

# Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations

Haris Aziz<sup>1</sup> and Chun Ye<sup>2</sup>

<sup>1</sup> NICTA and UNSW, Sydney Australia NSW 2033  
haris.aziz@nicta.com.au

<sup>2</sup> Columbia University, New York, NY 10027-6902, USA  
cy2214@columbia.edu

**Abstract.** Cake cutting is one of the most fundamental settings in fair division and mechanism design without money. In this paper, we consider different levels of three fundamental goals in cake cutting: fairness, Pareto optimality, and strategyproofness. In particular, we present robust versions of envy-freeness and proportionality that are not only stronger than their standard counter-parts but also have less information requirements. We then focus on cake cutting with piecewise constant valuations and present three desirable algorithms: *CCEA* (*Controlled Cake Eating Algorithm*), *MEA* (*Market Equilibrium Algorithm*) and *CSD* (*Constrained Serial Dictatorship*). *CCEA* is polynomial-time, robust envy-free, and non-wasteful. It relies on parametric network flows and recent generalizations of the probabilistic serial algorithm. For the subdomain of piecewise uniform valuations, we show that it is also group-strategyproof. Then, we show that there exists an algorithm (*MEA*) that is polynomial-time, envy-free, proportional, and Pareto optimal. *MEA* is based on computing a market-based equilibrium via a convex program and relies on the results of Reijnierse and Potters [24] and Devanur et al. [15]. Moreover, we show that *MEA* and *CCEA* are equivalent to mechanism 1 of Chen et. al. [12] for piecewise uniform valuations. We then present an algorithm *CSD* and a way to implement it via randomization that satisfies strategyproofness in expectation, robust proportionality, and unanimity for piecewise constant valuations. For the case of two agents, it is robust envy-free, robust proportional, strategyproof, and polynomial-time. Many of our results extend to more general settings in cake cutting that allow for variable claims and initial endowments. We also show a few impossibility results to complement our algorithms. The impossibilities show that the properties satisfied by *CCEA* and *MEA* are maximal subsets of properties that can be satisfied by any algorithm for piecewise constant valuation profiles.

## 1 Introduction

Cake cutting is one the most fundamental topics in fair division (see e.g., [23, 8, 25]). It concerns the setting in which a cake is represented by an interval  $[0, 1]$  and each of the  $n$  agents has a value function over the cake. The main aim is to divide the cake fairly. The framework is general enough to encapsulate the important problem of allocating

a heterogeneous divisible good among multiple agents with different preferences. The cake cutting problem applies to many settings including the division of rent among housemates, disputed land between land-owners, and work among co-workers. It is especially useful in scheduling the use of a valuable divisible resource such as server time.

Within the cake cutting literature, the most important criteria of a fair allocation are *envy-freeness* and *proportionality*. In an envy-free allocation, each agent considers his allocation at least as good as any other agent's allocation. An envy-free allocation is guaranteed to exist (see e.g., [28, 29]). In a proportional allocation, each agent gets at least  $1/n$  of the value he assigns to the cake. A desirable aspect of envy-freeness is that it implies proportionality.<sup>3</sup>

Computation of a fair allocation of cake is one of the fundamental problems in algorithmic economics. Brams and Taylor [7] designed an envy-free cake cutting algorithm for an arbitrary number of players. Although their algorithm is guaranteed to eventually terminate, its running time is unbounded. Moreover, the algorithm can divide the cake into infinitely small segments. Since the result of Brams and Taylor [7], researchers have examined restricted value density functions and proposed envy-free algorithms for them. In order to ascertain the running time of a cake cutting algorithm, it is important to know the computational model and input to the problem. In most of the literature (see e.g., [25]), it is assumed that the value an agent ascribes to any segment of the cake can be queried or evaluated via an oracle. While the classical literature uses this query model, computer scientists recently looked at the problem from the point of view of full report, as is common in mechanism design. Throughout the paper we focus on *piecewise constant value density functions* and *piecewise uniform value density functions*. Piecewise constant value density functions are one of the most fundamental class of value functions. Piecewise uniform valuations are a restricted class of piecewise constant valuations. Chen et al. [11, 12] presented a discrete, strategyproof, polynomial-time, envy-free and Pareto optimal algorithm for piecewise uniform valuations. They stated that generalizing their results for piecewise constant valuations is an open problem. They also presented an envy-free and proportional algorithm that satisfies strategyproofness by resorting to randomization.

In this paper, we consider three of the most enduring goals in mechanism design and fair division: fairness, Pareto optimality and strategyproofness. Since many fair division algorithms need to be deployed on a large scale, we will also aim for algorithms that are *computationally efficient*. Our main research question in this paper is as follows: *among the different levels of fairness, Pareto optimality, strategyproofness, and efficient computability, what are the maximal set of properties that can be satisfied simultaneously for piecewise constant and piecewise uniform valuations?* Our main contribution is a detailed study of this question including the formulation of a number of desirable cake cutting algorithms satisfying many of the properties.

In the case where Pareto optimality cannot be satisfied, we also consider a weaker notion of efficient called *non-wastefulness*. Non-wastefulness dictates that every portion

---

<sup>3</sup> This statement holds with the additional assumption of *full allocation*: that every portion of the cake that is desired by at least one agent is allocated to some agent. Otherwise, the empty allocation satisfies envy-freeness, but not proportionality.

of the cake that is desired by at least one agent is allocated to some agent who desires it.

For fairness, we not only consider the standard notions envy-freeness and proportionality but we also propose the concept of *robust fairness* — in particular *robust envy-freeness* and *robust proportionality*. The main idea of an allocation being robust envy-free is that even if an agent readjusts his value density function, as long as the ordinal information of the function is unchanged, then the allocation remains envy-free. The main advantages of robust envy-freeness are less information requirements and envy-freeness under uncertainty.<sup>4</sup>

For strategic properties, we consider three notions of truthfulness namely strategyproofness, weak group-strategyproofness and group-strategyproofness. In most of the cake-cutting literature, an algorithm is considered ‘strategyproof’ if truth-telling is a maximin strategy [6] and it need not be dominant strategy incentive compatible. When we refer to strategyproofness, we will mean the reporting the truthful valuations is a dominant strategy. This stronger notion of strategyproofness has largely been ignored in cake-cutting barring a few recent exceptions [11, 12, 21, 20].<sup>5</sup>

We present *CCEA (Controlled Cake Eating Algorithm)* for piecewise constant valuations and show that it is polynomial-time and robust envy-free and robust proportional. CCEA depends on a reduction to the generalizations [1, 18] of the *PS (probabilistic serial)* algorithm introduced by Bogomolnaia and Moulin [3] in the context of random assignments.<sup>6</sup> The algorithm relies on solving the parametric network flows (see e.g., [16]). We show that the algorithm can handle variable claims and private endowments for piecewise constant valuations and also satisfies group-strategyproofness under piecewise uniform valuations.

If we insist on Pareto optimality, then we show that there exists an algorithm which we refer to as the *MEA (Market Equilibrium Algorithm)* that is discrete, polynomial-time Pareto optimal, envy-free, and proportional for piecewise constant valuations. The algorithm relies on the Walras equilibrium formulation of Reijnierse and Potters [24] for finding an  $\alpha$ -envy-free for general cake cutting valuations and the result of Devanur et al. [15] that market equilibrium for Fischer markets with linear utilities can be computed in polynomial time. Both CCEA and MEA not only coincide on piecewise uniform valuations but are also group-strategyproof.

