Games and Economic Behavior 77 (2013) 284–297

Contents lists available at SciVerse ScienceDirect

Games and Economic Behavior

www.elsevier.com/locate/geb

Truth, justice, and cake cutting [☆]

Yiling Chen [a], John K. Lai [a], David C. Parkes [a,*], Ariel D. Procaccia [b]

[a] *School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, United States*
[b] *Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States*

ARTICLE INFO

Article history:
Received 11 October 2010
Available online 1 November 2012

JEL classification:
C78
D63
D82

Keywords:
Fair division
Envy-freeness
Cake cutting
Maximum flows
Strategy-proofness
Mechanism design

ABSTRACT

Cake cutting is a common metaphor for the division of a heterogeneous divisible good. There are numerous papers that study the problem of fairly dividing a cake; a small number of them also take into account self-interested agents and consequent strategic issues, but these papers focus on fairness and consider a strikingly weak notion of truthfulness. In this paper we investigate the problem of cutting a cake in a way that is truthful, Pareto-efficient, and fair, where for the first time our notion of dominant strategy truthfulness is the ubiquitous one in social choice and computer science. We design both deterministic and randomized cake cutting mechanisms that are truthful and fair under different assumptions with respect to the valuation functions of the agents.

© 2012 Elsevier Inc. All rights reserved.

> Superman: “I’m here to fight for truth, justice, and the American way.”
> Lois Lane: “You’re gonna wind up fighting every elected official in this country!”
>
> *Superman (1978)*

1. Introduction

Cutting a cake is often used as a metaphor for allocating a divisible good. The difficulty is not cutting the cake into pieces of equal size, but rather that the cake is not uniformly tasty: different agents prefer different parts of the cake, depending, e.g., on whether the toppings are strawberries or cookies. The goal is to divide the cake in a way that is “fair”; the definition of fairness is a nontrivial issue in itself, which we discuss in the sequel. The cake cutting problem dates back to the work of Steinhaus in the 1940s (Steinhaus, 1948), and for over sixty years has attracted the attention of mathematicians, economists, political scientists, and more recently, computer scientists (Edmonds and Pruhs, 2006a, 2006b; Procaccia, 2009).

[☆] A significantly shorter version of this paper appeared in the proceedings of AAAI 2010. Here we fill in missing proofs, discuss the Lorenz dominance of our allocation, and provide an extended discussion of related work and future directions. The paper was also presented in the Third International Workshop on Computational Social Choice in Düsseldorf, Germany, September 2010, at a workshop on prior-free mechanism design in Guanajuato, Mexico, May 2010, and in the Harvard EconCS seminar, February 2010.

[*] Corresponding author.
E-mail addresses: yiling@eecs.harvard.edu (Y. Chen), jklai@post.harvard.edu (J.K. Lai), parkes@eecs.harvard.edu (D.C. Parkes), arielpro@cs.cmu.edu (A.D. Procaccia).

0899-8256/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.geb.2012.10.009

Slightly more formally, the cake is represented by the interval $[0, 1]$. Each of n agents has a valuation function over the cake, which assigns a value to every given piece of cake and is additive. The goal is to find a partition of the cake among the agents (while possibly throwing a piece away) that satisfies one or several fairness criteria. In this paper we consider the two most prominent criteria. A *proportional* allocation is one where the value each agent has for its own piece of cake is at least $1/n$ of the value it assigns to the entire cake. An *envy-free (EF)* allocation is one where the value each agent assigns to its own piece of cake is at least as high as the value it assigns to any other agent's piece of cake. There is a rather large body of literature on fairly cutting a cake according to these two criteria (see, e.g., the books by Robertson and Webb, 1998 and Brams and Taylor, 1996).

So far we have briefly discussed "justice", but have not yet mentioned "truth". Taking the game-theoretic point of view, an agent's valuation function is its private information, which is reported to a cake cutting mechanism. We would like a mechanism to be *truthful*, in the sense that agents are motivated to report their true valuation functions. Like fairness, this idea of truthfulness also lends itself to many interpretations. One variation, referred to as *strategy-proofness* in previous papers by Brams et al. (2006, 2008), assumes that an agent will report its true valuation rather than lie if there *exist* valuations of the other agents such that reporting truthfully yields at least as much value as lying. In the words of Brams et al., "the players are risk-averse and never strategically announce false measures if it does not guarantee them more-valued pieces. ... Hence, a procedure is strategy-proof if no player has a strategy that dominates his true value function" (Brams et al., 2008, p. 362).

The foregoing notion is strikingly weak compared to the notion of truthfulness that is common in the social choice literature. Indeed, strategy-proofness is usually taken to mean that an agent can *never* benefit by lying, that is, *for all* valuations of the other agents reporting truthfully yields at least as much value as lying. Put another way, truth-telling is a dominant strategy. This notion is worst-case, in the sense that an agent cannot benefit by lying even if it is fully knowledgeable of the valuations of the other agents. In order to prevent confusion we will avoid using the term "strategy-proof", and instead refer to the former notion of Brams et al. as "weak truthfulness" and to the latter standard notion as "truthfulness".

To illustrate the difference between the two notions, consider the most basic cake cutting mechanism for the case of two agents, the *Cut and Choose* mechanism.[1] Agent 1 cuts the cake into two pieces that are of equal value according to its valuation; agent 2 then chooses the piece that it prefers, giving the other piece to agent 1. This mechanism is trivially proportional and EF.[2] It is also weakly truthful, as if agent 1 divides the cake into two pieces that are unequal according to its valuation then agent 2 may prefer the piece that is worth more to agent 1. Agent 2 clearly cannot benefit by lying. The mechanism is even truthful (in the strong sense) if the agents have the same valuation function. However, the mechanism is not truthful in general. Indeed, consider the case where agent 1 would simply like to receive as much cake as possible, whereas the single-minded agent 2 is only interested in the interval $[0, \epsilon]$ where ϵ is small (for example, it may only be interested in the cherry). If agent 1 follows the protocol it would only receive half of the cake. Agent 1 can do better by reporting that it values the intervals $[0, \epsilon]$ and $[\epsilon, 1]$ equally.

In this paper we consider the design of truthful and fair cake cutting mechanisms. We will design mechanisms where the agents report valuations and the mechanism, taking the role of a direct revelation mechanism, determines the cuts of the cake. However, there is a major obstacle that must be circumvented: regardless of strategic issues, and when there are more than five agents, even finding a proportional and EF allocation in a bounded number of steps with a deterministic mechanism is a long-standing open problem! See Procaccia (2009) for an up-to-date discussion.[3] We shall therefore restrict ourselves to specific classes of valuation functions where efficiently finding fair allocations is a non-issue; the richness of our problem stems from our desire to additionally achieve truthfulness.

In particular, we will focus on the case where agents hold *piecewise uniform, piecewise constant, and piecewise linear* valuation functions. Piecewise uniform valuation functions capture the case where each agent is interested in a collection of subintervals of $[0, 1]$ and has the same marginal value for each fractional piece in each desired subinterval. They can be thought of as dichotomous preferences since intervals can be separated into acceptable and unacceptable intervals, and the acceptable intervals all have the same marginal utility. While these valuations are restrictive, they are expressive enough to capture some realistic settings. Piecewise uniform valuations capture the case when some parts of the cake satisfy a certain property that an agent cares about and an agent desires as much of these parts as possible. An example of such a setting is a shared computing server. Each researcher has certain times during which the server is not useful because of scheduling constraints or waiting times for experiments, but outside of these times, the researcher would like as much time on the computing server as possible. Piecewise constant and linear valuation functions capture the case where an agent's marginal value is constant and linear respectively, and are significantly more expressive. In particular, these valuation functions can approximate any valuation function.

[1] This mechanism is described here with the agents taking actions; equivalently, the mechanism acts on behalf of agents using the reported valuations.

[2] Proportionality and envy-freeness coincide if there are two agents and the entire cake is allocated.

[3] To be precise, previous work assumed that the entire cake has to be allocated, but this does not seem to be a significant restriction in the context of fairness.

Our results In Section 3 we consider deterministic mechanisms when agent valuations are piecewise uniform. Our main result is a deterministic mechanism for any number of agents that is truthful, proportional, EF, Lorenz dominant in a certain sense, Pareto-efficient, and polynomial time when agents have piecewise uniform valuations.

To gain intuition for our mechanism, it is insightful to examine the special case of two agents. In the two agent mechanism, the crux of the allocation is how we should allocate intervals that both agents desire. We can throw away intervals that are undesired by any agent, and allocate the intervals that are only desired by a single agent to that agent. Our mechanism then allocates the intervals desired by both agents to try to equalize the total lengths given to each agent. If the length of intervals only desired by agent i exceeds the length of intervals only desired by agent j, then agent i will be given a smaller share of the mutually desired intervals. To see how this encourages truthfulness, consider the deviation where an agent misstates and increases the lengths of intervals only desired by that agent. The agent receives all of these intervals (which include intervals it does not actually desire), but obtains a smaller share of the mutually desired intervals. It turns out that the equalization of lengths removes the incentive for agents to misstate their valuations.

