

Proof. Consider the case in which $n-1$ players each have normalized utility n for the whole good but no utility for any part of it. We call such players *single minded*. The remaining player has normalized utility 1 for parts of size 1 or more, and no utility for smaller parts. We call such a player *unitary*. In this case, in every allocation at most one player can receive positive utility. Consider an arbitrary randomized allocation. As the good is allocated with probability at most 1, either the unitary player receives positive utility with probability at most $\frac{n}{2n-1}$, or some single minded player receives positive utility with probability at most $\frac{1}{2n-1}$. In the former case, the expected normalized utility of the unitary

player is at most $\frac{n}{2n-1}$. It the latter case, the expected normalized utility of the corresponding single minded player is at most $n \cdot \frac{1}{2n-1} = n/(2n-1)$. ■

5.2 Tight bounds for bin matching

Recall the bin matching framework of Section 4.1. The report R_i of player i is a vector $(r_{i1}, \ldots, r_{in})$ with $\sum_j r_{ij} \leq n$ and $r_{ij} \geq 0$ for every $1 \leq j \leq n$. Every r_{ij} is referred to as an *item*. Altogether there are n players, and each player has n items of nonnegative sizes summing up to at most n. One needs to pack all n^2 items in B bins, where a packing is *legal* if every item is placed in exactly one bin, no two items of the same player are placed in the same bin, and furthermore, the total size of items placed in a bin does not exceed n. We let $B(n)$ denote the minimum number of bins that surely suffices for a legal packing, regardless of the reports of the players.

Theorem 18 *For the bin matching problem, $B(n) = 2n - 1$ suffices, and is sometimes necessary.*

Proof. In agreement with the notation in the proof of Proposition 13, a player will be called *unitary* if its n items are each of size 1, and *single minded* if it has one item of size n and its remaining items have size 0. The case of $n-1$ single minded players and one unitary player requires $2n-1$ bins in order to legally pack all the items, showing that $B(n) \geq 2n - 1$.

To show that $B(n) = 2n - 1$ suffices, temporarily remove the $n-1$ largest items (among all n^2 items), breaking ties arbitrarily. Let S denote the set of $n^2 - n + 1$ remaining items, and for every player i, let S_i denote those items of S that belong to player i. Let s denote the size of the largest item in S (which originally was the nth largest item). Pack S into n bins using a bin matching algorithm that we call *sorted first fit* (SFF). This algorithm packs the items player by player into the given number of bins (which is n in our case). For every $1 \leq i \leq n$ and $1 \leq j \leq n$, Player i puts his jth largest item (among items of S_i) in the bin that was jth least loaded with respect to the previous bin matching (breaking ties arbitrarily). SFF ensures that the difference in load between any two bins is at most the size of the largest item, which is s in our case.

Suppose for the sake of contradiction that when running SFF on S with n bins, some bin overflows. Then on one hand, at least one bin has volume above n, and $n-1$ bins have volume above $n-s$. On the other hand, the total volume of S is at most $n^2 - (n-1)s$ (because the $n-1$ largest items were removed). But $n^2 - (n-1)s \leq n + (n-1)(n-s)$ (in fact, equality holds), contradicting the possibility of an overflow. Hence S can be packed in n bins, and $n-1$ additional bins suffice in order to pack the $n-1$ largest items, giving a total of $2n-1$ bins. ■

The combination of Theorem 18 and the discussion in Section 4.1 prove Theorem 14.

5.3 Allocation mechanisms with completeness

Throughout this section we assume that all utility functions are monotone. We shall prove Theorem 15, in which one wants all the good to be allocated. Observe that the proof of Theorem 18 selects a bin that might not be full, and hence not allocate all the good. Simply giving the surplus to one of the players will result in a mechanism in which players might not have dominant strategies, because the optimal report of a player might then depend on the distribution of surplus among bins, and the player does not a-priori know what this distribution is going to be. Hence one needs a different approach. We first present a generic approach that might entail a modest loss in economic efficiency.

Proposition 19 *Consider an arbitrary randomized allocation mechanism M for n players (that might not be complete) that under dominant strategies ensures every player expected utility at least ρ. Then M can be modified to become complete and ensures every player an expected utility at least $(1 - 1/n)\rho$.*

Proof. Apply mechanism M but with slightly modified reports: pick one player P at random and change his report to be some arbitrary default report (e.g., n items of size 1). If the resulting allocation of M happens to be complete, keep it. Else, give the unallocated part of the good to P, thus obtaining a complete allocation. The property of having dominant strategies in the new mechanism is inherited from the corresponding property of M (only the allocation for P changes but P has no influence on his allocation). The guarantee of expected utility for players other than P did not decrease, and the probability of a player to be chosen as P is $1/n$. Hence the new guarantee for a player is an expected utility of at least $(1 - \frac{1}{n})\rho$. ■

Combining Theorem 18 and Proposition 19 gives a complete mechanism that is ρ-safe for $\rho = \frac{n-1}{2n-1}$, which almost matches the requirements of Theorem 15. We now prove Theorem 15 via a modification of the proof of Proposition 19.

Proof. Pick one player P at random and change his report to a default report of being single minded (one item of size n, and the rest of the items of size 0). Now obtain a legal packing BP of all items in $2n - 1$ bins (this can be done by Theorem 18). Pick a bin at random, and for each item in the bin, give the respective player a part of the same size as the item. If not all the good is allocated, give the remaining part to P.