Although CCEA and MEA are desirable algorithms, they are not strategyproof for piecewise constant valuations. We present another algorithm called *CSD (Constrained Serial Dictatorship)* which is strategyproof in expectation, robust proportional, and satisfies unanimity. For the important case of two agents<sup>7</sup>, it is polynomial-time, and robust

<sup>4</sup> Although full information is a standard assumption in cake cutting, it can be argued that it is unrealistic that agents have exact utility value for each segment of the cake. Even if they do report exact utility values, they may be uncertain about these reports. Robust fairness bypasses these issues.

<sup>5</sup> Procaccia [23] writes that the “design of strategyproof cake cutting algorithms is largely an open problem.”

<sup>6</sup> The CC algorithm of Athanassoglou and Sethuraman [1] is a generalization of the EPS algorithm [18] which in turn is a generalization of PS algorithm of Bogomolnaia and Moulin [3].

<sup>7</sup> Many fair division problems involve disputes between two parties.

envy-free. To the best of our knowledge, it is the first cake cutting algorithm for piecewise constant valuations that satisfies strategyproofness, proportionality, and unanimity at the same time. CSD requires some randomization to achieve strategyproof in expectation. However, CSD is discrete in the sense that it gives the same utility guarantee (with respect to the reported valuation functions) over all realizations of the random allocation. Although CSD uses some essential ideas of the well-known *serial dictatorship* rule for discrete allocation, it is significantly more involved.

Our main technical results are as follows.

**Theorem 1.** *For cake cutting with piecewise constant valuations, there exists an algorithm (CCEA) that is discrete, polynomial-time, robust envy-free, and non-wasteful.*

**Theorem 2.** *For cake cutting with piecewise constant valuations, there exists an algorithm (MEA) that is discrete, polynomial-time, Pareto optimal, and envy-free.*

**Theorem 3.** *For cake cutting with piecewise uniform valuations, there exists algorithms (CCEA and MEA) that are discrete, group strategyproof, polynomial-time, robust envy-free and Pareto optimal.*

**Theorem 4.** *For cake cutting with piecewise constant valuations, there exists a randomized implementation of an algorithm (CSD) that is (ex post) robust proportional, (ex post) symmetric, and (ex post) unanimous and strategyproof in expectation. For two agents, it is polynomial-time, robust proportional and robust envy-free.*

Our positive results are complemented by the following impossibility results. These impossibility results suggest the properties satisfied by CCEA and MEA are maximal subsets of properties that can be satisfied by any algorithm for piecewise constant valuation profiles.

**Theorem 5.** *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is both Pareto optimal and robust proportional.*

**Theorem 6.** *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, Pareto optimal, and proportional.*

**Theorem 7.** *For piecewise constant valuation profiles with at least two agents, there exists no algorithm that is strategyproof, robust proportional, and non-wasteful.*

As a consequence of CCEA and MEA, we generalize the positive results regarding piecewise uniform valuations in [11, 12] and piecewise constant valuations in [13] in a number of ways such as handling richer cake cutting settings, handling more general valuations functions, achieving a stronger envy-free concept, or a stronger strategyproofness notion. Moreover, we prove that CCEA and MEA — two different algorithms — are equivalent in the domain of piecewise constant valuations. Furthermore, we show that both CCEA and MEA are generalizations of the main mechanism in [12, 11]. We also show which combinations of properties are impossible to satisfy simultaneously. Some of our main results are summarized in Table 1.

After presenting our main results, we enrich the cake cutting domain in two ways. We allow agents to have initial endowments of the cake. Moreover we consider the more general setting in which agents may have different claims or entitlements to the cake. We show that many of our results carry over to these more general settings.

Algorithms	Restriction	DISC	R-EF	EF	R-PROP	PROP	GSP	W-GSP	SP	PO	UNAN	POLYT
CCEA	-	+	+	+	+	+	-	-	-	-	+	+
CCEA	pw uniform	+	+	+	+	+	+	+	+	+	+	+
MEA	-	+	-	+	-	+	-	-	-	+	+	+
MEA	pw uniform	+	+	+	+	+	+	+	+	+	+	+
CMSD	-	-	-	-	+	+	-	-	+	-	+	-
CMSD	pw uniform	-	-	-	+	+	-	-	+	+	+	-
CMSD	2 agents	-	+	+	+	+	-	-	+	-	+	+

**Table 1.** Properties satisfied by the cake cutting algorithms for pw (piecewise) constant valuations: DISC (discrete), R-EF (robust envy-freeness), EF (envy-freeness), R-PROP (robust proportionality), PROP (proportionality), GSP (group strategyproof), W-GSP (weak group strategyproof), SP (strategyproof), UNAN (unanimity), PO (Pareto optimal) and POLYT (polynomial-time).

## 2 Preliminaries

### 2.1 Cake cutting setting

We consider a cake which is represented by the interval  $[0, 1]$ . A *piece of cake* is a finite union of disjoint subintervals of  $[0, 1]$ . The length of an interval  $I = [x, y]$  is  $len(I) = y - x$ . As usual, the set of agents is  $N = \{1, \dots, n\}$ . Each agent has a piecewise continuous *value density function*  $v_i : [0, 1] \rightarrow [0, \infty]$ . The value of a piece of cake  $X$  to agent  $i$  is  $V_i(X) = \int_X v_i(x) dx = \sum_{I \in X} \int_I v_i(x) dx$ . As generally assumed, valuations are non-atomic ( $V_i([x, x]) = 0$ ) and additive:  $V_i(X \cup Y) = V_i(X) + V_i(Y)$  where  $X$  and  $Y$  are disjoint. The basic cake cutting setting can be represented by the set of agents and their valuations functions, which we will denote as a *profile of valuations*. In this paper we will assume that each agent's valuation function is private information for the agent that is not known to the algorithm designer. Each agent reports his valuation function to the designer and the designer then decides how to make the allocations based on the reported valuations.

Later on we will also consider two important extensions of cake cutting: claims and private endowments. We will assume that agents have the following *claims* on the cake respectively:  $c_1, \dots, c_n$ . In the original cake cutting problem agents have equal claims. Each agent  $i \in N$  has a *private endowment*  $\omega(i)$  which is a segment of the cake privately owned by  $i$ . The cake is assembled by joining the pieces  $\omega(1), \dots, \omega(n)$ . Therefore the cake cutting setting in its full generality can be represented as a quadruple  $(N, v, \omega, c)$ .

An allocation is a partitioning of the cake into  $n$  pieces of cake  $X_1, \dots, X_n$  such that the pieces are disjoint (aside from the interval boundaries) and  $X_i$  is allocated to agent  $i \in N$ . A cake cutting algorithm takes as input  $(N, v, \omega, c)$  and returns an allocation.

### 2.2 Preference functions

In this paper we will only consider *piecewise uniform* and *piecewise constant* valuations functions. A function is *piecewise uniform* if the cake can be partitioned into a finite number of intervals such that for some constant  $k_i$ , either  $v_i(x) = k_i$  or  $v_i(x) = 0$

over each interval. A function is *piecewise constant* if the cake can be partitioned into a finite number of intervals such that  $v_i$  is constant over each interval. In order to report his valuation function to the algorithm designer, each agent will specify a set of points  $\{d_1, \dots, d_m\}$  that represents the consecutive points of discontinuity of the agent's valuation function as well as the constant value of the valuation function between every pair of consecutive  $d_j$ 's.