In the general mechanism for n agents, our mechanism examines what the fairness requirement means for the allocation given to different agents. Agents who receive a small amount of the cake due to fairness requirements are handled first and given the best possible fair allocation. As an example of agents who might be handled first, there could be a set of agents who all desire the same small interval of the cake. For fairness to be maintained, each of these agents must receive the same fraction of this small interval. These agents would be selectively handled earlier by the mechanism. By protecting the disadvantaged agents in this way, the mechanism removes the incentive to report larger intervals than those that are truly desired. Our mechanism is highly dependent on the assumption that agents hold piecewise uniform valuations and there is no easy way to generalize our techniques to piecewise constant valuations.

In Section 4 we consider randomized mechanisms. We slightly relax truthfulness by asking that the mechanism be *truthful in expectation*, that is, an agent cannot hope to increase its expected value by lying for any reports of other agents. For general valuations, we present a simple randomized mechanism that is truthful in expectation, and is *ex-post* proportional and EF. Our result relies on the existence of *perfect partitions*, which are partitions of the cake where every agent has value exactly $1/n$ for every element of the partition. Given a perfect partition, the randomized mechanism constructs an allocation by indexing the elements of the partition, and allocating the elements to agents based on a randomly drawn permutation. The perfect partition guarantees fairness, while truthfulness in expectation is preserved because the agent's expected value is a function of the agent's value for the entire cake and not of the specific partition. While perfect partitions are known to exist, there is no known algorithm for constructing perfect partitions for general valuations. We provide explicit constructions of perfect partitions when agents hold *piecewise linear* valuations.

Related work We have recently learned of independent parallel work by Mossel and Tamuz (2010), who ask similar questions about truthful and fair cake cutting. However, their work only considers randomized mechanisms, and they provide existence results rather than concrete mechanisms for specific classes of valuations. They also focus solely on proportionality rather than proportionality and envy-freeness together. Under general assumptions on valuation functions (which are essentially the same as our definition of valid valuations), they show that there exists a mechanism that is truthful in expectation and always guarantees each agent a value of more than $1/n$. The results are then extended to the case of indivisible goods. The technical overlap between the two papers is very small; we refer the reader's attention to this overlap in a footnote in Section 4.

Thomson (2007) showed that a truthful and Pareto-efficient mechanism must be dictatorial in the slightly different setting of pie cutting. In pie cutting, the pie is modeled as a circular object and the feasible cuts are wedges, while in cake cutting the cake is modeled as the interval [0,1] and feasible cuts are subintervals. Though the two settings appear similar and even possibly equivalent, they are actually distinct and results for one setting do not readily carry over to the other. Brams et al. (2008) provide a more in depth discussion. Note that Pareto-efficiency is not a fairness property and neither implies, nor is implied by, envy-freeness or proportionality.

Our deterministic mechanism is closely related to the *egalitarian solution* studied by Dutta and Ray (1989) for cooperative games and Bogomolnaia and Moulin (2004) for the random assignment problem with dichotomous preferences. It is also related to the *probabilistic serial* mechanism studied by Bogomolnaia and Moulin (2001) and Katta and Sethuraman (2006) for the random assignment problem. However, the cake cutting setting is distinct from the cooperative games setting and the random assignment problem, so though our mechanism uses similar ideas and techniques, our results are not implied by these earlier results.

Dutta and Ray (1989) propose the *egalitarian solution* for cooperative games with transferable utility. As cake cutting is not a transferable utility setting, the connection to Dutta and Ray (1989) involves defining a convex, cooperative game when agents have piecewise uniform valuations in the cake cutting setting. Our deterministic mechanism finds an allocation that is closely related to the egalitarian solution, and our algorithm for finding the solution also resembles that proposed in this earlier work. We expand on this connection in Section 3.5.3, where we use it to show that our deterministic mechanism finds an allocation that is Lorenz dominant in a certain sense.

Our mechanism also resembles the mechanisms proposed by Bogomolnaia and Moulin (2001), Katta and Sethuraman (2006), and Bogomolnaia and Moulin (2004) for the random assignment problem under different assumptions on preferences. Bogomolnaia and Moulin (2001) study the setting where agents have strict ordinal preferences, Katta and Sethuraman (2006) extend this to the full preference domain (where indifferences are allowed), and Bogomolnaia and Moulin (2004) study the case when agents have dichotomous preferences (i.e., items are either acceptable or unacceptable and agents are indifferent between any two acceptable items).

Because we study cake cutting when agents have piecewise uniform valuations, we can relate the cake cutting setting to the random assignment setting as follows. Given agents' reports, make a mark at the beginning of and end of each agent's desired intervals. Construct an item for every subinterval formed by consecutive marks. Giving an agent an item with probability p is the same as allocating the agent a p-fraction of the associated subinterval. While this transformation converts cake cutting into a discrete allocation problem, there are fundamental differences between our setting and the random assignment setting. First, in our setting, agents' valuations affect the items that are created, whereas in the random assignment problem items are assumed to be given. Second, in our setting, agents want as many items as possible, while the random assignment setting assumes that each agent wants at most one item. In addition to these two general differences, there are differences specific to each of the previously studied settings. If we look specifically at dichotomous preferences studied by Bogomolnaia and Moulin (2004), our agents may not have dichotomous preferences for items created based on agent reports; that is, some subintervals might be larger than others and agents, though they have piecewise uniform valuations, might not have the same value for all desired items. Even if we discretize the cake into many tiny items (to try and equalize agent values for desired items), our agents still desire as many of the created items as possible. If we look specifically at strict ordinal preferences studied by Bogomolnaia and Moulin (2001), our agents may be indifferent between the items formed by subintervals between consecutive marks. On the other hand, our agents cannot hold arbitrary ordinal preferences over subintervals between consecutive marks, since if two agents desire two subintervals, both agents would value the longer subinterval more than the shorter. This last difference also persists under the full preference domain studied by Katta and Sethuraman (2006). Because of these differences, our results cannot be derived from these earlier results, despite the underlying dependence on similar ideas.

The network flow ideas that we use to prove properties of our mechanism are similar to those used by Katta and Sethuraman (2006) to show that the egalitarian assignment solution can be computed in polynomial time. Interestingly, Katta and Sethuraman (2006) note that the egalitarian assignment solution is identical to another independent mechanism for finding a lexicographically optimal flow in a network due to Megiddo (1979).

2. Preliminaries

We consider a heterogeneous cake, represented by the interval $[0,1]$. A *piece of cake* is a *finite* union of subintervals of $[0,1]$. We sometimes abuse this terminology by treating a piece of cake as the set of the (inclusion-maximal) intervals that it contains. The length of the interval $I=[x,y]$, denoted $\text{len}(I)$, is $y-x$. For a piece of cake X we denote $\text{len}(X)=\sum_{I\in X}\text{len}(I)$.

The set of agents is denoted $N=\{1,\dots,n\}$. Each agent $i\in N$ holds a private valuation function V_i, which maps given pieces of cake to the value agent i assigns them. Formally, each agent i has a *value density function*, $v_i:[0,1]\to[0,\infty)$, that is piecewise continuous. The function v_i characterizes how agent i assigns value to different parts of the cake. The value of a piece of cake X to agent i is then defined as

$$V_i(X)=\int\limits_X v_i(x)\,dx=\sum_{I\in X}\int\limits_I v_i(x)\,dx.$$

We note that the valuation functions are *additive*, i.e. for any two disjoint pieces X and Y, $V_i(X\cup Y)=V_i(X)+V_i(Y)$, and *non-atomic*, that is $V_i([x,x])=0$ for every $x\in[0,1]$. The last property implies that we do not have to worry about the boundaries of intervals, i.e., open and closed intervals are identical for our purposes. We further assume that the valuation functions are *normalized*, i.e. $V_i([0,1])=\int_0^1 v_i(x)\,dx=1$.

A cake cutting mechanism is a function f from the valuation function of each agent to an allocation $(A_1,\dots,A_n)$ of the cake such that the pieces A_i are pairwise disjoint. For each $i\in N$ the piece A_i is allocated to agent i, and the rest of the cake, i.e., $[0,1]\setminus\bigcup_{i\in N}A_i$, is thrown away. Here we are assuming *free disposal*, that is, the mechanism can throw away resources without incurring a cost.

We say that an allocation $A_1,\dots,A_n$ is *proportional* if for every $i\in N$, $V_i(A_i)\geqslant 1/n$, that is, each agent receives at least a $(1/n)$-fraction of the cake according to its own valuation. We say that an allocation is *envy-free (EF)* if for every $i,j\in N$, $V_i(A_i)\geqslant V_i(A_j)$, i.e., each agent prefers its own piece of cake to the piece of cake allocated to any other agent. A proportional (resp., EF) cake cutting mechanism always returns a proportional (resp., EF) allocation.

Note that when $n=2$ proportionality implies envy-freeness. Indeed, $V_i(A_i)+V_i(A_{3-i})\leqslant 1$, and hence if $V_i(A_i)\geqslant 1/2$ then $V_i(A_{3-i})\leqslant 1/2$. Under the free disposal assumption the converse is not true. For example, an allocation that throws away the entire cake is EF but not proportional. In general, when $n>2$ proportionality neither implies nor is implied by envy-freeness.[4]

[4] If free disposal is not assumed, that is, the entire cake is allocated, then envy-freeness implies proportionality for any n.