For every player other than P this randomized mechanism offers expected utility $\frac{n}{2n-1}$. We now show that for P it offers at least as much. Consider the packing BP into $2n - 1$ bins and the true (original) report of P. We show that all items of the true report can be added to BP without placing two of these items in the same bin and without any bin overflowing. P's largest item can be placed in the bin of BP that originally contained P's default item of size n, since this default item is no longer needed. P's jth largest item is of size at most n/j. It will be placed in the bin $(j - 1)$th least loaded in BP (excluding the bin that was already used in order to pack P's largest item). This bin had

volume at most $\frac{(n-1)n}{2n-j}$, as otherwise the total volume of bins in BP exceeds n^2. The jth item fits there because $\frac{(n-1)n}{2n-j} + \frac{n}{j} \leq n$ for every $2 \leq j \leq n$.

Observe that when the complete mechanism picks a bin, the size of the part that it gives P is at least as large as the size of the true item packed there. By the assumption that the utility function is nondecreasing, the expected utility obtained by P is at least $n/(2n-1)$. ■

5.4 Better bounds for t-concave utility functions

The crux of the proof of Theorem 17 is to show that for t-concave utility functions, $n+\lceil t \rceil+1$ bins suffice for every corresponding instance of the bin matching problem (if t divides n). In Section 5.4.1 we show this for allocation rules (the utility functions and hence also the value of t are given a-priori). In Section 5.4.2 we show this in the more difficult case of allocation mechanisms (in which players report inputs to the mechanism, and one needs to maintain the property that players have dominant strategies).

5.4.1 Allocation rules with bounded value of t

Let $B(n,t)$ denote the minimum number of bins that surely suffice for bin matching when there are n players and no item is larger than t. Observe that $B(n,1) = n$ and that $B(n,n) = B(n) = 2n-1$. Hence one might be tempted to interpolate and conjecture that $B(n,t) = n+t-1$ for every t. Allowing for noninteger t, the corresponding conjecture might be that $B(n,t) = n + \lceil t \rceil - 1$. Indeed, $B(n,t) = n + \lfloor t \rfloor - 1$ would not suffice, as the following example shows.

Example 20 *Let $t > 1$ be noninteger but divide n. There is one unitary player, and each other player has n/t items of size t and the remaining items of size 0. Each of the n bins that contains an item of the unitary player must have volume loss of $t-1$, hence $(t-1)n$ volume needs to be packed in other bins. This requires $\lceil t \rceil - 1$ additional bins.*

The bound $B(n,t) = n + \lceil t \rceil - 1$ is too optimistic as well. We say that a player is *maximal* if t divides n and the player has n/t items of size t and $n - n/t$ items of size 0.

Proposition 21 *There are instances of bin matching with noninteger t and sufficiently large n that require $n + \lceil t \rceil$ bins.*

Proof. Let $t > 1$ be noninteger, let $n > t$ be divisible by t and sufficiently large. Consider $\lceil t \rceil$ unitary players and $n - \lceil t \rceil$ maximal players. If there are only $n+\lceil t \rceil-1$ bins, then $n+\lceil t \rceil-1-\lceil t \rceil(\lceil t \rceil-1) > n-t^2$ of them have items from all unitary players. Each such bin must lose a volume of at least $2t - \lceil t \rceil$, and so the total volume lost is more than $(n-t^2)(2t-\lceil t \rceil) > 2nt - n\lceil t \rceil - t^3$. The extra space available is only $(\lceil t \rceil - 1)n$, which does not suffice if $(2t+1)n > 2\lceil t \rceil n + t^3$.

This last inequality can be made to hold if the fractional part of t is larger than $1/2$, and n is sufficiently large (in particular, larger than t^3). ■

The following is a straightforward upper bound on $B(n,t)$.

Theorem 22 *For the bin matching problem,* $B(n,t) < n + \lceil t + \frac{t(t-1)}{n-t} \rceil$. *In particular,* $B(n,t) < n + \lceil t \rceil + 1$ *when* $t \leq \sqrt{n}$.

Proof. As in the proof of Theorem 18 we shall use the *sorted first fit* (SFF) bin matching algorithm. Pack the items player by player into the given number of bins (which we shall denote by $n + c$). For every $1 \leq i \leq n$ and $1 \leq j \leq n$, Player i puts his jth largest item in the bin that was jth least loaded with respect to the previous bin matching (breaking ties arbitrarily). SFF ensures that the difference in load between any two bins is at most the size of the largest item. In particular, if there is an overflow, at least one bin has volume above n, and all remaining bins volume above $n - t$. The total volume packed is more then $n + (n + c - 1)(n - t) = n^2 + c(n - t) - t(n - 1)$. This exceeds n^2 when $c \geq t\frac{n-1}{n-t}$. ■

The upper bound in Theorem 22 is nearly tight when $t \leq \sqrt{n}$ (matches the negative examples up to additive 1 or 2, depending on the exact values of t and n). However, as t grows the gap between the upper bound of Theorem 22 and the known lower bound grows linearly with t. Our next theorem narrows this gap to a constant for many values of t, and in particular whenever t divides n. We first introduce some notation.

$k = \lfloor n/t \rfloor$ is the number of items of size t that can be packed in a bin.

$b = \lceil (n-1)/k \rceil$ is an upper bound on the number of bins that suffice in order to pack the $n - 1$ largest items, if they belong to different players.

If t divides n then $b \leq \lceil t \rceil$. However, if t does not divide n then it could be that $b > \lceil t \rceil$. In particular, if $t > n/2$ then $b = n - 1$. In any case, $b \geq t - t/n$.