For a function  $v_i$ , we will refer by  $\hat{V}_i = \{v'_i : v_i(x) \geq v_i(y) > 0 \iff v'_i(x) \geq v'_i(y) > 0 \ \forall x, y \in [0, 1]\}$  as the set of density functions *ordinally equivalent* to  $v_i$ . Note that if an algorithm takes as input  $v_i$  and returns the same output for any  $v'_i \in \hat{V}_i$ , then it is oblivious to the exact cardinal information of  $v_i$ .

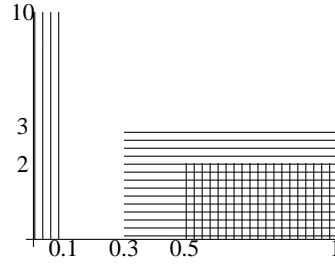
### 2.3 Properties of allocations

An *allocation* is a partition of the interval  $[0, 1]$  into a set  $\{X_1, \dots, X_n, W\}$ , where  $n$  is the number of agents and  $X_i$  is a piece of cake that is allocated to agent  $i$ . And  $W$  is the piece of the cake that is not allocated. All of the fairness and efficiency notations that we will discuss next are with respect to the reported valuation functions. In an *envy-free allocation*, we have  $V_i(X_i) \geq V_i(X_j)$  for each pair of agent  $i, j \in N$ . An allocation is *individually rational* if  $V_i(X_i) \geq V_i(\omega(i))$ . In a *proportional allocation*, each agent gets at least  $1/n$  of the value he has for the entire cake. An allocation satisfies *symmetry or equal treatment of equals* if any two agents with identical valuations get same utility. Clearly, envy-free implies proportionality and also symmetry. An allocation  $X$  is *Pareto optimal* if no agent can get a higher value without some other agent getting less value. Formally,  $X$  is Pareto optimal if there does not exist another allocation  $Y$  such that  $V_i(Y_i) \geq V_i(X_i)$  for all  $i \in N$  and  $V_i(Y_i) > V_i(X_i)$  for some  $i \in N$ . For any  $S \subseteq [0, 1]$ , define  $D(S) = \{i \in N | V_i(S) > 0\}$ . An allocation  $X$  is *non-wasteful* if for all  $S \subseteq [0, 1]$ ,  $S \subseteq \cup_{i \in D(S)} X_i$ . In other words, an allocation is non-wasteful if every portion of the cake desired by at least one agent is allocated to some agent who desires it.

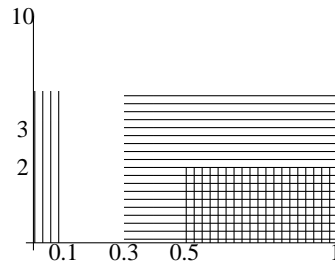
We now define robust analogues of the fairness concepts defined above. An allocation satisfies *robust proportionality* if for all  $i, j \in N$  and for all  $v'_i \in \hat{V}_i$ ,  $\int_{X_i} v'_i(x) dx \geq 1/n \int_0^1 v'_i(x) dx$ . An allocation satisfies *robust envy-freeness* if for all  $i, j \in N$  and for all  $v'_i \in \hat{V}_i$ ,  $\int_{X_i} v'_i(x) dx \geq \int_{X_j} v'_i(x) dx$ . Notice that both robust envy-freeness and robust proportionality would require each agent to get a piece of cake of the same length if every agent desires the entire cake.

We give an example of piecewise constant value density function and demonstrate how the standard concept of envy-freeness is not robust under uncertainty.

*Example 1 (A cake-cutting problem with piecewise constant valuations).* Consider the cake cutting problem in Figure 1. An allocation in which both agents get regions in which their value density function is the highest is envy-free. Agent 1 gets utility one for his allocation and has the same utility for the allocation of agent 2. However if its probability density function is slightly lower in region  $[0, 0.1]$ , then agent 1 will be envious of agent 2.



**Fig. 1.** Example of a cake cutting problem with piecewise constant value density functions. The area with vertical lines is under the value density function of agent 1 and the area with horizontal lines is under the value density function of agent 2.



**Fig. 2.** Example of a cake cutting problem with piecewise constant value density functions. The area with vertical lines is under the value density function of agent 1 and the area with horizontal lines is under the value density function of agent 2. The valuation functions of both agents are ordinally equivalent to the ones in Figure 1.

## 2.4 Properties of cake cutting algorithms

A *deterministic cake cutting algorithm* is a mapping from the set of valuation profiles to the set of allocations. A *randomized cake cutting algorithm* is a mapping from the set of valuation profiles to a space of distributions over the set of allocations. The output of the algorithm in this case for a specific valuation profile is a random sample of the distributional function over the set of allocation for that profile.

An algorithm (either deterministic or randomized) satisfies property  $P$  if it always returns (with probability 1) an allocation that satisfy property  $P$ . A deterministic algorithm is *strategyproof* if no agent ever has an incentive to misreport in order to get a better allocation. The notion of strategyproofness is the one well-established in social choice (see e.g., [17]) and much stronger than the notion of ‘strategyproofness’ used in some of the cake-cutting literature (see e.g., [6]). By strategyproof, we mean truthful as has been used in [11]. Similarly, a deterministic algorithm is *group-strategyproof* if there exists no coalition  $S \subseteq N$  such that members of  $S$  can misreport their preferences so that each agent in  $S$  gets as preferred an allocation and at least one agent gets a strictly better payoff. A deterministic algorithm is *weak group-strategyproof* if there exists no coalition  $S \subseteq N$  such that members of  $S$  can misreport their preferences so that each agent in  $S$  gets a strictly more preferred allocation. A randomized algorithm is *strategyproof in expectation* if the expected utility from the random allocation that every agent receives in expectation under a profile where he reported truthfully is at least as large as the expected that he receives under a profile where he misreports while fixing the other agents’ reports.

We say that a cake cutting algorithm satisfies *unanimity*, if when each agent’s most preferred  $1/n$  length of the cake is disjoint from another agent,  $1/n$  length of the cake, then each agent gets their most preferred piece of cake of length  $1/n$ .

## 2.5 Relation between the properties of cake cutting algorithms

In this subsection, we recap the main properties of cake cutting algorithms: i) proportionality, ii) robust proportionality, iii) envy-freeness, iv) robust envy-freeness, v) symmetry, vi) non-wastefulness, vii) Pareto optimality, viii) unanimity ix) strategyproofness, x) weak group strategyproofness, xi) group strategyproofness, xii) polynomial-time.

*Remark 1.* For cake cutting, a) Envy-freeness and non-wastefulness  $\implies$  proportionality; b) Robust proportionality  $\implies$  proportionality; c) Robust envy-freeness  $\implies$  envy-freeness; d) Robust envy-freeness and non-wastefulness  $\implies$  robust proportionality; e) Group strategyproofness  $\implies$  weak group strategyproofness  $\implies$  strategyproofness; f) Pareto optimality  $\implies$  non-wastefulness; g) Pareto optimality  $\implies$  unanimity; h) two agents, proportionality  $\implies$  envy-freeness; i) two agents, robust proportionality  $\implies$  robust envy-freeness.