An allocation $A'_1, \ldots, A'_n$ *Pareto-dominates* $A_1, \ldots, A_n$ if $V_i(A'_i) \geqslant V_i(A_i)$ for every i, with the inequality strict for some i. An allocation $A_1, \ldots, A_n$ is *Pareto-efficient* if there is no allocation that Pareto-dominates $A_1, \ldots, A_n$.

A cake cutting mechanism f is *truthful* if when an agent lies it is allocated a piece of cake that is worth, according to its real valuation, no more than the piece of cake it was allocated when reporting truthfully. Formally, denote $A_i = f_i(V_1, \ldots, V_n)$, and let $\mathcal{V}$ be a class of valuation functions. The mechanism f is truthful if for every agent i, every collection of valuations functions $V_1, \ldots, V_n \in \mathcal{V}$, and every $V'_i \in \mathcal{V}$, it holds that $V_i(f_i(V_1, \ldots, V_n)) \geqslant V_i(f_i(V_1, \ldots, V_{i-1}, V'_i, V_{i+1}, \ldots, V_n))$.

Our results require that agent valuations be elements of a restricted family of valuation functions. In particular, we define the families of piecewise linear, piecewise constant, and piecewise uniform valuation functions. We say that a valuation function V_i is *piecewise linear* if its corresponding value density function v_i is piecewise linear, that is $[0, 1]$ can be partitioned into a finite number of intervals such that v_i is linear on each interval. We define piecewise constant valuations in the same way, requiring that densities be constant rather than linear. We say that V_i is *piecewise uniform* if V_i is piecewise constant and moreover v_i is either some constant $c \in \mathbb{R}_+$ (the same one across intervals) or zero. Piecewise uniform valuations can be thought of as dichotomous valuations where intervals are either acceptable or unacceptable, all acceptable intervals have the same marginal utility, and all unacceptable intervals have marginal utility 0. We call the intervals that the agent finds acceptable its *reference piece of cake*, denoted by U_i. Because we require densities to be normalized, U_i pins down the density function of the agent:

$$v_i(x) = \begin{cases} \frac{1}{\mathrm{len}(U_i)} & \text{if } x \in U_i, \\ 0 & \text{otherwise.} \end{cases}$$

Given a piece of cake X, it holds that $V_i(X) = \mathrm{len}(X \cap U_i)/\mathrm{len}(U_i)$. From the computational perspective, the size of the input to the cake cutting mechanism is the number of bits that define the boundaries of the intervals in the agents' reference pieces of cake.

Our deterministic mechanism requires piecewise uniform valuations, while our randomized mechanism allows for piecewise linear valuations. We believe that piecewise uniform valuations are very natural. An agent would have such a valuation function if it is simply interested in pieces of the good that have a certain property, e.g., a child only likes portions of the cake that have chocolate toppings, and wants as much cake with chocolate toppings as possible. As a more practical example, assume that the cake represents access time to a backup server. Each agent is equally interested in time intervals when its computer is idle or when it is not modifying its data.

3. Deterministic mechanisms

In order to attain a truthful, deterministic mechanism, we focus on the restricted family of piecewise uniform valuations. The restricted family is still rich enough to make the problem challenging and capture real world situations, and we believe that our results provide a foundation for future work on truthful cake cutting. Indeed, it remains an open question whether truthful, deterministic mechanisms exist for even the slightly richer family of piecewise constant valuations. In the rest of this section, we assume that valuations are piecewise uniform.

3.1. A deterministic mechanism

Before introducing our mechanism we present some required notation. Let $S \subseteq N$ be a subset of agents and let X be a piece of cake. Let $D(S, X)$ denote the portions of X that are valued by at least one agent in S. Formally, $D(S, X) = (\bigcup_{i \in S} U_i) \cap X$, and is itself a union of intervals.

Let $\mathrm{avg}(S, X) = \mathrm{len}(D(S, X))/|S|$ denote the average length of intervals in X desired by at least one agent in S. We say that an allocation is *exact* with respect to S and X if it allocates to each agent in S a piece of cake of length $\mathrm{avg}(S, X)$ comprised *only* of desired intervals. Clearly this requires allocating all of $D(S, X)$ since the total length of allocated intervals is $\mathrm{avg}(S, X) \cdot |S| = \mathrm{len}(D(S, X))$. Suppose $S = \{1, 2\}$ and $X = [0, 1]$: if $U_1 = U_2 = [0, 0.2]$ then agents 1 and 2 receiving $[0, 0.1]$ and $[0.1, 0.2]$ respectively is an exact allocation; but if $U_1 = [0, 0.2]$, $U_2 = [0.3, 0.7]$ then there is no exact allocation.

The deterministic mechanism for n agents with piecewise uniform valuations is a recursive mechanism that finds a subset of agents with a certain property, makes the allocation decision for that subset, and then makes a recursive call on the remaining agents and the remaining intervals. Specifically, for a given set of agents $S \subseteq N$ and a remaining piece of cake to be allocated X, we find the subset $S' \subseteq S$ of agents with the smallest $\mathrm{avg}(S', X)$. We then give an exact allocation of $D(S', X)$ to S'. We recurse on $S \setminus S'$ and the intervals not desired by any agent in S', i.e. $X \setminus D(S', X)$. The pseudocode of the mechanism is given as Mechanism 1.

Mechanism 1 $(V_1, \ldots, V_n)$

```
1. SubRoutine($\{1, \ldots, n\}, [0,1], (V_1, \ldots, V_n)$)

SubRoutine($S, X, V_1, \ldots, V_n$):

1. If $S = \emptyset$, return.
2. Let $S_{\min} \in \operatorname{argmin}_{S' \subseteq S} \operatorname{avg}(S', X)$ (breaking ties arbitrarily).
3. Let $E_1, \ldots, E_n$ be an exact allocation with respect to $S_{\min}, X$ (breaking ties arbitrarily). For each $i \in S_{\min}$, set $A_i = E_i$.
4. SubRoutine($S \setminus S_{\min}, X \setminus D(S_{\min}, X), (V_1, \ldots, V_n)$).
```

In particular, Steps 2 and 3 of SubRoutine imply that if $S = \{i\}$ then $A_i = D(S, X)$. For example, suppose $X = [0,1]$, $U_1 = [0, 0.1]$, $U_2 = [0, 0.39]$, and $U_3 = [0, 0.6]$. In this case, the subset with the smallest average is $\{1\}$, so agent 1 receives all of $[0, 0.1]$ and we recurse on $\{2,3\}$, $[0.1, 1]$. In the recursive call, set $\{2\}$ has average $0.39 - 0.1 = 0.29$, set $\{3\}$ has average $0.6 - 0.1 = 0.5$, and set $\{2,3\}$ has average $(0.6 - 0.1)/2 = 0.25$. As a result, the entire set $\{2,3\}$ is chosen as the set with smallest average, and an exact allocation of $[0.1, 1.0]$ is given to agents 2 and 3. One possible allocation is to give agent 2 $[0.1, 0.35]$ and agent 3 $[0.35, 0.6]$. Note that if agent 1 uniformly values $[0, 0.2]$ instead, the first call would choose $\{1,2\}$ as the subset with the smallest average, equally allocating $[0, 0.39]$ between agents 1 and 2 and giving the rest, $[0.39, 0.6]$, to agent 3.

Our goal in the rest of this section is to prove the following theorem.

Theorem 3.1. *Assume that the agents have piecewise uniform valuation functions. Then Mechanism 1 is truthful, proportional, EF, Lorenz dominant (in a certain sense), Pareto-efficient, and polynomial time.*

To prove the theorem we exploit a connection to network flow in Section 3.3. As explained in Section 1, this technique is a simple generalization of related results in the economics and operations research literature, but we include some of the details for completeness. Our main technical contributions in this section are the truthfulness and fairness of Mechanism 1, which are established in Section 3.5.

3.2. The two agent mechanism

To gain intuition for the general case of Theorem 3.1 we first describe the special case of two agents. Note that designing truthful, proportional and EF mechanisms even for this case is nontrivial. To see the difficulty, consider an intuitive first attempt at a proportional and EF mechanism. We have already seen that Cut and Choose is not truthful for two agents. Another straightforward approach would be to mark the end points of all submitted intervals and divide every resulting subinterval equally between the two agents. One possibility is to always give the left half of each subinterval to agent 1 and the right half to the agent 2. This mechanism is clearly proportional and EF since every agent receives value exactly 1/2. However, it is not truthful due to a simple example. Under this mechanism, if both agents value the entire cake, agent 1 receives $[0, 0.5]$ and the agent 2 receives $[0.5, 1]$. Suppose that agent 1's true valuation consists of only $[0, 0.5]$. If it reports truthfully, it receives $[0, 0.25], [0.5, 0.75]$ which gives value 0.5. The agent can gain by instead reporting that it values all of $[0,1]$ and receive $[0, 0.5]$ which gives it value 1. In particular, suppose that when both agents report $[0,1]$, agent 1 receives $[0, 0.5]$. In order for the mechanism to be truthful, whenever agent 1 reports some set of subintervals of $[0, 0.5]$ and agent 2 reports $[0,1]$, the mechanism must allocate to agent 1 all of this agent's desired intervals.