The proof of Theorem 18 suggests that for b as above, $n + b$ bins may suffice (namely, that $B(n,t) \leq n+b$). Indeed, this is the case whenever the $n-1$ largest items each belong to a different player. Then they can be packed into b bins, and the remaining items into n bins, as in the proof of Theorem 18. However if one of the players holds more than b of the largest items, then it is no longer true that the $n - 1$ largest items can be packed into b bins. Alternatively, one may try to pack only one item from each player (its largest item) in b dedicated bins, and hope that the remaining items can still be packed into n bins. However, this last packing step is sometimes impossible.

Example 23 *Let* $r \geq 2$, $n = 54r$, $h = 3r$, $t = 9r$, *and allow* $n + r = 55r$ *bins. There are* h *heavy players with* $n/t = 6$ *items of size* t. *The other* $n - h$ *light players have items of size 1. Remove one item (the largest) from each player. For the remaining items, any packing of the heavy players uses at most* $5h = 15r$ *bins. Any packing of the light players can place* $n + r - 15r = 40r$ *items per player in the remaining bins, and hence at least* $14r - 1$ *items in the*

$15r$ *bins. The total volume available in the* $15r$ *bins is* $15rn$*. The total demand is* $5hn/6 + (14r-1)(n-h)$*. Hence things fit only if* $\frac{25}{2}r \cdot 54r > 51r(14r-1)$*, or equivalently,* $675r > 714r - 51$ *which does not hold for* $r \geq 2$*.*

Taking increasing values of r*, the above example shows that the number of extra bins (beyond* n*) that are needed in order to legally pack all items but one from each player grows linearly with* n*.*

Nevertheless, the bound $B(n,t) \leq n + b$ can be proved whenever $b \geq \sqrt{n}$ (complementing Theorem 22 that handles the case $b < \sqrt{n}$ quite well).

Theorem 24 *Let* $c = \min[b, \sqrt{n}]$*, where* b *is as defined above. Then* $B(n,t) \leq n + c$*.*

Proof. If the $n-1$ largest items (breaking ties arbitrarily) can be packed into c bins, then the proof follows from that of Theorem 18. Hence it is left to deal with the case that the $n-1$ largest items cannot be packed into c bins. Given that $c \geq b$, this happens only if at least $c+1$ of the $n-1$ largest items belong to the same player. This implies that $s \leq n/(c+1)$ (where as in the proof of Theorem 18, s denotes the size of the nth largest item). We use this last inequality to present a packing. The packing has three phases.

Phase 1. In this phase, all items of size strictly larger than $s = n/(c+1)$ are packed. Observe that no player can have more than c such items, that the total number of these items is at most $n-1$ (as otherwise we are done, as explained above), and hence they all fit in c bins (because $c \geq b$, for b as defined above). Call the c bins used in this phase the *large-item* bins.

Phase 2. For each player, sort its items in order of decreasing size. By the end of Phase 2, the items packed for each player will be a prefix of its sorted order. (Note that this also holds at the end of Phase 1.) The bins used for the packing in Phase 2 are only the large-item bins. As there are only c such bins, for each player a prefix of length at most c is packed (in Phases 1 and 2 combined). Phase 2 ends with an arbitrary maximal packing that satisfies these conditions: no additional item can be packed without violating either the capacity constraints, or the matching constraints, or the prefix constraints. The exact procedure by which such a packing is reached does not matter. There are two possible outcomes of Phase 2 (*small volume* versus *large volume*), and they will determine two different versions of Phase 3.

Phase 3 (small volume). *At least one of the large-item bins had volume at most* $n-s$ *by the end of Phase 2.* This can happen only if the bin contains at least one item from each player (otherwise, the packing is not maximal as there is room for an additional item). The prefix property then implies that if we take one yet unpacked item from every player, all these items also fit in a single bin. Hence the n remaining bins can be used in order to pack all remaining items. (In fact, $n-1$ bins would suffice.)

Phase 3 (large volume). *All large-item bins have volume larger than* $n-s$ *by the end of Phase 2..* Pack all remaining items (each of which has size at most s) in the remaining n bins using algorithm SFF of Theorem 18. This ensures

that the gap between any two the n bins is at most s. An overflow will require total volume greater than $n+(n+c-1)(n-s) \geq n+(n+c-1)cn/(c+1) > n^2$, where the last inequality follows by the condition $c \geq \sqrt{n}$. Hence indeed all items can be packed. ■

Corollary 25 *If $t \geq \sqrt{n}$ and t divides n, then $B(n,t) \leq n + \lceil t \rceil$.*

Proof. The condition that t divides n implies that $b = \lceil t \rceil$. Now the corollary follows from Theorem 24. ■

Corollary 25 is best possible up to one bin, by Example 20. Unfortunately, Proposition 21 cannot be used in order to remove this slackness of one bin, as its proof does not work when $t \geq \sqrt{n}$.

5.4.2 An allocation mechanism with per-instance guarantees

Ideally, for every instance of the bin matching problem, one would like the number of bins used to be the smallest possible for that particular instance. However, doing so conflicts with the requirement of having dominant strategies – given his beliefs about the reports of other players, a player might contemplate changing his own report regarding item sizes so as to make the underlying bin matching problems solvable in fewer bins. Hence instead we propose the following mechanism that we call *minbin*.

For the given reports by all players, pack the items legally in the minimum possible number of bins. Let this number be n_{opt}. In addition, for player i, let n_i be the number of bins that would suffice for every bin matching instance in which the report of player i is arbitrary and the reports of all other players remain unchanged. Observe that necessarily $n_{opt} \leq n_i \leq 2n-1$. Now pick at random one bin out of n_{opt}, and give each player i its item from that bin (if there is such an item) with probability n_{opt}/n_i. Hence each player i gets a fraction of n/n_i of his fair share.