## 2.6 The free disposal assumption

We may assume without lost of generality that every part of the cake is desired by at least one agent. If that is not the case, then we can discard the parts that are desired



by no one and rescale what is left so that we get a  $[0, 1]$  interval representation of the cake. Notice that this procedure preserves the aforementioned properties of fairness, efficiency and truthfulness. The free disposal assumption that we are making is necessary to ensure strategyproofness for piecewise uniform valuation functions. See [12] for a discussion on the necessity of this assumption.

Before we present our algorithms, we will first take a detour to the literature on random assignments, as some of the algorithms in the random assignment literature are closely related to our algorithms.

### 3 A detour to random assignments

An assignment problem is a triple  $(N, H, \succsim)$  such that  $N = \{1, \dots, n\}$  is a set of agents,  $H = \{h_1, \dots, h_n\}$  is a set of houses and  $\succsim = (\succsim_1, \dots, \succsim_n)$  is the preference profile in which  $\succsim_i$  denotes the preferences of agent  $i$  over houses  $H$ . A *deterministic assignment* is a one-to-one mapping from  $N$  to  $H$ . A *random allocation* is a probability distribution over  $H$ . A random assignment  $p$  gives a random allocation to each agent. It can be represented by a bistochastic matrix  $p$  in which the  $i$ th row is denoted by  $p_i$  and all  $i \in A$ , and  $h \in H$ ,  $p_{ih} \geq 0$ ,  $\sum_{j \in A} p_{jh} = \sum_{h' \in H} p_{ih'} = 1$ .<sup>8</sup> The term  $p_{ih}$  denotes the probability with which agent  $i$  gets house  $h$ . An assignment problem has commonalities with cake cutting with piecewise constant valuations. They also have some fundamental differences. For example, in cake cutting, the agents do not have continuous constant valuations over pre-determined segments of the cake.

Given two random assignments  $p$  and  $q$ ,  $p_i \succsim_i^{SD} q_i$  i.e., a player  $i$  weakly SD prefers  $p_i$  to  $q_i$  if for all  $h$ ,  $\sum_{h_j \in \{h_k: h_k \succsim_i h\}} p_{ih_j} \geq \sum_{h_j \in \{h_k: h_k \succsim_i h\}} q_{ih_j}$ . Another way to see the SD relation is as follows. A player  $i$  weakly SD prefers allocation  $p_i$  to  $q_i$  if for all vNM utilities consistent with his ordinal preferences,  $i$  gets at least as much expected utility under  $p_i$  as he does under  $q_i$ . Furthermore  $p \succsim^{SD} q$ , i.e.,  $p$  *stochastically dominates*  $q$  if  $p_i \succsim_i^{SD} q_i$  for all  $i \in N$  and  $q \neq p$ .

An algorithm satisfies *SD-efficiency* if each returned assignment is Pareto optimal with respect to the SD-relation (see e.g., [34]). An algorithm satisfies *SD envy-freeness* if each agent (weakly) SD prefers his allocation to that of any other agent. SD envy-freeness is a very demanding notion of fairness. The reader may be able to notice that our notion of robust envy-freeness in cake cutting is based on a similar idea as SD envy-freeness. We will consider random allocations as fractional allocations and random assignments as fractional assignments. Viewing the probability of getting a house simply as getting a fraction of the house is especially useful when some houses are not complete but only partial. In this vein, the definition of SD dominance should also be considered from the perspective of fractional allocations rather than probability distributions.

The most basic assignment problem concerns  $n$  agents having strict preferences over  $n$  objects. For this basic setting a simple yet ingenious *PS (probabilistic serial)* algorithm introduced by Bogomolnaia and Moulin [3] and Crès and Moulin [14] which

<sup>8</sup> The bistochasticity of the matrix  $p$  holds when there are the same number of agents as there are objects, which can be assumed without loss of generality by adding dummy agents or objects

uses the *simultaneous eating algorithm (SEA)*. Each house is considered to have a divisible probability weight of one, and agents simultaneously and with the same eating rate consume the probability weight of their most preferred house until no house is left. The random allocation allocated to an agent by PS is then given by the amount of each object he has eaten until the algorithm terminates. The main result of Bogomolnaia and Moulin [3] was that the fractional assignment returned by the PS algorithm is SD envy-free and SD-efficient.

The PS algorithm has been extended in various ways. The EPS (extended PS algorithm) of Katta and Sethuraman [18] generalized PS to the case for indifferences using parametric network flows. EPS also generalized the *egalitarian rule* of Bogomolnaia and Moulin [5] for dichotomous preferences. Yilmaz [34] and Athanassoglou and Sethuraman [1] extended the work of [18] to propose PS generalization which also takes care of private endowments where  $e_i$  indicates the endowment of agent  $i$ . For the case of endowments, Yilmaz [34] introduced the idea of justified envy-freeness. An assignment  $p$  satisfies justified envy-freeness if for all  $i, j \in N$ ,  $p_i \succeq_i^{SD} p_j$  or  $\neg(p_i \succeq_j^{SD} e_j)$ . The algorithms in [34, 1] satisfy justified envy-freeness in the presence of private endowments. In our algorithm CCEA, we rely on the full power of the controlled-consuming (CC) algorithm of Athanassoglou and Sethuraman [1] which combines almost all the desirable features of other extensions of PS. In particular, we use the following fact. *There exists an extension of the PS algorithm which can simultaneously handle indifferences in preferences, unacceptable objects in preferences, allocation of multiple objects to agents, agent owning fractions of houses, partial houses being available, and still returns an assignments which satisfies SD justified envy-freeness and SD-efficiency.* In addition, if there are no private endowments, then the extension can also handle variable eating rates. The Controlled-Consuming (CC) algorithm of Athanassoglou and Sethuraman [1] can handle the case where each agent owns fractions of the complete houses. We also require that for some houses, only an arbitrary fraction of the house is available to the market. This can be handled by a modification to CC (page 30, [1]). Finally we require the agents to want to be allocated as many houses as possible. This does not require any modification to CC. In the absence of endowments but presence of variable eating rates, CC is equivalent to the EPS algorithm that can also cater for variable eating rates (Section 6.4, [18]).

## 4 CCEA — Controlled Cake Eating Algorithm

CCEA is based on CC (Controlled Consuming) algorithm of Athanassoglou and Sethuraman [1]. Since the original PS algorithm utilized the simultaneous eating algorithm, hence the name *Controlled Cake Eating Algorithm*. CCEA first divides the cake up into disjoint intervals each of whose endpoints are consecutive points of discontinuity of the agents' valuation functions. We will refer to these intervals as *intervals induced by the discontinuity points*. The idea is to form a one-to-one correspondence of the set of cake intervals with a set of houses of an assignment problem. Since intervals may have different lengths, we consider the house corresponding to the interval with the maximum length as a complete house where as intervals corresponding to other houses are partial houses. The preferences of agents over the houses are naturally induced by the relative

height of the piecewise constant function lines in the respective intervals. If an agent  $i$  owns a sub-interval  $J_j$ , then in the housing setting,  $e_{ih_j}$  is set to  $size(h_j)$  and not to one. The reason is that an agent can only own as much of the house as exists. The technical heart of the algorithm is in CC (Controlled Consuming) algorithm of Athanassoglou and Sethuraman [1]. We recommend the reader to Section 3.2 of [1] in which an illustrative example on CC is presented. Once CC has been used to compute a fractional assignment  $p$ , it is straightforward to compute a corresponding cake allocation. If an agent  $i$  gets a fraction of house  $h_j$ , then in the cake allocation agent  $i$  gets the same fraction of subinterval  $J_j$ .