We now discuss our truthful Mechanism 1 for the case of two agents. Recall that U_i denotes the reference piece of cake of agent $i \in N$, and for $i \in \{1,2\}$ let $W_i = U_i \setminus U_{3-i}$, and $W_{12} = U_1 \cap U_2$. W_1, W_2, W_{12} are disjoint, and this will allow us to interchangeably write $\operatorname{len}(W_1) + \operatorname{len}(W_2) + \operatorname{len}(W_{12})$ and $\operatorname{len}(W_1 \cup W_2 \cup W_{12})$.

For two agents, the decision of the mechanism hinges on the choice of $S_{\min}$ in the first call to SubRoutine. There are three possibilities for $S_{\min}$: $\{1\}$, $\{2\}$, $\{1,2\}$. If $S_{\min} = \{i\}$, then agent i receives $U_i = W_i \cup W_{12}$ and agent $3 - i$ receives $U_{3-i} \setminus U_i = W_{3-i}$. On the other hand, if $S_{\min} = \{1,2\}$, then the mechanism gives an exact allocation with each agent receiving equal lengths. Agent i receives all of W_i and W_{12} is split so that the total lengths for each agent are equal.

3.2.1. Properties of the two agent mechanism

To prove that the mechanism for two agents is well-defined, fair, Pareto-efficient, and truthful,[5] it is useful to introduce variables $\delta_i = (\operatorname{len}(W_{12}) - \operatorname{len}(W_i) + \operatorname{len}(W_{3-i}))/2$ for each agent. It holds that $\delta_1 + \delta_2 = \operatorname{len}(W_{12})$ and $\delta_1 + \operatorname{len}(W_1) = \operatorname{len}(W_{12} \cup W_1 \cup W_2)/2 = \delta_2 + \operatorname{len}(W_2)$. Qualitatively, δ_i measures the share of W_{12} that agent i is entitled to due to the size of W_i. Smaller values of $\operatorname{len}(W_i)$ give the agent a larger claim to W_{12} and increase δ_i. We can relate values of these variables to the choices of $S_{\min}$ made by the mechanism.

[5] We defer a discussion of Lorenz dominance to the proof for the general case with n agents.

By definition of W_i and W_{12}, $\text{avg}(\{1\}) = \text{len}(W_1 \cup W_{12})$, $\text{avg}(\{2\}) = \text{len}(W_2 \cup W_{12})$, $\text{avg}(\{1,2\}) = \text{len}(W_1 \cup W_2 \cup W_{12})/2$. There are three cases:

1. $\delta_1 < 0$, $\delta_2 > \text{len}(W_{12})$. Since $\text{len}(W_1) + \delta_1 = \text{len}(W_2) + \delta_2$, $\text{len}(W_1) > \text{len}(W_2)$ and therefore $\text{avg}(\{1\}) > \text{avg}(\{2\})$. Moreover, using $\delta_2 > \text{len}(W_{12})$, we have
$$\frac{\text{len}(W_{12}) - \text{len}(W_2) + \text{len}(W_1)}{2} > \text{len}(W_{12}),$$
and it follows that
$$\frac{\text{len}(W_{12}) + \text{len}(W_2) + \text{len}(W_1)}{2} > \text{len}(W_{12}) + \text{len}(W_2),$$
that is, $\text{avg}(\{1,2\}) > \text{avg}(\{2\})$.
This case corresponds to $S_{\min} = \{2\}$, and the mechanism gives all of $W_2 \cup W_{12}$ to agent 2 and all of W_1 to agent 1.
2. $0 \leqslant \delta_1 \leqslant \text{len}(W_{12})$, $0 \leqslant \delta_2 \leqslant \text{len}(W_{12})$. We can use the same arguments as the previous case with the inequalities flipped to show that $\text{avg}(\{1,2\}) < \text{avg}(\{1\})$ and $\text{avg}(\{1,2\}) < \text{avg}(\{2\})$. This case corresponds to $S_{\min} = \{1,2\}$, and the mechanism gives an exact allocation of $W_1 \cup W_2 \cup W_{12}$ to the agents.
3. $\delta_2 < 0$, $\delta_1 > \text{len}(W_{12})$. The analysis is the same as case 1, with agents 1 and 2 changing roles. This corresponds to $S_{\min} = \{1\}$. Agent 1 receives all of $W_1 \cup W_{12}$ and agent 2 receives all of W_2.

First, we show that Mechanism 1 is well-defined. Mechanism 1 calls for an exact allocation to be made to $S_{\min}$ at each step, but this may not always be possible. When $|S_{\min}| = 1$ an exact allocation is trivial, but when $S_{\min} = \{1,2\}$, we need to guarantee that we can exactly split $D(\{1,2\},[0,1])$. Because $S_{\min} = \{1,2\}$ only when $0 \leqslant \delta_1, \delta_2 \leqslant \text{len}(W_{12})$, this is possible by giving W_i and δ_i of W_{12} to agent i.

Next, we establish the fairness properties of proportionality and envy-freeness. If $S_{\min} = \{1,2\}$, proportionality and envy-freeness are clear. If $S_{\min} = \{1\}$, then agent 1 receives all of its desired intervals so its value is 1. Agent 2 receives only W_2, but since $S_{\min} = \{1\}$, $\text{len}(W_1 \cup W_{12}) \leqslant \text{len}(W_1 \cup W_{12} \cup W_2)/2$ which implies $\text{len}(W_{12}) \leqslant \text{len}(W_2)$, and hence agent 2 receives value at least $1/2$ and is not envious of agent 1.

Pareto-efficiency is easy to see because in all cases, we allocate W_i to agent i and fully allocate W_{12}, so it is not possible to give more lengths to one agent without decreasing the allocation of the other.

In order to establish Theorem 3.1 for the two agent case it remains to show truthfulness. We will take the point of view of agent $i \in \{1,2\}$. If it holds that $\delta_i > \text{len}(W_{12})$ then the agent receives all desired intervals and has no incentive to deviate, hence we can assume that $\delta_i \leqslant \text{len}(W_{12})$.

Note that an agent always receives all of W_i, so profitable manipulations will try to obtain more of W_{12} by increasing δ_i while not losing too much of W_i. Also note that by definition of δ_i, an increase or decrease of $\text{len}(W_i)$ by k will respectively decrease or increase δ_i by $k/2$.

Suppose that agent i reports $U'_i \neq U_i$, inducing new pieces W'_{12}, W'_1, W'_2. Before manipulating the agent receives $\text{len}(W_i) + \max(\delta_i, 0)$. Note that
$$\text{len}\big(W'_{12}\big) + \text{len}\big(W'_{3-i}\big) = \text{len}(U_{3-i}) = \text{len}(W_{12}) + \text{len}(W_{3-i}),$$
that is, this sum is not affected by the report of agent i.

If $\text{len}(W'_1) = \text{len}(W_1)$ then δ_i is unchanged, and the agent receives the same length of intervals (though the actual intervals received may not be desired). If $\text{len}(W'_i) = \text{len}(W_i) - k$ then the agent loses at least k of W_i and gains at most $k/2$ from an increase in δ_i; this is not profitable. Finally, if $\text{len}(W'_i) = \text{len}(W_i) + k$, then the agent gains undesired intervals of length k and δ_i is weakly smaller, so the agent does not gain any more of W_{12} by deviating.

3.2.2. *An interpretation of the two agent mechanism*

When $|S| = 2$, the mechanism is equivalent to a swapping procedure. The proof of this equivalence is postponed to Appendix A. As before, let W_1, W_2, W_{12} describe the intervals that only agent 1 desires, only agent 2 desires, and both agents desire. Discard the intervals that neither agent desires, and give an initial grant of half of W_1, W_2, W_{12} to each agent. Assume without loss of generality that $\text{len}(W_1) \leqslant \text{len}(W_2)$. The swapping procedure can be described as follows:

1. Swap pieces X, Y of equal length where agent 1 owns X, agent 2 owns Y, $X \subseteq W_2$, $Y \subseteq W_1$.
2. Swap pieces X, Y of equal length where agent 1 owns X, agent 2 owns Y, $X \subseteq W_2$, $Y \subseteq W_{12}$.
3. If there are still pieces of W_2 owned by agent 1, give these intervals to agent 2.

This procedure first gives each agent an equal piece of each kind of interval. The first swaps performed are mutually beneficial: each agent gives pieces of the other agent's desired intervals in exchange for pieces of its own desired intervals. However, eventually, agent 1 obtains all of W_1 and these swaps no longer exist. The second swaps involve trades where agent 2 receives remaining pieces of W_2 in exchange for giving up pieces of W_{12}. This trade does not improve the utility of agent 2, but does improve the utility of agent 1. If agent 2 gives away all of its share of W_{12}, then agent 1 receives

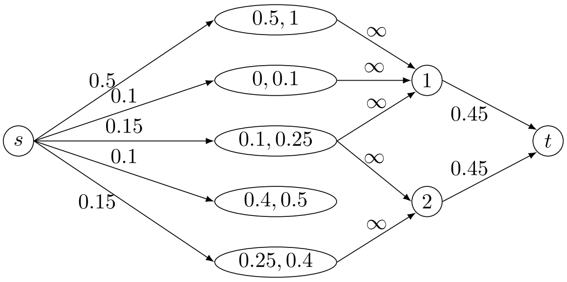

Fig. 1. The flow network induced by the example.

all of W_1 and W_{12}, and agent 2 is given all of W_2 due to step 3 which gives the remaining shares of W_2 to agent 2 for free. If agent 2 obtains all of W_2 without relinquishing all of its share of W_{12}, then agents 1 and 2 split W_{12} (potentially unequally), and each agent i receives W_i.