Proposition 26 *The minbin mechanism is ρ-safe for $\rho = n/(2n-1)$.*

Proof. Player i has no influence on the value of n_i. Given an arbitrary value of n_i (which is necessarily at least n), the dominant report of a player (that wishes to maximize his expected utility) is the one maximizing $\sum_{j=1}^{n} \hat{u}_i(r_{ij})$ subject to the constraints $\sum_j r_{ij} \leq n$ and $r_{ij} \geq 0$ for every i. Under this report his expected utility is $n/n_i \geq n/(2n-1)$, where the last inequality follows because $n_i \leq 2n-1$ by Theorem 18. ■

The minbin mechanism as described above requires computing n_{opt} and n_i for every i, tasks that might be computationally expensive. Hence one may consider variations on minbin in which the number of bins used is determined by some polynomial time algorithm (and is not necessarily optimal). We now show that in interesting special cases the guarantees of the minbin mechanism are in fact significantly better than $n/(2n-1)$, and moreover, that these improved guarantees can be obtained by polynomial time versions of minbin.

Proposition 27 *If all players are unitary and follow their dominant strategies, then minbin offers every player expected utility of* $\rho = n/(n+1)$.

Proof. We need to show that in the setting of the proposition $n_i = n+1$ for every player i. Equivalently, one needs to show that $n+1$ bins suffice when $n-1$ reports are unitary and the report of player i is arbitrary. Start with n bins. Consider an arbitrary report for player i and pack its items in distinct bins. In addition, put one item of each of the remaining players in each bin. This packing may not be legal (as player i is no longer unitary). But the total overflow from all bins is at most equivalent to $n-1$ unitary items (of players other than i). As each bin contains all players, one can arrange the overflow items to be items of distinct players, and then all of them fit in one extra bin. ■

We now build upon the results of Section 5.4.1 and prove the following.

Theorem 28 *If no item is larger than* t *and* t *divides* n*, then minbin recovers at least* $n/(n+\lceil t \rceil + 1)$ *of the fair share.*

Proof. The proof breaks into three cases, depending on the value of t.

Case 1. For $t \leq 1$, a bound that is even stronger than required by the theorem is proved in Proposition 27.

Case 2. We now prove Theorem 28 in the special case that $t \leq \sqrt{n}$. In this special case we will be able to drop the condition that t divides n. We shall assume that $t \geq 2$. This assumption can be made without loss of generality because all $1 < t < 2$ allow the same number of bins as the case $t = 2$. We shall show that for $2 \leq t \leq \sqrt{n}$, if only one player has items larger than t, then $n + \lceil t + \frac{t(t-1)}{n-t} \rceil$ bins suffice. Observe that this is the exact number of bins obtained in Theorem 22, but unlike Theorem 22, we now allow one player to have items larger than t. This will imply Theorem 28 when $t \leq \sqrt{n}$.

Let $c = \lceil t + \frac{t(t-1)}{n-t} \rceil$, and assume that we have $n + c$ bins. Call the items larger than t *large*, and all other items small. At most one player, the *large* player, has large items. Run the sorted first fit (SFF) algorithm of the proof of Theorem 22, starting from the large player.

Lemma 29 *Using the notation above, if* $t \leq \sqrt{n}$ *and* $c = \lceil t + \frac{t(t-1)}{n-t} \rceil$*, if there is at most one large player and there are* $n + c$ *bins, then SFF will produce a legal packing, with no overflow.*

Proof. After an iteration of SFF, call a bin *oversized* if its volume exceeds that of the minimum bin by more than t, and *bounded* otherwise. After the first iteration of SFF, those bins containing the large items are oversized. Observe that if a bin is bounded at the end of an SFF iteration, it remains bounded in all future iterations of SFF (in every iteration it receives an item no larger than the one received by the current least loaded bin).

Assume for the sake of contradiction that some bin B overflows. Consider the situation just before the arrival of the player P who caused the overflow,

and let k be the index of bin B in the sorted order. If $k > n$, the bin cannot overflow because the player places no item in B. Hence necessarily $k \leq n$. The size of item placed in bin k is at most n/k. Hence the volume of bin k prior to the arrival of the item (and likewise, the volume of bins larger than k) is at least $n - n/k$. Hence the total volume of bins from k up to $n + c$ is at least $(n + c - k + 1)(n - n/k)$. How did this volume build up? In the first iteration of SFF, the large player could have contributed volume at most n to these bins. Suppose now that the following assumption holds:

Order assumption. The bins in locations k up to $n + c$ where in these locations in all iterations starting from the end of the first iteration to the end of the iteration prior to the arrival of the overflow causing player P.

Using the order assumption, SFF implies that every player between the large player and P (there are at most $n - 2$ such players) could contribute volume at most $(n - k + 1)$ to bins from k onwards. Hence to have an overflow the following inequality must hold:

$$n + (n - 2)(n - k + 1) > (c + n - k + 1)(n - n/k)$$

Observe that $c \geq 3$ (implying among other things that $n \geq 4$) and $n/2 < k \leq n$ (because $t \geq 2$ and at most n/t bins are initially oversized). Replacing n/k by 2 and simplifying one gets the inequality $n > c(n - 2)$, which cannot hold for $c \geq 3$. This completes the proof subject to the order assumption.

We now show that even though the order assumption need not hold, it can be assumed without loss of generality. That is, if there is overflow for SFF in an instance in which the order assumption was violated, there is also overflow for a modified version of SFF in an instance in which this assumption was not violated.