**Input:** Piecewise constant value functions.  
**Output:** A robust envy-free allocation.

- 1 Divide the regions according to agent value functions. Let  $\mathcal{J} = \{J_1, \dots, J_m\}$  be the set of subintervals of  $[0, 1]$  formed by consecutive marks.
- 2 Consider  $(N', H, \succsim, size(\cdot))$  where
  - $H = \{h_1, \dots, h_m\}$  where  $h_i = J_i$  for all  $i \in \{1, \dots, m\}$
  - $\succsim$  is defined as follows:  $h \succsim_i h'$  if and only if  $v_i(x) \geq v_i(y)$  for  $x \in h$  and  $y \in h'$ ;  $\succsim_{n+1}$  is defined in the way that each house in the market is unacceptable to agent  $n + 1$ .
  - $size(h_j) = 1$  for  $h_j \in \arg \max_{j \in \{1, \dots, m\}} (len(J_j))$ ;  
 $size(h_j) = \frac{len(J_j)}{(\max_{j \in \{1, \dots, m\}} (len(J_j)))}$   
 for all  $h_j \notin \arg \max_{j \in \{1, \dots, m\}} (len(J_j))$ ;
- 3  $p \leftarrow CC(N', H, \succsim, size(\cdot))$
- 4 For interval  $J_j$ , agent  $i$  is an allocated subinterval of  $J_j$ , denoted by  $J_j^i$ , which is of length  $p_{ih_j} / size(h_j) \times len(J_j)$ . For example, if  $J_j = [a_i, b_i]$ , then one possibility of  $J_j^i$  can be  $[a_i + \sum_{n=1}^{i-1} p_{ih_j} / size(h_j) \times len(J_j), a_i + \sum_{n=1}^i p_{ih_j} / size(h_j) \times len(J_j)]$ .  
 $X_i \leftarrow \bigcup_{j=1}^m J_j^i$  for all  $i \in N$
- 5 **return**  $X = (X_1, \dots, X_n)$

**Algorithm 1:** Controlled Cake Eating Algorithm (CCEA) to compute a robust envy-free allocation for piecewise constant value functions.

In Example 2, we show how CCEA transforms a cake cutting problem with piecewise constant valuations into a random assignment problem.

*Example 2 (Illustration of CCEA).* We examine how CCEA runs on the cake cutting problem in Figure 1. Firstly, let  $\mathcal{J} = \{J_1, \dots, J_4\}$  be the set of subintervals of  $[0, 1]$  formed by consecutive points of discontinuity are identified:  $J_1 = [0, 0.1]$ ,  $J_2 = [0.1, 0.3]$ ,  $J_3 = [0.3, 0.5]$ , and  $J_4 = [0.5, 1]$ .  $J_2$  is discarded because it is desired by no agent. In set  $\{h_1, h_3, h_4\}$ , each house  $h_j$  corresponds to subinterval  $J_j$ . The preferences of the agents over  $H$  are inferred from their valuation function height in the subintervals so that  $h_1 \succ_1 h_4 \succ_1 h_3$  and  $h_3 \sim_2 h_4 \succ_2 h_1$ . We also set the number of units of each house that is available. Since  $J_4$  is the biggest interval, we consider  $h_4$  as complete house.

So,  $size(h_4) = 1$ ,  $size(h_1) = 0.2$ , and  $size(h_3) = 0.4$ . If we run CC over the housing market instance with the specified set of agents, houses, fraction of houses available to the market, and agent preferences, then the assignment returned by CC is as follows:  $p_{1h_1} = 0.2$ ,  $p_{1h_3} = 0$ ,  $p_{1h_4} = 0.6$ ,  $p_{2h_1} = 0$ ,  $p_{2h_3} = 0.4$ , and  $p_{2h_4} = 0.4$ . The house assignment  $p$  can be used to divide the subintervals among the agents:  $X_1 = \{[0, 0.1], [0.7, 1]\}$  and  $X_2 = \{[0.3, 0.5], [0.5, 0.7]\}$ .

CCEA satisfies the strong fairness property of robust envy-freeness.

**Proposition 1.** *For piecewise constant valuations, CCEA satisfies robust envy-freeness and non-wastefulness.*

Let  $m$  be the number of relevant subintervals in a cake cutting problem with piecewise constant valuations.

**Proposition 2.** *CCEA runs in time  $O(n^5 m^2 \log(n^2/m))$ , where  $n$  is the number of agents and  $m$  is the number of subintervals defined by the union of discontinuity points of the agents' valuation functions.*

Although CCEA satisfies the demanding property of robust envy-freeness, it is not immune to manipulation. We show that CCEA is not strategyproof even for two agents. In the next section, we will present a different algorithm that is both robust envy-free and strategyproof for two agents.

**Proposition 3.** *For piecewise constant valuations, CCEA is not strategyproof even for two agents.*

If we restricted the preferences to piecewise uniform with no private endowment or variable claims, then CCEA is not only strategyproof but group-strategyproof. We first show that in this restricted setting, CCEA is in fact equivalent to the algorithm of [11].

**Lemma 1.** *For piecewise uniform value functions with no private endowments and variable claims, CCEA is equivalent to Mechanism 1 of Chen et al. [11].*

Since the set of valuations that can be reported is bigger in cake cutting than in the assignment problem, establishing group strategyproofness does not follow automatically from group-strategyproofness of CC for dichotomous preferences (Theorem 2, [5]). Using similar arguments, we give a detailed proof that CCEA and hence Mechanism 1 of Chen et al. [11] is group strategyproof for piecewise uniform valuations.<sup>9</sup> In Section 8, we extend the result to the case where agents may have variable claims.

**Proposition 4.** *For cake cutting with piecewise uniform value functions, CCEA is group strategyproof.*

For piecewise uniform valuations, CCEA is also Pareto optimal. The result follows directly from lemma 1 along with the fact that Mechanism 1 of Chen et al. [11] is Pareto optimal.

**Proposition 5.** *For cake cutting with piecewise uniform value functions, CCEA is Pareto optimal.*

<sup>9</sup> Chen et al. [11] had shown that their mechanism for piecewise uniform valuations is strategyproof.

## 5 MEA — Market Equilibrium Algorithm

In the previous section we presented CCEA which is not Pareto optimal for piecewise constant valuations. It turns out that if we relax the robust notion of fairness to envy-freeness, then we can use fundamental results in general equilibrium theory and recent algorithmic advances [15] to formulate a convex program that always returns an envy-free and Pareto optimal allocation as its optimal solution. For each valuation profile, let  $J = \{J_1, \dots, J_k\}$  be the intervals whose endpoints are consecutive points in the union of break points of the agents' valuation functions. Let  $x_{ij}$  be the length of any subinterval of  $J_i$  that is allocated to agent  $j$ . Then we run a convex program to compute a Pareto optimal and envy-free allocation. We will call the convex program outlined in Algorithm 2 as the *Market Equilibrium Algorithm (MEA)*. MEA is based on computing the market equilibrium via a primal-dual algorithm for a convex program that was shown to be polynomial-time solvable by Devanur et al. [15]. Notice that if we ignore strategyproofness, or in other words, if we assume that agents report truthfully, then agents are truly indifferent between which subinterval they receive since their valuation function is a constant over any  $J_i$ . Hence, once we determine the length of  $J_j$  to be allocated to an agent, we can allocate any subinterval of that length to the agent.