3.3. Exact allocations and maximum flows

Having defined Mechanism 1 and shown that it has the desired properties in the case of two agents, we now generalize the proofs to n agents. Before turning to properties of truthfulness and fairness, we point out that, as in the two agent case, it is unclear whether Mechanism 1 is well-defined. In particular, the mechanism requires an exact allocation E with respect to the subset $S_{\min}$ and X, but it remains to show that such an allocation exists, and to provide a way to compute it. To this end we exploit a close relationship between exact allocations and maximum flows in networks.

For a given set of agents $S \subseteq N$ and a piece of cake to be allocated X, define a graph $G(S,X)$ as follows. We keep track of a set of marks, which will be used to generate nodes in $G(S,X)$. First mark the left and right boundaries of all intervals that are contained in X. For each agent $i \in N$ and subinterval in U_i, mark the left and right boundaries of subintervals that are contained in $U_i \cap X$. When we have finished this process, each pair of consecutive markings will form an interval such that each agent either uniformly values the entire interval or values none of the interval. In $G(S,X)$, create a node for each interval I formed by consecutive markings, and add a node for each agent $i \in N$, a source node s, and a sink node t. For each interval I, add a directed edge from source s to I with capacity equal to the length of the interval. Each agent node is connected to t by an edge with capacity $\text{avg}(S,X)$. For each interval–agent pair (I,i), add a directed edge with infinite capacity from node I to the agent i if agent i desires interval I.

For example, suppose $U_1 = [0, 0.25] \cup [0.5, 1]$ and $U_2 = [0.1, 0.4]$. If $X = [0,1]$ then the interval markings will be $\{0, 0.1, 0.25, 0.4, 0.5, 1\}$. Agent 1 values $[0, 0.1]$, both agents value $[0.1, 0.25]$, agent 2 values $[0.25, 0.4]$, neither agent values $[0.4, 0.5]$ and agent 1 values $[0.5, 1]$. It holds that $\text{len}(D(\{1,2\},[0,1])) = 0.9$. Average values are 0.75, 0.3 and 0.45 for sets $\{1\}$, $\{2\}$ and $\{1,2\}$ respectively. See Fig. 1 for an illustration of the induced flow network.

Lemma 3.2. *Let $S \subseteq N$, and let X be a piece of cake. There is a flow of size* $\text{len}(D(S,X))$ *in* $G(S,X)$ *if and only if for all* $S' \subseteq S$, $\text{avg}(S',X) \geqslant \text{avg}(S,X)$.

We prove the lemma using an application of the classic Max-Flow Min-Cut Theorem (see, e.g., Cormen et al., 2001).

Proof. Assume that for all $S' \subseteq S$, $\text{avg}(S',X) \geqslant \text{avg}(S,X)$. By the Max-Flow Min-Cut Theorem, the minimum capacity removed from a graph in order to disconnect the source and sink is equal to the size of the maximum flow. The only edges with finite capacity in $G(S,X)$ are the ones that connect agent nodes to the sink, and the ones that connect the source to the interval nodes.

Construct a candidate minimum cut by disconnecting some set of agent nodes $T \subseteq S$ from the sink at cost $|T| \cdot \text{avg}(S,X)$ and then disconnecting all the (s,I) connections to interval nodes I desired by an agent $i \in S \setminus T$. This means that the total additional capacity we need to remove is $\text{len}(D(S \setminus T, X))$, the total length of intervals desired by at least one agent in $S \setminus T$. By assumption, this is at least $|S \setminus T| \cdot \text{avg}(S,X)$. As a result, this cut has capacity of at least

$$|T| \cdot \text{avg}(S,X) + |S \setminus T| \cdot \text{avg}(S,X) = |S| \cdot \text{avg}(S,X) = \text{len}\big(D(S,X)\big).$$

In the other direction, assume that there is a flow of size $\text{len}(D(S,X))$ in $G(S,X)$, and assume for contradiction that there exists $S' \subseteq S$ such that $\text{avg}(S',X) < \text{avg}(S,X)$. Construct a cut by disconnecting the (s,I) connections to interval nodes desired by an agent $i \in S'$, and disconnecting the agent nodes $S \setminus S'$ from the sink. The total capacity of the cut is

$$\begin{aligned}\left|S'\right| \cdot \text{avg}\big(S',X\big) + \left|S \setminus S'\right| \cdot \text{avg}(S,X) &< \left|S'\right| \cdot \text{avg}(S,X) + \left|S \setminus S'\right| \cdot \text{avg}(S,X)\\ &= |S| \cdot \text{avg}(S,X)\\ &= \text{len}\big(D(S,X)\big),\end{aligned}$$

and by the Max-Flow Min-Cut Theorem the maximum flow must be of size less than $\text{len}(D(S, X))$, in contradiction to our assumption. □

The following lemma establishes that a flow of size $\text{len}(D(S, X))$ in $G(S, X)$ induces an exact allocation.

Lemma 3.3. *Let $S \subseteq N$, and let X be a piece of cake. There exists an exact allocation with respect to S, X if and only if there exists a maximum flow of size* $\text{len}(D(S, X))$ *in* $G(S, X)$.

Proof. Suppose that we have a maximum flow of size $\text{len}(D(S, X))$; we show how to use this flow to generate an exact allocation with respect to S, X. For each edge between interval node I and agent $i \in N$ that receives positive flow of c in the max flow, allocate c of I to agent i. This allocation is feasible because the interval nodes represent disjoint subintervals of X and the flow to the interval's node is limited by the capacity of the edge between s and the interval node, which is the length of the interval. In addition, this allocation must give each agent exactly $\text{avg}(S, X)$ in desired intervals. To see this, note that all paths to the sink must pass through agent nodes, and the sum of capacities of the edges between the agents and the sink is $D(S, X)$. For a maximum flow to have size $D(S, X)$, these edges must saturated.

In the other direction, suppose that we have an exact allocation with respect to S, X. We can generate a feasible flow of size $\text{len}(D(S, X))$ by setting a flow of c on an edge (I, i) if agent i receives c of interval I in the exact allocation, and saturating all the edges (s, I) and (i, t) for intervals I and agents $i \in N$. □

By combining the "if" directions of Lemma 3.2 and Lemma 3.3 we see that the mechanism is indeed well-defined: if S has the smallest average then there exists an exact allocation with respect to S, X.[6] Moreover, we obtain a tractable mechanism for computing an exact allocation, by computing the maximum flow and deriving an exact allocation. A maximum flow can be computed in time that is polynomial in the number of nodes, that is, polynomial in our input size (see, e.g., Cormen et al., 2001).

3.4. Polynomial time

In order to show that Mechanism 1 can be implemented in polynomial time it remains to show that it is also possible to implement Step 2 of SubRoutine in polynomial time. Indeed, an efficient implementation of Step 2 would mean that SubRoutine can be implemented in polynomial time, and there are at most $n + 1$ calls to SubRoutine. So, the task is to find $S_{\min} \in \text{argmin}_{S' \subseteq S} \text{avg}(S, X)$ in polynomial time, given $S \subseteq N$ and a piece of cake X. This can be done using network flow arguments, which are an easy variation on those employed by Katta and Sethuraman (2006) (see Section 1 above). For completeness we quickly describe an implementation that is less efficient than Katta and Sethuraman (2006) but nevertheless polynomial time.

Given $S \subseteq N$, a piece of cake X, and $c > 0$, construct a graph $G'(S, X, c)$; this graph is identical to $G(S, X)$ as defined above, except that the capacity on the edges between the agents and the sink is c (instead of $\text{avg}(S, X)$). The proof of the following statement is identical to the proof of Lemma 3.2 (by replacing $\text{avg}(S, X)$ with c everywhere): there is a flow of size $c|S|$ in the network $G'(S, X, c)$ if and only if for all $S' \subseteq S$, $\text{avg}(S', X) \geqslant c$.

Assume that the boundaries of the agents' reference pieces of cake are represented by at most k bits, i.e. multiples of $1/2^k$. For ease of exposition (so that we deal with integers rather than rationals), scale the interval by $2^k n!$ so that the boundaries are an element of $\{0, n!, 2n!, \ldots, 2^k n!\}$.

There is a maximum c^* such that $G'(S, X, c^*)$ has a flow of size $c^*|S|$. This c^* is a member of a set of numbers that includes all the possible values of $\text{avg}(S', X)$ for $S' \subseteq S$. Because we have scaled the boundaries, the possible values for the average are contained in $\{0, 1, \ldots, 2^k n!\}$. Indeed, the numerator is an element of $\{0, n!, 2n!, \ldots, 2^k n!\}$, and the denominator is in $\{1, \ldots, n\}$. It holds that we can search for c^* by performing binary search over $\{0, 1, \ldots, 2^k n!\}$ in polynomial time. Binary search takes $O(k + n \log n)$ iterations, with each iteration taking time polynomial in k, n.[7]

Now, consider $c^* + 1$. This network does not have a network flow of size $(c^* + 1)|S|$. We can find a minimum cut in this network in polynomial time (see, e.g., Cormen et al., 2001). In this cut there is a subset $T \subseteq S$ such that the cut separates $S \setminus T$ from the sink, and the intervals desired by T from the source. This subset T must have $\text{avg}(T, X) < c^* + 1$, and since $\text{avg}(T, X)$ (after scaling) must be an integer, $\text{avg}(T, X) \leqslant c^*$.