Observe that the proof of Theorem 22 implies that B cannot be a bounded bin, hence it has to be an oversized bin. That is, B must contain a large item. Moreover, all bins from k to $n + c$ must contain large items, as otherwise one of them would be bounded, implying that B is bounded as well. Consider now a modified version of SFF. The key aspect of the modification is that at the beginning of each iteration we only have a relaxation of the requirement that all bins are sorted in increasing order of size. We still start with the bounded bins sorted in increasing order, but for the oversized bin we only require that the bins from location k onwards are at least as large as any of the other bins. We shall still have the property that a bin can be oversized only if it contains a large item, and that once a bin becomes bounded it remains bounded. The proof given above goes through even for such a modified-SFF algorithm, provided that the order assumption holds for it.

Suppose now that at some iteration the order assumption failed. Namely, at that iteration there where two oversized bins i and j with $i < k \leq j$, such that after the new player N was added bin i was moved to index $i' \geq k$ whereas bin j moved to index $j' < k$. Observe that the item just placed in bin i cannot be large. Hence the new volume of i exceeds the new volume of j by at most $s \leq t$. Consider now a modified input instance in which the sizes of items of

the large player were modified: the item in bin j was larger by s and the item in i was smaller by s. For modified-SFF this modification need not have any effect on the order of bins until the arrival of player N. Then after the arrival of N we get the same configuration of heights of bins as without the modification, except that now it is bin $i < k$ who moved to $j' < k$ and bin $j \geq k$ who moved to $i' \geq k$.

We use the above modification repeatedly whenever the order assumption is violated, so as to get an instance were the order assumption holds. For the repeated applications we need to establish that once the amount of volume shifted from a large item is sufficient to make it small, the corresponding bin i necessarily becomes bounded (and hence is no longer a candidate to move to a location above k). Indeed, its volume cannot exceed that of the current minimum bin M by more than t, by the following argument. If M was always smaller than i, every player except the large player must have contributed to M an item at least as large as the one that it contributed to bin i. Hence the surplus in size of i compared to M is at most its large item, but the size of its large item is brought down to become at most t. If M was at some point larger than i, then i cannot overtake M by more than t (the size or largest item that could arrive since that point). ■

Case 3. It remains to prove Theorem 28 in the case that $t > \sqrt{n}$ (and recall that t divides n). This will be based on a modification of the proof of Theorem 24. Let c be defined as in Theorem 24. We shall have n main bins, c large-item bins, and an auxiliary bin. As in Theorem 24, we shall first consider the $n-1$ largest items. The difference now is that items (that belong to the large player) might be larger than t. We call such items *huge*. We shall show that one auxiliary bin (beyond the c large-item bins) suffices in order to handle this difficulty. The proof breaks into two subcases.

Case 3.1. Consider first the case that no player holds more than c of the $n-1$ largest items. To simplify the proof, round up the size of each of the $n-1$ largest items to the nearest multiple of t. Let $\ell \leq c$ be the number of huge items (all these items must belong to the large player). All remaining large items can be packed (player by player, filling the bins in a round robin fashion) in the c bins, while leaving at least ℓ *eligible* bins, where a bin is eligible if it has at least t-volume still available, and it contains no item of the large player. Now pack all huge items in distinct eligible bins. This might cause some of the eligible bins to overflow. To handle the overflow, remove from each overflowed bin the minimum necessary number of items of size t (hence not belonging to the large player) and place them in the auxiliary bin. We need to show that this can be done in way insuring the the auxiliary bin is legally packed.

First, let us consider the total volume placed in the auxiliary bin. The total increase of volume of huge items that resulted from rounding up to the nearest multiple of t is strictly less than ℓt. This rounding up volume is more than compensated by the t empty volume initially available in each eligible bin in which a huge item is placed. Hence the total volume moved out of eligible bins is strictly less than the total volume of huge items. As all huge items belong to

one player, their total volume is at most n. Hence strictly less than volume n is moved into the auxiliary bin.

Second, let us show that the auxiliary bin need not contain two items of the same player. By Hall's theorem, it suffices to show that in every i eligible bins, if $j \geq i$ items need to be evacuated from them, then at least j distinct players have items in at least one of the i bins. Recall that $k = n/t$. In every eligible bin, w.l.o.g. there are items of $k-1$ players (if there are fewer items simply add items of size t of imaginary distinct players). Hence one need only show that $j \leq k-1$. But if $j = k$ then the total volume moved to the auxiliary bin is $kt = n$, contradicting the previous paragraph that showed that it is strictly less than n.

Case 3.2. Consider now the case that some player holds more than c of the $n-1$ largest items. All items of size larger than $n/(c+1)$ can be packed into $c+1$ bins as in Case 3.1. Thereafter, proceed as in the proof of Theorem 24. ■

The advantage of Theorem 28 over the results of Section 5.4.1 is that the upper bound of t need not be imposed on the players. Rather, it is determined empirically based on the reports.

The combination of Proposition 26, Proposition 27 and Theorem 28 establishes Theorem 17.

6 Extensions and discussion

6.1 An alternative normalization for fairness

For Definition 5 of fairness, we required that utility functions be normalized as in Definition 4. Under $\hat{u}$, the normalized version of utility function u, a normalized utility of 1 is the expected utility that a player may receive if all players have the same utility function u. This was referred to as *unanimity normalization*. In this section we present an alternative normalization, that we call *optimistic normalization*.

Given a utility function u_i for player i, consider choosing a finite list of nonnegative sizes of parts $A_1(i), A_2(i), \ldots$ and probabilities $p_1, p_2, \ldots$ subject to the conditions that $\sum p_j = 1$, $\sum_j p_j A_j(i) \leq 1$ (if utility functions are monotone then this condition can be replaced by $\sum_j p_j A_j(i) = 1$), and $\max_j A_j(i) \leq n$, while maximizing $\sum_j p_j u_i(A_j(i))$. Let t be the maximum possible value of $\sum_j p_j u_i(A_j(i))$ under the above constraints. Then under optimistic normalization, the normalized version of u_i is $\hat{u}_i(x) = u_i(x)/t$. In analogy to the case of unanimity normalization, also for optimistic normalization we may think of an expected normalized utility of 1 as the fair share of a player. In analogy to Definition 5 that refers to unanimity-fairness, the following definition refers to optimistic fairness.