**Input:** Cake-cutting problem with piecewise constant valuations.

**Output:** A proportional, envy-free, and Pareto optimal allocation.

- 1 Let  $J = \{J_1, \dots, J_k\}$  be the intervals whose endpoints are consecutive points in the union of break points of the agents' valuation functions.
- 2 Let  $x_{ij}$  be the length of any subinterval of  $J_i$  that is allocated to agent  $j$ .
- 3  $l_i \leftarrow \text{len}(J_i)$
- 4 Solve the following convex program.

$$\begin{aligned}
 \min \quad & - \sum_{j=1}^n \log(u_j) \\
 \text{s.t.} \quad & u_j = \sum_{i=1}^k v_{ij} x_{ij} \quad \forall j = 1, \dots, n \\
 & \sum_{j=1}^n x_{ij} \leq l_i \quad \forall i = 1, \dots, k \\
 & x_{ij} \geq 0 \quad \forall i, j.
 \end{aligned}$$

- 5 Let  $u_j^*, x_{ij}^*$  be an optimal solution to the convex program. Partition every interval  $J_i$  into  $n$  subintervals where the  $j$ -th subinterval  $J_i^j$  has length  $x_{ij}^*$ .
- 6  $Y_j \leftarrow \cup_{i=1}^k J_i^j$  be the allocation of each  $j = 1, \dots, n$ .
- 7 **return**  $Y = (Y_1, \dots, Y_n)$ .

**Algorithm 2:** The Market Equilibrium Algorithm to compute a Pareto optimal, envy-free, and proportional allocation.

**Proposition 6.** *MEA is polynomial-time, Pareto efficient and envy free.*

We mention here that the connection between a fair and efficient algorithm for cake cutting and computing market equilibria was already pointed by Reijnierse and Potters [24]. Reijnierse and Potters [24] presented an algorithm to compute an approximately envy-free and Pareto optimal allocation for cake cutting with general valuations. However their algorithm is not polynomial-time even for piecewise constant valuations [37].

MEA requires the machinery of convex programming. It remains open whether MEA can be implemented via linear programming. Cohler et al. [13] presented an algorithm that uses a linear program to compute an optimal envy-free allocation. The allocation is Pareto optimal among all envy-free allocations. However it need not be Pareto optimal in general.

Although MEA is not robust envy-free like CCEA, it is Pareto optimal because it maximizes the Nash product. What is interesting is that under uniform valuations, both MEA and CCEA are equivalent. In the next result we demonstrate this equivalence (Proposition 7). The proof requires a careful comparison of both CCEA and MEA under uniform valuations.

**Proposition 7.** *For piecewise uniform valuations, the allocation given by CCEA is identical to that given by MEA.*

**Corollary 1.** *For piecewise uniform valuations, MEA is group-strategyproof.*

Thus if we want to generalize Mechanism 1 of Chen et al. [11] to piecewise constant valuations and maintain robust envy-freeness then we should opt for CCEA. On the other hand, if want to still achieve Pareto optimality, then MEA is the appropriate generalization. In both generalization, we lose strategyproofness.

## 6 Impossibility Results

Thus far, we presented two polynomial time algorithms, each of which satisfies a different set of properties. CCEA is robust envy-free and non-wasteful, whereas MEA is Pareto optimal and envy-free. This naturally leads to the following question: does there exist an algorithm satisfies all of the properties that CCEA and MEA satisfy? It turns out that the answer is no, as Theorem 5 shows that there is no algorithm that is both Pareto efficient and robust proportional. Similarly, Theorem 6 argues that there is no algorithm that satisfies the properties CCEA satisfies along with strategyproofness. Lastly, Theorem 7 argues that there is no algorithm that satisfies the properties CCEA satisfies plus strategyproofness. The impossibility results are summarized in Table 2.

Impossibility	Reference
Pareto efficient and robust proportional	Theorem 5
Strategyproof, Pareto optimal, and proportional	Theorem 6
Strategyproof, robust proportional, and non-wasteful	Theorem 7

**Table 2.** Impossibility results for cake-cutting algorithms for piecewise constant valuations.

Consequently, we may conclude that the properties satisfied by CCEA and MEA are respectively maximal subsets of properties that an algorithm can satisfy for piecewise constant valuations.

## 7 CSD — Constrained Serial Dictatorship Algorithm

In the previous sections, we saw that CCEA and MEA are only strategyproof for piecewise uniform valuations. In light of the impossibility results established in the previous section, it is reasonable to ask what other property along with strategyproofness can be satisfied by some algorithm. It follows from (Theorem 3, [26]) that the only type of strategyproof and Pareto optimal mechanisms are dictatorships. Chen et al. [12] raised the question whether there exists a strategyproof and proportional algorithm for piecewise constant valuations. The algorithm CSD answers this question partially.

Before diving into the CSD algorithm, it is worth noting that there is some fundamental difference between random assignment setting and the cake cutting setting. In the random assignment setting, the objects that we are allocating are well defined and known to the public. On the other hand, in the cake cutting setting, the discontinuity points of each agent's valuation function is private information for the agent. Hence, any algorithm that uses the reported discontinuity points to artificially create the objects runs into the risk of having the objects created by the algorithm be manipulated by the reports of the agents. In order to illustrate this difficulty, consider the uniform allocation rule. The uniform allocation rule (that assigns  $1/n$  of each house) [10] is both strategyproof and proportional in the random assignment setting. However it cannot be adapted for cake cutting with piecewise constant valuations since strategyproofness is no longer satisfied if allocating  $1/n$  of each interval (induced by the agent valuations) is done deterministically.

**Proposition 8.** *The uniform allocation rule (done deterministically) is not strategyproof.*

Now we are ready to present CSD. In order to motivate CSD, we will give a randomized algorithm that is strategyproof and robust proportional in expectation. The algorithm is a variant of random dictatorship: each agent has uniform probability of being chosen as a dictator. However, if the whole cake is acceptable to each agent, then each time a dictator is chosen, he will take the whole cake. This approach is not helpful since we return to square one of having to divide the whole cake. We add an additional requirement which is helpful. We require that each time a dictator is chosen, the piece he takes has to be of maximum value  $1/n$  length of the total size of the cake. We will call this algorithm Constrained Random Serial Dictatorship (CRSD). Formally speaking, CRSD draws a random permutation of the agents. The algorithm then makes the allocation to agents in the order that the lottery is drawn. Everytime when it is agent  $i$ 's turn to receive his allocation, CRSD looks at the remaining portion of the cake and allocates a maximum value  $1/n$  length piece of the cake to agent  $i$  (break ties arbitrarily). Notice that CRSD is strategyproof, as in every draw of lottery, it is in the best interest of the agents to report their valuation function truthfully in order to obtain a piece that maximizes his valuation function out of the remaining pieces of cake. Later

**Input:** Cake-cutting problem with piecewise constant valuations.  
**Output:** A robust proportional allocation.

```

1 for each  $\pi \in \Pi^N$  do
2    $C \leftarrow [0, 1]$  (intervals left)
3   for  $i = 1$  to  $n$  do
4      $X_{\pi(i)}^\pi \leftarrow$  maximum preference cake piece of size  $1/n$  from  $C$ 
5      $C \leftarrow C - X_{\pi(i)}^\pi$ 
6      $i \leftarrow i + 1$ .
7   end for
8 end for
9 Construct a disjoint and exhaustive interval set  $\mathcal{J}'$  induced by the discontinuities in agent
  valuations and the cake cuts in the  $n!$  cake allocations.
10  $Y_i \leftarrow$  empty allocation for each  $i \in N$ .
11 for each  $J_j = [a_j, b_j] \in \mathcal{J}'$  do
12   for each  $i \in N$  do
13     Let  $p_{ij} = \frac{\text{count}(i, J_j)}{n!}$  where  $\text{count}(i, J_j)$  is the number of permutations in which  $i$  gets  $J_j$ .
14     Generate  $A_{ij} \subseteq J_j$ , which is of length  $p_{ij}|J_j|$  according to some subroutine.
15      $Y_i \leftarrow Y_i \cup A_{ij}$ 
16   end for
17 end for
18 return  $Y = (Y_1, \dots, Y_n)$ 

```

**Algorithm 3:** CSD (Constrained Serial Dictatorship)— proportional and unanimous algorithm for piecewise constant valuations

on we will see, through the proof of Proposition 10, that CRSD is robust proportional in expectation.