We wish to claim that T is the $S_{\min}$ we are looking for. Indeed, note that since there is a flow of size $c^*|S|$ in $G'(S, X, c^*)$, it must hold that for all $S' \subseteq S$, $\text{avg}(S', X) \geqslant c^*$, which directly implies the claim.

3.5. Fairness, efficiency, truthfulness

Our main tool in proving that Mechanism 1 is truthful and EF is the following lemma.

[6] Note that the network in Fig. 1 does not satisfy the minimum average requirement and does not provide a corresponding exact allocation.

[7] Even though we multiply values by $2^k n!$, the number of bits needed is only $O(k + n \log n)$ and still polynomial in n, k.

Lemma 3.4. *Let $S_1, \dots, S_m$ be the ordered sequence of agent sets with the smallest average as chosen by Mechanism 1 and $X_1, \dots, X_m$ be the ordered sequence of pieces to be allocated in calls to* SubRoutine. *That is, $X_1 = [0,1]$, $X_2 = X_1 \setminus D(S_1, X_1), \dots, X_m = X_{m-1} \setminus D(S_{m-1}, X_{m-1})$. Then for all $i > j$, $\operatorname{avg}(S_i, X_i) \geqslant \operatorname{avg}(S_j, X_j)$, and agents that are members of later sets receive weakly more in desired lengths.*

Proof. Suppose not. Then at some point, $\operatorname{avg}(S_i, X_i) > \operatorname{avg}(S_{i+1}, X_{i+1})$. Now consider $S_i \cup S_{i+1}$. We will show that $\operatorname{avg}(S_i \cup S_{i+1}, X_i) < \operatorname{avg}(S_i, X_i)$, contradicting the choice of S_i as the subset of agents with the smallest avg at step i. Note that S_i and S_{i+1} are disjoint since agents are removed once they have been part of the subset with the smallest average. Thus,

$$\begin{aligned}
\operatorname{avg}(S_i \cup S_{i+1}, X_i) &= \frac{\operatorname{len}(D(S_i \cup S_{i+1}, X_i))}{|S_i| + |S_{i+1}|} \\
&= \frac{\operatorname{len}(D(S_i, X_i)) + \operatorname{len}(D(S_{i+1}, X_i \setminus D(S_i, X_i)))}{|S_i| + |S_{i+1}|} \\
&= \frac{|S_i| \cdot \operatorname{avg}(S_i, X_i) + |S_{i+1}| \cdot \operatorname{avg}(S_{i+1}, X_{i+1})}{|S_i| + |S_{i+1}|} \\
&< \frac{|S_i| \cdot \operatorname{avg}(S_i, X_i) + |S_{i+1}| \cdot \operatorname{avg}(S_i, X_i)}{|S_i| + |S_{i+1}|} \\
&= \operatorname{avg}(S_i, X_i),
\end{aligned}$$

where the second transition is true since S_i and S_{i+1} are disjoint, and the inequality follows from our assumption. □

3.5.1. Envy-freeness

Envy-freeness now follows immediately from Lemma 3.4. Indeed, consider an agent $i \in N$, and as before let S_j be the subset of agents with the smallest average in the j'th call to SubRoutine. Suppose $i \in S_j$. The agent does not envy other agents in S_j since these agents are given an exact allocation and all receive the same length in desired intervals. By Lemma 3.4, the agent does not envy agents in S_k for $k < j$ because the amount agents receive weakly increases with each call. The agent does not envy agents in S_k for $k > j$ because all intervals desired by the agent are allocated and removed from consideration when the agent receives its allocation.

3.5.2. Proportionality

Before we establish proportionality, we prove the following useful lemma, which shows that no intervals are wasted in the allocation produced by Mechanism 1.

Lemma 3.5. *Mechanism 1 assigns all subintervals desired by at least one agent to an agent who desires the interval.*

Proof. Every interval in $[0,1]$ is either allocated in one of the calls to SubRoutine, or it is left unallocated by the final call to SubRoutine. If an interval is allocated in one of the calls to SubRoutine, then it is part of an exact allocation, and in an exact allocation, each allocated agent receives only desired intervals. If an interval is left unallocated in the final call to SubRoutine, then it must be that no agent desired the interval. Indeed, all intervals desired by agents allocated prior to the final call have already been allocated, and the final call allocates all of $D(S', X')$ to agents in S', where S' and X' are the inputs to the final call of SubRoutine. □

Combining Lemma 3.5 with envy-freeness gives us proportionality. Indeed, suppose that some agent i receives less than $\operatorname{len}(D(\{i\}, [0,1])/|S|$ of its desired intervals. Because no desired intervals are thrown away, some other agent must receive a length of at least $\operatorname{len}(D(\{i\}, [0,1])/|S|$ of the desired intervals of agent i, in contradiction to envy-freeness.

3.5.3. Lorenz dominance

The allocation produced by Mechanism 1 is related to the egalitarian solution for supermodular transferable utility cooperative games described by Dutta and Ray (1989). In particular, the length in intervals received by an agent in Mechanism 1 is equal to the value assigned to an agent in the egalitarian solution of a corresponding convex cooperative game.

To translate our setting into the cooperative game setting, consider the following coalitional value function for each set of agents S:

$$v(S) = \operatorname{len}\bigl(D\bigl(S, [0,1]\bigr)\bigr).$$

This value function is clearly concave, so to translate to the convex setting, we consider the negation. Viewed in this light, we have a cost-sharing scenario, where the interval $[0,1]$ represents inputs to production which must be purchased. For instance, we can think of $[0,1]$ as a strip of land, and each agent requires irrigation canals on certain subintervals of this strip of land. Once irrigation canals are built on a given subinterval, they can be used jointly by all the agents.

For convex games, Dutta and Ray (1989) give an algorithm (which is very similar to Mechanism 1) to find the egalitarian solution, and they show that the egalitarian solution Lorenz dominates every solution in the core of the cooperative game. To understand the significance of this result in our setting, we first need to understand Lorenz dominance and the core.

Translated to our setting, Lorenz dominance is the following criterion. For a given allocation $A_1, \dots, A_n$, define a lengths vector $\ell(A_1, \dots, A_n)$ which sorts the lengths in intervals agents receive from smallest to largest. An allocation $A_1, \dots, A_n$ Lorenz dominates an allocation $A'_1, \dots, A'_n$ if $\sum_{i=1}^{k} \ell_i(A_1, \dots, A_n) \geqslant \sum_{i=1}^{k} \ell_i(A'_1, \dots, A'_n)$ for all $1 \leqslant k \leqslant n$ with strict inequality for some k. In other words, the partial sums of the lengths vector are weakly greater, with strict inequality for at least one partial sum. As an example, if we have two agents that split the interval $[0, 1]$ (they each receive lengths of 0.5), then this allocation Lorenz dominates an allocation that gives the entire interval to a single agent.

Translated to our setting, the core consists of all allocations where no subset of agents S is allocated more than $\text{len}(D(S, [0, 1]))$ in intervals. In particular, the core contains at least all allocations that do not allocate intervals to agents who do not desire them. Indeed, if agents only receive desired intervals, there is no way that the total length of intervals received by agents in a subset S can exceed $\text{len}(D(S, [0, 1])$.

Given the correspondence between the egalitarian solution and the allocation produced by Mechanism 1, the allocation Lorenz dominates any other allocation that only allocates intervals to agents who desire them and has a distinct lengths vector.[8] It was *a priori* unclear whether there would exist an allocation in which the desired lengths received by agents corresponds to the values in the egalitarian solution because our setting is not a transferable utility setting. This is what is established in Section 3.3. Moreover, our definition of Lorenz dominance looks at the lengths vector and not agent utilities. Indeed, it is easy to show that a similar criterion is not satisfied if we characterize allocations by the vectors of agent's utilities.

3.5.4. *Pareto-efficiency*

Each agent's utility is a function of the length in desired intervals it receives. As a result, Pareto-efficiency follows directly from the discussion on Lorenz dominance.

3.5.5. *Truthfulness*

We establish truthfulness by examining the possible ways in which an agent can affect the progression of Mechanism 1. In particular, we examine whether an agent $i \in S$ would want to deviate from truthfulness in each call to $\textsc{SubRoutine}(S, X)$. Let $S_{\min}$ denote the set of agents with the smallest average if agent i reports truthfully. For a given call to $\textsc{SubRoutine}$, agent i can report a valuation which changes the exact allocation made to $S_{\min}$ or which changes the choice of $S_{\min}$ (or both).