Definition 30 *A distribution D over allocations is (optimistically)* fair *with respect to utility functions $(u_1, \ldots, u_n)$ if and only if every player gets exactly the same normalized expected utility, under optimistic normalization (as defined*

above). A randomized allocation rule is fair *if given a tuple* $(u_1, \ldots, u_n)$ *it produces a fair randomized allocation.*

For every utility function u, the scaling factor t involved with optimistic normalization is at least as large as the scaling factor t_u involved with unanimity normalization. This is because both normalization can be thought of as referring to the expected utility that the player can receive from a distribution over allocations that in expectation gives the player at most one unit of good, but unanimity normalization places additional restrictions on the distributions over allocations that a player may choose from: they are required to be a uniform distribution over n allocations. As a consequence, having a ρ-fair allocation with respect to optimistic normalization offers at least as much utility (for every player) as a ρ-fair allocation with respect to unanimity normalization (for the same value of ρ). Hence it is desirable to prove a theorem similar to Theorem 14 with respect to optimistic normalization. It turns out that a small loss in the value of ρ is involved.

Theorem 31 *With respect to optimistic normalization, there is a ρ-fair allocation mechanism with $\rho = 1/2$.*

The following example shows that the value of $\rho = 1/2$ in Theorem 31 is best possible. Suppose that $\hat{u}_1(x) = 1$ for $x \geq 1$ and 0 otherwise, and $\hat{u}_2(x) = 1 + \epsilon$ for $x \geq 1 + \epsilon$ and 0 otherwise. One may verify that both utility functions are normalized (with respect to optimistic normalization, not unanimity normalization). As any allocation gives average normalized utility (over the two players) of no more that $\frac{1+\epsilon}{2}$, the maximum possible value of ρ is $\frac{1+\epsilon}{2}$. Letting ϵ tend to 0 implies that the value of $\rho = 1/2$ is best possible. The example also illustrates that distributions that are fair according to optimistic normalization need not be fair according to unanimity normalization and vice versa (because u_2 is normalized differently in these two cases, whereas u_1 is normalized in the same way).

We shall only sketch the proof of Theorem 31.

Proof.(Sketch.) We first describe the allowable actions of a player i in the randomized allocation mechanism. Player i can choose an arbitrary report of either one value $r_i \leq 1$ or two values (r_{i1}, r_{i2}) satisfying $0 \leq r_{i1} < 1 < r_{i2} \leq n$.

We next describe the intended outcome of the randomized mechanism. If his report was r_i, Player i gets a part of size r_i with probability 1/2, and hence the expected size is at most 1/2. In the report was (r_{i1}, r_{i2}) the player gets a part of size r_{i1} with probability $\frac{r_{i2}-1}{2(r_{i2}-r_{i1})}$ and a part of size r_{i2} with probability $\frac{1-r_{i1}}{2(r_{i2}-r_{i1})}$. The expected size that a player gets in this case can be readily seen to be 1/2.

Given the intended outcome, a player i clearly has a dominant strategy. Namely, the player may choose either $r_i \leq 1$ that offers normalized utility $\hat{u}_i(r_i)/2$, or the two values $0 \leq r_{i1} < 1 < r_{i2} \leq n$ that maximize $\frac{r_{i2}-1}{2(r_{i2}-r_{i1})}\hat{u}_i(r_{i1}) + \frac{1-r_{i1}}{2(r_{i2}-r_{i1})}\hat{u}_i(r_{i2})$, whichever is larger. Moreover, it can be shown that the dominant choice offers normalized utility (with respect to optimistic normalization)

of exactly 1/2. This follows from the fact that only two constraints are involved in the definition of normalized utility ($\sum p_j = 1$ and $\sum_j p_j A_j(i) \leq 1$), and hence to find an optimal solution it suffices that only two of the p_j variables are nonzero.

Finally, we describe a randomized allocation rule that generates the intended outcome. Recall that the good can be divided into parts that are multiples of ϵ and that ϵ divides 1. In particular, this implies that the probabilities $\frac{r_{i2}-1}{2(r_{i2}-r_{i1})}$ and $\frac{1-r_{i1}}{2(r_{i2}-r_{i1})}$ are rational as well. Let N be the least common multiple of $\frac{r_{i2}-r_{i1}}{\epsilon}$ for all i. This leads to the following variant of the bin matching scenario. There are n players. Each player has N items, where for player i all these items have size r_i if his report was r_i, and otherwise $\frac{r_{i2}-1}{r_{i2}-r_{i1}}N$ items have size r_{i1} and $\frac{1-r_{i1}}{r_{i2}-r_{i1}}N$ items have size r_{i2}. All these items need to be legally packed in $2N$ bins of size n, with no two items of the same player in the same bin. This is doable using a proof similar to that of Theorem 14. Pack the N largest items in N distinct bins. For the rest of the items, pack them in the remaining bins using the algorithm SFF. Further details are omitted. ■

6.2 Nonuniform entitlements

We propose ways of extending our definitions to the case that players might have different entitlements for the good. As before, there are n players and the available amount of good is n. With every player i associate a weight $w_i \geq 0$, representing his *entitlement* to the good. For simplicity of notation and without loss of generality, we assume that weights satisfy $\sum_{i=1}^{n} w_i = n$. Given these entitlements, we shall normalize the utility functions. Recall that for the case of uniform entitlements we offered two proposals for normalization: unanimity normalization and optimistic normalization. Each of them extends in a different way to the weighted case.