CSD is an algorithm that derandomizes CRSD by looking at its allocation for all  $n!$  different permutations and aggregate them in a suitable manner. The algorithm is formally presented as Algorithm 3.

Although CSD does not necessarily require  $n!$  cuts of the cake, it may take exponential time if the number of agents is not a constant. In Example 3, we illustrate how CSD works.

*Example 3 (Illustration of CSD).* We implement CSD on the cake cutting problem in Figure 1. For permutation 12, agent 1 first chooses the cake piece  $\{[0, 0.1], [0.6, 1]\}$  and agent 2 then takes the remaining piece  $\{[0.1, 0.6]\}$ . For permutation 21, agent 2 first chooses the cake piece  $\{[0.3, 0.8]\}$  and agent 1 then takes the remaining piece  $\{[0, 0.3], [0.8, 1]\}$ .

The set of all relevant subintervals induced by the two permutations are  $\{[0, 0.1], [0.1, 0.3], [0.3, 0.6], [0.6, 0.8], [0.8, 1]\}$ . When we additionally consider the discontinuities in the players' valuations, the set of relevant subintervals is  $\mathcal{J}' = \{[0, 0.1], [0.1, 0.3], [0.3, 0.5], [0.5, 0.6], [0.6, 0.8], [0.8, 1]\}$ .

Then

$$Y_1 = \{[0, 0.1], \frac{1}{2}[0.1, 0.3], \frac{1}{2}[0.6, 0.8], [0.8, 1]\}$$



- 1: Generate  $U_j \sim \text{unif}[a_j, b_j]$ .
- 2: For  $a_j \leq x \leq 2b_j - a_j$ , let  $\text{mod}(x) = x$  if  $a_j \leq x \leq b_j$  and  $x - (b_j - a_j)$  if  $x > b_j$ . Let

$$A_{ij} = [\text{mod}(U_j + \sum_{k=1}^{i-1} p_{kj}(b_j - a_j)), \text{mod}(U_j + \sum_{k=1}^i p_{kj}(b_j - a_j))]$$

if  $\text{mod}(U_j + \sum_{k=1}^{i-1} p_{kj}(b_j - a_j)) \leq \text{mod}(U_j + \sum_{k=1}^i p_{kj}(b_j - a_j))$  and

$$A_{ij} = [a_j, \text{mod}(U_j + \sum_{k=1}^i p_{kj}(b_j - a_j))] \cup [\text{mod}(U_j + \sum_{k=1}^{i-1} p_{kj}(b_j - a_j)), b_j]$$

otherwise.

**Algorithm 4:** A subroutine that converts fractional allocation into subintervals via randomization

and

$$Y_2 = \{\frac{1}{2}[0.1, 0.3], [0.3, 0.5], [0.5, 0.6], \frac{1}{2}[0.6, 0.8]\}.$$

**Proposition 9.** *For piecewise constant valuations, CSD is well-defined and returns a feasible cake allocation in which each agents gets a piece of size  $1/n$ .*

**Proposition 10.** *For piecewise constant valuations, CSD satisfies robust proportionality and symmetry.*

Notice that unlike CRSD, CSD interprets the probability of allocating each interval to an agent as allocation a fractional portion of the interval to that agent. Unless the actual way of allocating the fractions is specified, one cannot discuss the notion of strategyproofness for CSD because a deviating agent is unable to properly evaluate his allocation against his true valuation function in the reported profile. Contrary to intuition, CSD may or may not be strategyproof depending on how the fractional parts of each interval are allocated. In fact, the following remark illustrates this issue.

*Remark 2.* CSD is not strategyproof if the fraction of each interval of  $\mathcal{J}'$  is allocated deterministically (please see the appendix for the proof).

In light of difficulty, we will implement a method (see Algorithm 4) that randomly allocating the fractions of intervals to agents. With this implementation, CSD is strategyproof in expectation.

We will refer this randomized implementation of CSD as *Constrained Mixed Serial Dictatorship*, or *CMSD* for short.

**Proposition 11.** *CSD implemented with the aforementioned random allocation rule is strategyproof in expectation.*

Although CSD is strategyproof in expectation, it fails to satisfy truthfulness based on group-based deviations no matter how the fractional parts of each interval are allocated.

**Proposition 12.** *For cake cutting with piecewise constant valuations, CSD is not weakly group-strategyproof even for two agents.*

Moreover, for cake cutting with piecewise uniform valuations, CSD is not weakly group-strategyproof for at least seven agents. The statement above follows from the fact that RSD is not weakly group-strategyproof for dichotomous preferences when there are at least seven agents [4, 5].

Even though CSD satisfies both proportionality and symmetry, it does not satisfy the stronger notion of envy-freeness.

**Proposition 13.** *CSD is not necessarily envy-free for three agents even for piecewise uniform valuations.*

Another drawback of CSD is that it is not Pareto optimal for piecewise constant valuations. The statement follows from the fact that RSD is not SD-efficient [3]. However for the case of two agents, it is robust envy-free and polynomial-time.

**Proposition 14.** *For two agents and piecewise constant valuations, CSD is robust envy-free, and polynomial-time but not Pareto optimal.*

*Remark 3.* For piecewise uniform valuations, CSD can be modified to be made Pareto optimal. The main change is that for each permutation CSD, the resultant outcome needs to be made Pareto optimal. This can be done by using the idea in [2].

## 8 Extensions

In this section, we show how some of our positive results concerning CCEA extend to more general settings where agents may have variable claims or they may have initial endowments (please see Algorithm 5).

### 8.1 Variable claims

We consider the concept of *variable claims* in cake cutting. It is generally assumed in the literature that each agent has equal claim to the cake. In this case we modify the definition of proportionality and envy-freeness accordingly. Reasoning about claims has a long tradition in the fair division literature. However this strand of research assumes that agents have uniform and identical preferences over the whole of the divisible object and their only concern is the proportion of the object they get (see e.g., [30, 35]). Brams and Taylor [8] refer to variable claims as entitlements and touch upon entitlements in cake cutting at a few places in their book. They present a general idea of handling entitlements by cloning agents (Page 152, [8]) but doing so can lead to an exponential blowup of time and space. Brams et al. [9] also considered entitlements in pie-cutting but presented an impossibility result and also a positive result for two agents. On the other hand, one of our algorithms handles variable claims for arbitrary number of agents and does not require cloning.