We first consider deviations which try to modify the exact allocation for $S_{\min}, X$. The exact allocation for $S_{\min}$ only depends on the reports of agents in $S_{\min}$, so we assume $i \in S_{\min}$. If agent i reports truthfully, it receives exactly $\text{len}(D(S_{\min}, X))/|S_{\min}|$ in desired lengths. Agent i can deviate, causing the set of desired intervals to change to $D'(S_{\min}, X)$. If $\text{len}(D'(S_{\min}, X)) \leqslant \text{len}(D(S_{\min}, X))$, then this is not profitable. Suppose that $\text{len}(D'(S_{\min}, X)) > \text{len}(D(S_{\min}, X))$. Now each agent receives $\text{len}(D'(S_{\min}, X))/|S_{\min}|$ of intervals. The intervals received by agents other than i are in $D(S_{\min}, X)$ since the other agents have not changed their reports. Subtracting off these intervals, the maximum length in desired intervals agent i can receive from this deviation is $\text{len}(D(S_{\min}, X)) - (|S| - 1) \cdot \text{len}(D'(S_{\min}, X))/|S_{\min}| \leqslant \text{len}(D(S_{\min}, X))/|S_{\min}|$, and this deviation is not profitable.

We next consider deviations which attempt to change the choice of subset with the smallest average. There are two cases to consider. Let avg' denote the new averages induced by a deviation of agent i:

1. $i \notin S_{\min}$. For any set S', if $i \notin S'$, agent i cannot change $D(S', X)$ with its reports. As a result, agent i cannot make the mechanism choose some other $S', i \notin S'$ since the agent cannot change $\text{avg}(S_{\min}, X)$ or $\text{avg}(S', X)$. Therefore, the agent's only deviation is to try and make the mechanism choose a set $S', i \in S'$. However, in order to do so, the agent must make $\text{avg}'(S', X) \leqslant \text{avg}(S_{\min}, X)$ and will receive $\text{avg}'(S', X)$ in desired lengths. By Lemma 3.4, the agent was receiving at least $\text{avg}(S_{\min}, X)$ under truthful reports since the agent was chosen in a later round, so this is not profitable.
2. $i \in S_{\min}$. Agent i receives $\text{avg}(S_{\min}, X)$ if it reports truthfully. Suppose agent i deviates and forces selection of another set S' as the set with the smallest average. If $i \in S'$, then to be profitable, $\text{avg}'(S', X) \geqslant \text{avg}(S_{\min}, X)$. If $i \notin S'$, then agent i could not have affected $\text{avg}(S', X)$ and S' was not chosen when i was truthful, so $\text{avg}'(S', X) \geqslant \text{avg}(S_{\min}, X)$. Either way, to be profitable, it must be that $\text{avg}'(S', X) \geqslant \text{avg}(S_{\min}, X)$. Now consider some agent $j \in S_{\min}$. When agent i reported truthfully, agent j was receiving $\text{avg}(S_{\min}, X)$ of $D(S_{\min}, X)$. If $j \in S'$, then agent j receives exactly $\text{avg}'(S', X)$ in intervals. If $j \notin S'$, then by Lemma 3.4, agent j receives at least $\text{avg}'(S', X)$ in intervals. In either case, agent j receives at least $\text{avg}'(S', X)$ in intervals.

[8] There may be multiple allocations with the same lengths vector. As a result, the allocation we produce does not Lorenz dominate every other allocation; however, it does Lorenz dominate every other allocation that generates a distinct lengths vector.

Using arguments similar to the case above, we can now show that no deviation is profitable. While agent i deviates, the other agents in $S_{\min}$ do not, so the intervals they receive are a part of $D(S_{\min}, X)$. This leaves

$$\begin{aligned}\operatorname{len}\bigl(D(S_{\min}, X)\bigr) - \bigl(|S_{\min}| - 1\bigr) \cdot \operatorname{avg}'\bigl(S', X\bigr) &\leqslant \operatorname{len}\bigl(D(S_{\min}, X)\bigr) - \bigl(|S_{\min}| - 1\bigr) \cdot \operatorname{avg}(S_{\min}, X) \\ &= \operatorname{avg}(S_{\min}, X)\end{aligned}$$

in desired intervals for agent i, so the deviation is not profitable.

This completes the proof of truthfulness. Putting everything together gives us Theorem 3.1.

4. Randomized mechanisms

In the previous section we saw that designing deterministic truthful and fair mechanisms is not an easy task, even if the valuation functions of the agents are rather restricted. In this section we demonstrate that by allowing randomness we can obtain significantly more general results.

A *randomized cake cutting mechanism* outputs a random allocation given the reported valuation functions of the agents. There are very few previous papers regarding randomized mechanisms for cake cutting. A rare example is the paper by Edmonds and Pruhs (2006b), where they give a randomized mechanism that achieves approximate proportionality with high probability. We are looking for a more stringent notion of fairness. We say that a randomized mechanism is *universally proportional* (resp., *universally EF*) if it always returns an allocation that is proportional (resp., EF).

One could also ask for *universal truthfulness*, that is, require that an agent may never benefit from lying, regardless of the randomness of the mechanism. A universally truthful mechanism is simply a probability distribution over deterministic truthful mechanisms. However, asking for both universal fairness and universal truthfulness would not allow us to enjoy the additional flexibility that randomization provides. Therefore, we slightly relax our truthfulness requirement. Informally, we say that a randomized mechanism is *truthful in expectation* if, for all possible valuation functions of the other agents, the expected value an agent receives for its allocation cannot be increased by lying, where the expectation is taken over the randomness of the mechanism.

We remark that while truthfulness in expectation seems natural, consistent as it is with expected utility maximization, fairness (i.e., proportionality and envy-freeness) is something that we would like to hold *ex-post*; fairness is a property of the specific allocation that is being made, and continues to be relevant after the mechanism has terminated. Interestingly enough, if we were to turn this around, then achieving universal truthfulness and envy-freeness/proportionality in expectation is trivial: simply allocate the entire cake to a uniformly random agent!

There is a simple class of randomized mechanisms which are truthful in expectation. Specifically, define a *partition mechanism* as follows. On input $V_1, \ldots, V_n$ a partition mechanism chooses a partition $\{X_1, \ldots, X_n\}$ of $[0, 1]$. The mechanism then uniformly samples a random permutation π over the n agents, and assigns agent i the piece $X_{\pi(i)}$. If an agent reports truthfully, then in expectation, it receives value $\sum_{j=1}^{n} \frac{1}{n} V_i(X_j) = 1/n$. If the agent reports some other valuation V_i', then the partition the mechanism chooses may change to $\{X_1', \ldots, X_n'\}$, but the agent still receives value $\sum_{j=1}^{n} \frac{1}{n} V_i(X_j') = 1/n$.

However, depending on the partitions chosen, a partition mechanism may not be universally proportional and EF. In order to obtain a mechanism that is universally proportional and EF, we require that the partitions selected are *perfect partitions*. A partition is perfect for reports $V_1, \ldots, V_n$ if $V_i(X_j) = 1/n$ for every i and j. As the agents have value exactly $1/n$ for every element of the partition, a partition which always selects perfect partitions is clearly universally proportional and EF.[9]

Though partition mechanisms which select perfect partitions are truthful in expectation and universally proportional and EF, there still remains the obstacle of actually finding a perfect partition given the valuation functions of the agents. Does such a partition exist, and can it be computed? More than sixty years ago, Neyman (1946) proved that if the valuation functions of the agents are defined by the integral of a continuous probability measure then there *exists* a perfect partition. Unfortunately, Neyman's proof is nonconstructive, and to this day there is no known constructive method under general assumptions on the valuation functions. This is not surprising since a perfect partition induces an EF allocation, and finding an EF allocation in a bounded number of steps for more than four agents is an open problem.

However, if we assume more structure on agent preferences, then it is possible to explicitly construct perfect partitions. As an example, consider the class of piecewise constant valuation functions. Each agent makes a mark at the left and right boundaries of each of the intervals it is interested in, and we add two marks at 0 and 1. The value density function of each agent is constant over each subinterval between two consecutive marks, as this subinterval is either contained in one of the subintervals agent i is interested in or completely disjoint from any of them. Hence, the allocation that assigns to each agent exactly $1/n$ of each of the subintervals between two consecutive marks is a perfect partition. A similar procedure can be applied to piecewise linear valuation functions by observing that if agent densities are linear on a subinterval, we can partition the subinterval into n elements that have the same value to each agent. Specifically, we can divide the subinterval into $2n$ pieces of equal length and label the pieces $1, \ldots, 2n$ from left to right. To construct element i of the partition, we

[9] Mossel and Tamuz (2010) also observe that partition mechanisms which select perfect partitions are truthful in expectation and universally proportional.

match piece i with piece $2n - i + 1$. An agent has the same value for every element of the partition due to the assumption of linear densities on the subinterval.

5. Discussion

We have made progress on the new question of fair and (dominant strategy) truthful cake cutting. On the way we have made several conceptual contributions, including our consideration of restricted valuation functions in the context of cake cutting, and (independently of Mossel and Tamuz, 2010) the introduction of randomized notions of fairness and a focus on truthfulness which is familiar from mechanism design but has not been the focus of existing cake cutting literature.

The free disposal assumption For our deterministic result we rely heavily on the ability to throw undesired pieces of the cake away. The reason this is important is that it prevents agents from being able to gain by reporting smaller intervals, in hopes of gaining a larger share of the overlap while obtaining desired intervals for free. The following example illustrates why removing the free disposal assumption can be problematic.

Since we can no longer discard undesired intervals, we must specify how to allocate the undesired intervals. Suppose we decide to allocate undesired intervals evenly between agents 1 and 2, giving the left half to agent 1. While this seems benign, our mechanism is no longer truthful.