Optimistic normalization. Given utility function u_i and weight w_i, player i needs to choose a finite list of nonnegative sizes of parts $A_1(i), A_2(i), \ldots$ and probabilities $p_1, p_2, \ldots$ subject to the conditions that $\sum p_j = 1$, $\sum_j p_j A_j(i) \leq w_i$ (if utility functions are monotone then this condition can be replaced by $\sum_j p_j A_j(i) = w_i$), and $\max_j A_j(i) \leq n$, while maximizing $\sum_j p_j u_i(A_j(i))$. Let t be the maximum possible value of $\sum_j p_j u_i(A_j(i))$ under the above constraints. Then under optimistic normalization, the normalized version of u_i is $\hat{u}_i(x) = u_i(x)/t$. Definition 30 of fairness now applies without change, and so does Definition 7 of safety.

Unanimity normalization. Here we propose a normalization of u that depends on the whole vector $(w_1, \ldots, w_n)$ of entitlements. In analogy to the discussion leading up to Definition 4, the normalizing factor t_u with respect to a utility function u and a vector $(w_1, \ldots, w_n)$ is computed as follows: t_u is the maximum value such that there is a distribution D over allocations satisfying $E_{A \leftarrow D}[u(A(i))] = w_i t_u$ for all $1 \leq i \leq n$. Proposition 2 still holds, though now its proof is not as explicit. (To show that distribution D can be assumed to

be supported on only n allocations one may appeal to the fact that it involves only n constraints.) The required equality in Definition 5 of fair randomized allocations is modified to $\frac{E_{A \leftarrow D}[\hat{u}_i(A(i))]}{w_i} = \frac{E_{A \leftarrow D}[\hat{u}_j(A(j))]}{w_j}$. Definition 7 of safety will then require player i to get utility ρw_i.

Determining the maximum possible value of ρ for ρ-fair randomized allocations under nonuniform entitlements is left to future work.

6.3 Pareto optimality

In this section we formally define the notion of Pareto-optimality of allocations and of randomized allocations, and point out that requiring Pareto-optimality would conflict with requiring fairness (under Definition 5).

Definition 32 *An allocation A is* Pareto-optimal *if there is no other allocation in which every player gets at least as much utility as in A and at least one player gets strictly more utility than in A.*

For randomized allocations, there are two possible notions of Pareto-optimality. One is simply a distribution over Pareto-optimal allocations. The other is given in the following definition.

Definition 33 *A randomized allocation D (namely, a distribution over allocations) is* Pareto-optimal *if there is no randomized allocation D' in which every player gets at least as much expected utility as in D and at least one player gets strictly more expected utility than in D.*

A Pareto-optimal randomized allocation is necessarily a distribution over Pareto-optimal allocations, but the converse need not hold. As an example, consider the case in which all utility functions are strictly convex (have positive second derivative). Giving every player a part of size 1 is a Pareto-optimal allocation. However, every player would obtain strictly higher expected utility from a uniform distribution over the following n allocations: one player gets all the good and the other players get nothing.

Pareto-optimality might conflict with our notion of fairness, as the following proposition shows.

Proposition 34 *For $n = 2$, there is a choice of normalized utility functions $\hat{u}_1$ and $\hat{u}_2$ with respect to which there is no fair Pareto-optimal allocation, and no fair Pareto-optimal randomized allocation.*

Proof. Recall that when $n = 2$ utility functions are defined over the domain $0 \leq x \leq 2$. Let $\hat{u}_1(x) = x$, and let $\hat{u}_2(x) = 0$ for $x < 1/2$ and $\hat{u}_2 = 1$ for $x \geq 1/2$. These are indeed normalized utility functions in the sense of Definition 4 (and also in the sense of optimistic normalization as in Section 6.1). There are only two Pareto-optimal allocations, namely $(3/2, 1/2)$ and $(2, 0)$. In both these allocations the utility of Player 1 is strictly higher than that of Player 2. Hence neither one of them is fair in the sense of Definition 5. Likewise, there is no fair

Pareto-optimal randomized allocation D as it would need to be supported over Pareto-optimal allocations. ■

It is an open question whether for some $\rho > 0$ there are ρ-safe Pareto-optimal randomized allocation mechanisms. Theorem 15 establishes a partial result in this direction: it provides a distribution over complete allocations, and in the special case that utility functions are *strictly* monotone a complete allocation is Pareto-optimal. However, Theorem 15 does not guarantee Pareto-optimality when utility functions are monotone rather than strictly monotone, and even if utility functions are strictly monotone, Theorem 15 does not necessarily meet the stronger requirements of Definition 33.

6.4 Attitudes towards lotteries

In this manuscript we assumed that players are expectation maximizers. This aspect is crucial to the claim that when players follow their dominant strategies the expected normalized utility that they get is at least ρ (for a value of ρ that depends on the particular theorem involved). However, this aspect is not crucial if one is only concerned with the question of whether players have a dominant strategy. The nature of the allocation mechanisms that we design is such that even players that are not expectation maximizers still have a dominant strategy. For the mechanisms of Theorems 14 and 15 it suffices to assume that given a utility function, then among all lotteries of a certain class (specifically, involving a uniform distribution over $2n-1$ outcomes, $n-1$ of which are zero, and the remaining n outcomes satisfy the constraint that the sum of sizes received in all outcomes is at most n) there is a lottery that the player prefers at least as much as any other lottery. It does not matter why the player prefers this lottery (whether this is because it maximizes the expected utility, or whether there is some other reason such as minimizing variance). For the mechanism of Theorem 17 one needs a slightly stronger assumption (though still weaker than the assumption that the player maximizes expectation), namely, that the lottery remains preferable regardless of the total number of forced zero outcomes (which could be anything between 1 and $n-1$).

The following example illustrates the above point. Consider a good of size 4, and a player that has a utility function with $u(x) = 0$ for $x < 1$, $u(x) = 1$ for $1 \leq x < 2$, and $u(x) = 2$ for $2 \leq x \leq 4$. In the context of our allocation mechanism and when $n = 4$, the player is asked to propose a partition of the good into four parts (and then the player receives each one of the parts with probability $1/7$, and nothing otherwise). There are three partitions that are equivalent in the sense that they maximize the sum of utilities of the parts. They are $(1,1,1,1)$, $(2,0,2,0)$ and $(1,1,2,0)$. If the player is an expectation maximizer, then all three partitions are equivalent for him. However, there may be players who strictly prefer the partition $(1,1,2,0)$ over the first two. A justification for this preference can be that it avoids the emotional stress involved in having to make a choice: the partition $(1,1,1,1)$ excludes the possibility that the player will be allocated a part of size 2, the partition $(2,0,2,0)$ excludes

the possibility that the player will be allocated a part of size 1, whereas the partition $(1, 1, 2, 0)$ keeps both options open, leaving it to pure chance to decide the outcome. Observe further that a full utility function that strictly prefers the third partition over each of the other two implies that the player cannot be modeled as being an expectation maximizer regardless of what his utility function is (because as a lottery, the third partition is the average of the other two). Nevertheless, such a player has a dominant strategy in our mechanism.

6.5 Iterative allocation

Bin-matching based allocation mechanisms (such as those designed in this paper) ask each player to supply in advance one partition of the good into n parts, and guarantee every player at least an $n/(2n-1)$ fraction of his fair share. In this section we briefly review an iterative mechanism (that appeared in [4]) which may be preferable over bin-matching mechanisms under certain conditions.

The iterative mechanism considers players in a random order. When player i is considered, let D_i be the amount of good that was not yet allocated and n_i the number of players that remain (including i). Then player i gets to partition D_i into n_i parts, and is given one of these parts at random.

The iterative mechanism is inferior to bin-matching mechanisms in two respects.

1. Handling large items. If some player only has utility for very large items (of size t nearly equal to n), then the iterative mechanism does not guarantee the player a constant fraction of his fair-share. There are ways of modifying the iterative mechanism so as to overcome this issue, but they detract from some of the advantages of the iterative mechanism.

2. The communication pattern of the iterative mechanism is more demanding than that of the bin-matching based mechanism. Either it is adaptive (player i is asked for a report only after D_i and n_i are known), or the player needs to supply inputs for all possible combinations of n_i and D_i in advance. If the latter possibility is chosen, then one can consider limiting the expressiveness of players so as to reduce communication, for example, by requiring a player to select at most two item sizes, or by rounding D_i down to the nearest power of two. This somewhat hurts the performance of iterative mechanisms.

On the other hand, iterative mechanisms offer two important advantages.

1. In scheduling scenarios, if items are time allocations for jobs and a job finishes earlier than its allocation, the remaining time is returned to the mechanism and can be included in the time offered to future jobs.

2. If the demands of players happen to be low, iterative mechanisms may serve a higher fraction of the demand than bin-matching based mechanisms. In particular, if all players are unitary all players get their fair

share. Moreover, if some players are "altruistic" (namely, get maximum benefit already from items of size smaller than D_i/n_i, and moreover, are willing to report n_i parts of size smaller than D_i/n_i, allowing other players to enjoy the surplus), other players may get more than their fair share.

In summary, the iterative mechanism is most appropriate when the maximum size t of an item is small compared to the total bin size. In this case it guarantees each player a constant factor of its fair share, and this constant approaches 1 as t approaches 1.

References

[1] R. Banker. Equity considerations in traditional full cost allocation practices: An axiomatic perspective. In *Joint Cost Allocations*, ed. by S. Moriarity, University of Oklahoma Press, pages 110–130, 1981.

[2] S.J. Brams and A.D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[3] I. Caragiannis, J.K. Lai, and A.D. Procaccia. Towards more expressive cake cutting. In *IJCAI-11*, pages 127–132, 2011.

[4] U. Feige and M. Tenneholtz. Mechanism design with uncertain inputs. In *STOC-11*, pages 549–558, 2011.

[5] D. Foley. *Resource allocation and the public sector*. Yale Economic Essays 7, no. I, 45598., 1967.

[6] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–601, 1973.

[7] A. Kumar and J.M. Kleinberg. Fairness measures for resource allocation. *SIAM J. Comput.*, 36(3):657–680, 2006.

[8] H. Moulin. Uniform externalities: Two axioms for fair allocation. *Journal of Public Economics*, 43:305–326, 1990.

[9] H. Moulin. The proportional random allocation of indivisible units. *Social Choice and Welfare*, 19(2):381–413, 2002.

[10] A. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *EC-09*, pages 177–186, 2009.

[11] J.M. Robertson and W.A. Webb. *Cake Cutting Algorithms: Be Fair If You Can.* A. K. Peters, 1998.

[12] J. Schummer and R.V. Vohra. Mechanism design without money. In *Algorithmic Game Theory*, volume 2, pages 110–130. 2007.

[13] W. Thomson and H. Varian. Social goals and social organizations. In Hurwicz et al., editor, *Social goals and social organizations*. Yale Economic Essays 7, no. I, 45598., 1967.