Suppose there are two agents. Consider the allocation made when both agents report that they desire the interval $[0, 0.2]$. Mechanism 1 calls for any allocation that splits $[0, 0.2]$ equally between the agents.[10] Assume a simple implementation that gives $[0, 0.1]$ to agent 1 and $[0.1, 0.2]$ to agent 2. Our rule for allocating undesired intervals gives $[0.2, 0.6]$ to agent 1 and $[0.6, 1]$ to agent 2.

Now consider the case where agent 1 values $[0, 0.6]$ and agent 2 values $[0, 0.2]$. If the agents report truthfully, Mechanism 1 gives $[0, 0.2]$ to agent 2 and $[0.2, 0.6]$ to agent 1. Agent 1 receives $[0.6, 0.8]$ and agent 2 receives $[0.8, 1]$ due to our rule for allocating undesired intervals. Agent 1 receives value $2/3$. On the other hand, if agent 1 reports $[0, 0.2]$ (assuming agent 2 remains truthful), it receives $[0, 0.1] \cup [0.2, 0.6]$ which gives value $5/6$. Notice that the problem here persists even if we allocate $[0.2, 0.6]$ to agent 2 when both agents report $[0, 0.2]$. In this case, agent 2 can deviate when it has true value $[0, 0.6]$ and agent 1 reports $[0, 0.2]$.

It is unclear whether we can dispose of the free disposal assumption. Indeed, it may be that there is an impossibility result when we force all intervals to be allocated. It would be nice to be able to apply the same insights of Mechanism 1, though as the example shows, we now have to be careful about the exact mechanics of *how* intervals are allocated rather than simply the fraction of each interval each agent receives. This adds significant complexity to the problem which we are able to avoid in Mechanism 1 when we assume free disposal.

The communication model In general, communication issues in cake cutting are rather subtle since the input (that is, the valuations functions) is continuous. To deal with this issue, Robertson and Webb (1998) introduced a *concrete model of complexity* for cake cutting mechanisms. Under their model, a mechanism is restricted to making two types of queries, an evaluation query (whereby the mechanism learns the value of an agent with respect to a given interval) and a cut query (whereby the mechanism obtains a piece worth a given value to an agent). Through the queries the mechanism must obtain sufficient information to output a fair allocation. Existing work on computational aspects of cake cutting uses this framework (see, e.g., Busch et al., 2005; Edmonds and Pruhs, 2006a; Woeginger and Sgall, 2007; Procaccia, 2009), but without considering truthfulness.

In this paper we have implicitly assumed that the agents report their full valuation, an assumption that facilitates our focus on strategic issues. However, the classes of valuation functions that we have considered have compact representations. Therefore, agents can in fact efficiently report their full valuation functions. Indeed, in the case of piecewise uniform valuation functions it is sufficient to report the left and right boundaries of each of the desired intervals. In the case of piecewise linear valuation functions, agents also report a partition of the cake into intervals, along with the two parameters of the linear function on each of the intervals. This allows us to design mechanisms that are polynomial time in the size of the input.

That said, it is unclear whether our mechanisms can be efficiently implemented via evaluation and cut queries. For example, it seems difficult to isolate the desired intervals of an agent using a finite number of such queries. This does not preclude, however, the existence of results that are analogous to ours via mechanisms that only rely on evaluation and cut queries.

Future work The most prominent technical challenge is to generalize Mechanism 1. The first step would be a deterministic, truthful, proportional, and EF mechanism under the assumption that the agents have piecewise constant valuations, and the second step would be achieving the same result with respect to piecewise linear valuations. Unfortunately we cannot rule out the scenario where no such mechanisms exist; future work would have to resolve this issue.

[10] In Mechanism 1, any exact allocation suffices in SubRoutine, and it is not necessary to worry about the specific way an exact allocation is achieved.

Acknowledgments

We thank Noga Alon for his wise advice regarding the tractable implementation of Step 2 of SubRoutine. We also thank Anna Bogomolnaia and Hervé Moulin for pointing our attention to the connection between our problem and the random assignment problem. We also thank anonymous referees for insightful critiques and suggestions for improvements. This work is supported in part by NSF grant CCF-1215883. John Lai is supported by an NDSEG fellowship.

Appendix A. Equivalence of swapping procedure and Mechanism 1

We prove that the swapping procedure from Section 3.2.2 is equivalent to Mechanism 1 for two agents. In the swapping procedure from Section 3.2.2, there are two cases. The first is where agent 2 gives away all of its share of W_{12}, and the second is where agents 1 and 2 divide W_{12}. We can give precise conditions for when each of the two cases occurs. After the first set of swaps, agent 1 has all of W_1, and agent 2 has $\text{len}(W_2 \cup W_1)/2$ of W_2 and still desires the $(\text{len}(W_2) - \text{len}(W_1))/2$ of W_2 still held by agent 1. If $\text{len}(W_{12})/2 \geqslant (\text{len}(W_2) - \text{len}(W_1))/2$ then agent 2 can get all of W_2 without exhausting all of its share of W_{12}, and the agents equally split $W_1 \cup W_2 \cup W_{12}$. Otherwise, agent 1 receives all of W_1, W_{12} and agent 2 receives only W_2.

The crux of the swapping procedure is the allocation of W_{12}. Each agent always receives W_i, and W_{12} is allocated to try and equalize the total lengths of intervals obtained by each agent. However, this may not always be possible because agent 1 may desire less than $\text{len}(W_1 \cup W_2 \cup W_{12})/2$, and in this case, agent 1 receives all of W_{12}. To summarize, there are two cases in the swapping procedure:

1. $\text{len}(W_1 \cup W_{12}) \geqslant \text{len}(W_1 \cup W_2 \cup W_{12})/2$. Agent 1 receives W_1, agent 2 receives W_2 and the agents split W_{12} so that their allocated lengths are equal.
2. $\text{len}(W_1 \cup W_{12}) < \text{len}(W_1 \cup W_2 \cup W_{12})/2$. Agent 1 receives $W_1 \cup W_{12}$ and agent 2 receives W_2.

To see that this swapping procedure is exactly equivalent to Mechanism 1 for two agents, assume again without loss of generality that $\text{len}(W_1) \leqslant \text{len}(W_2)$. In Mechanism 1, either $\{1\}$ or $\{1, 2\}$ is the subset with the smallest average. $\{1, 2\}$ is the chosen subset if $\text{len}(W_1) + \text{len}(W_{12}) \geqslant \text{len}(W_1 \cup W_2 \cup W_{12})/2$. If $\{1, 2\}$ is chosen, then agents 1 and 2 split $W_1 \cup W_2 \cup W_{12}$ in an exact allocation. If $\{1\}$ is chosen, then agent 1 receives all of W_1, W_{12}, and agent 2 receives all of W_2.

References

Bogomolnaia, A., Moulin, H., 2001. A new solution to the random assignment problem. J. Econ. Theory 100, 295–328.
Bogomolnaia, A., Moulin, H., 2004. Random matching under dichotomous preferences. Econometrica 72, 257–279.
Brams, S.J., Taylor, A.D., 1996. Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press.
Brams, S.J., Jones, M.A., Klamler, C., 2006. Better ways to cut a cake. Not. Am. Math. Soc. 53 (11), 1314–1321.
Brams, S.J., Jones, M.A., Klamler, C., 2008. Proportional pie-cutting. Int. J. Game Theory 36 (3–4), 353–367.
Busch, C., Krishnamoorthy, M.S., Magdon-Ismail, M., 2005. Hardness results for cake cutting. Bull. Eur. Assoc. Theor. Comput. Sci. 86, 85–106.
Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2001. Introduction to Algorithms, 2nd edition. MIT Press.
Dutta, B., Ray, D., 1989. A concept of egalitarianism under participation constraints. Econometrica 57, 615–635.
Edmonds, J., Pruhs, K., 2006a. Cake cutting really is not a piece of cake. In: Proc. of 17th Symposium on Discrete Algorithms, pp. 271–278.
Edmonds, J., Pruhs, K., 2006b. Balanced allocations of cake. In: Proc. of 47th Foundations of Computer Science, pp. 623–634.
Katta, A., Sethuraman, J., 2006. A solution to the random assignment problem on the full preference domain. J. Econ. Theory 131, 231–250.
Megiddo, N., 1979. A good algorithm for lexicographically optimal flows in multi-terminal networks. Bull. Am. Math. Soc. 83, 407–409.
Mossel, E., Tamuz, O., 2010. Truthful fair division. In: Proc. of 3rd Symposium on Algorithmic Game Theory, pp. 288–299.
Neyman, J., 1946. Un théorèm d'existence. C. R. Acad. Sci. Paris 222, 843–845.
Procaccia, A.D., 2009. Thou shalt covet thy neighbor's cake. In: Proc. of 21st Int. Joint Conference on Artificial Intelligence, pp. 239–244.
Robertson, J.M., Webb, W.A., 1998. Cake Cutting Algorithms: Be Fair if You Can. A.K. Peters.
Steinhaus, H., 1948. The problem of fair division. Econometrica 16, 101–104.
Thomson, W., 2007. Children crying at birthday parties. Why? J. Econ. Theory 31, 501–521.
Woeginger, G.J., Sgall, J., 2007. On the complexity of cake cutting. Discrete Optim. 4, 213–220.