**Input:** Piecewise constant value functions with priority claims  $(c_1, \dots, c_n)$  or private endowments  $(\omega_1, \dots, \omega_n)$

**Output:** A robust envy-free and individually rational allocation.

- 1 Let  $N' = N \cup \{n+1\}$  where  $n+1$  is the agent owning all public cake but with no interest in any of the cake. Join the segments  $\omega(1), \dots, \omega(n), \omega(n+1)$  to assemble a cake.
- 2 Divide the regions according to agent value functions. Let  $\mathcal{J} = \{J_1, \dots, J_m\}$  be the set of subintervals of  $[0, 1]$  formed by consecutive marks.
- 3 Consider  $(N', H, \succsim, e, \text{rate}(\cdot), \text{size}(\cdot))$  where
  - $H = \{h_1, \dots, h_m\}$  where  $h_i = J_i$  for all  $i \in \{1, \dots, m\}$
  - $\succsim$  is defined as follows:  $h \succsim_i h'$  if and only if  $v_i(x) \geq v_i(y)$  for  $x \in h$  and  $y \in h'$ ;
  - $\succsim_{n+1}$  is defined in the way that each house in the market is unacceptable to agent  $n+1$ .
  - $\text{size}(h_j) = 1$  for  $h_j \in \arg \max_{j \in \{1, \dots, m\}} (\text{len}(J_j))$ ;
  - $\text{size}(h_j) = \frac{\text{len}(J_j)}{(\max_{j \in \{1, \dots, m\}} (\text{len}(J_j)))}$  for all  $h_j \notin \arg \max_{j \in \{1, \dots, m\}} (\text{len}(J_j))$ ;
  - $\text{rate}(i) = \text{claim}(i)$ ;
  - $e_{ih_j} = \text{size}(h_j)$  if  $J_j \in \omega(i)$  and zero otherwise.
- 4  $p \leftarrow CC(N', H, \succsim, e, \text{rate}(\cdot), \text{size}(\cdot))$
- 5 For interval  $J_j$ , agent  $i$  is allocated a subinterval of  $J_j$ , denoted by  $J_j^i$ , which is of length  $p_{ih_j} / \text{size}(h_j) \times \text{len}(J_j)$ .  
 $X_i \leftarrow \bigcup_{j=1}^m p_{ih_j} / \text{size}(h_j) J_j$  for all  $i \in N$
- 6 **return**  $X = (X_1, \dots, X_n)$

**Algorithm 5:** Controlled Cake Eating Algorithm (CCEA) to compute a robust envy-free allocation for piecewise constant value functions. CCEA works for either private endowments or variable claims but is not defined for both.

An allocation satisfies *proportionality for variable claims* if  $V_i(X_i) \geq c_i / \sum_{j \in N} c_j V_i([0, 1])$  for all  $i \in N$ . An allocation satisfies *robust proportionality for variable claims* if for all  $i \in N$  and for all  $v'_i \in \hat{V}_i$ ,  $\int_{X_i} v'_i(x) dx \geq (c_i / \sum_{j \in N} c_j) \int_{[0, 1]} v'_i(x) dx$ .

An allocation satisfies *envy-freeness for variable claims*, if  $V_i(X_i) \geq (c_i / c_j) V_i(X_j)$  for all  $i, j \in N$ .

An allocation satisfies *robust envy-freeness for variable claims* if for all  $i, j \in N$  and for all  $v'_i \in \hat{V}_i$ ,  $\int_{X_i} v'_i(x) dx \geq (c_i / c_j) \int_{X_j} v'_i(x) dx$ .

**Proposition 15.** *For cake cutting with variable claims and piecewise constant valuations, CCEA satisfies robust envy-freeness for variable claims.*

For uniform valuations, group-strategyproofness of CCEA even holds when agents have variable claims.

**Proposition 16.** *For cake cutting with piecewise uniform value functions and variable claims, CCEA is group-strategyproof.*

On the other hand, it can be shown that the following natural modification of CSD does not satisfy proportionality in the presence of variable claims: each agent  $i$  gets  $c_i / (\sum_{j \in N} c_j)$  length of the most preferred interval for each permutation.

We note that MEA can be easily adapted to cater for variable claims. Each agent's budget is proportional to his claim.

**Proposition 17.** *For cake cutting with piecewise constant valuations and variable claims, there exists an algorithm (MEA) that is polynomial-time, Pareto optimal, envy-free, and proportional.*

## 8.2 Private endowments

In the cake cutting literature, the cake is generally considered to be a public good which is divided among the agents. It could also be that each agent brings part of the cake and then the assembled cake needs to be reassigned among the agents fairly. We call the setting cake cutting with *private endowments*. In this context, individual rationality is a minimal requirement and envy-freeness need to be redefined. Classic cake cutting can be modelled by cake cutting with private endowments in the following manner: none of the agents have any endowments and one additional agent owns the whole cake but has no interest in the cake.

An allocation is *individually rational* if  $V_i(X_i) \geq V_i(\omega(i))$ . An allocation satisfies *justified envy-freeness for private endowments* if  $V_i(X_i) \geq V_i(X_j)$  or  $V_j(X_i) < V_j(\omega(j))$  for each pair of agent  $i, j \in N$ . Either  $i$  should not be envious of  $j$  or if he is envious, then giving  $i$ 's allocation to  $j$  violates individual rationality.<sup>10</sup> It is not entirely clear how to obtain the canonical concept of justified envy-freeness which takes into account both private endowments and variable claims.

An allocation satisfies *robust justified envy-freeness for private endowments* if for all  $i, j \in N$  either  $\forall v'_i \in \hat{V}_i: \int_{X_i} v'_i(x) dx \geq \int_{X_j} v'_i(x) dx$  or  $\exists v'_j \in \hat{V}_j: \int_{X_i} v'_j(x) dx < \int_{\omega(j)} v'_j(x) dx$ . If  $i$  does not get an unambiguously as preferred a piece as  $j$ 's,  $j$  has a counter claim that  $j$ 's endowment is better than  $i$ 's piece for some ordinally equivalent valuation function of  $j$ .

**Proposition 18.** *For cake cutting with private endowments and piecewise constant valuations, CCEA satisfies robust justified envy-freeness for private endowments.*

Generalizing CSD to handle endowments without losing its properties seems to be a challenging task. We have not addressed it in this paper.

## 9 Related work and discussion

A mathematical analysis of cake cutting started with the work of Polish mathematicians Steinhaus, Knaster, and Banach [27]. As applications of fair division have been identified in various multiagent settings, a topic which was once considered a mathematical curiosity has developed into a full-fledged sub-field of mathematical social sciences (see e.g., [22]). In particular, in the last few decades, the literature of cake cutting has grown considerably (see e.g., [22, 8, 25, 23]).

<sup>10</sup> The definition of justified envy-freeness for private endowments is based on a similar concept due to Yilmaz [34] for the domain of housing assignment problems.

The relation between the random assignment problem and cake cutting has been noticed before [12]. However, in their discussion of related work, Chen et al. argue that techniques from the random assignment literature cannot be directly applied even to piecewise uniform functions—a subclasses of piecewise constant functions. Stumbling blocks identified include the fact that in the random assignment problem, each agents gets one object. We observed that PS satisfies envy-freeness even when agents get multiple objects.<sup>11</sup> Another issue raised is that ‘if two agents desire two subintervals, both agents would value the longer subinterval more than the shorter.’ This problem can be circumvented by using the algorithm to compute the greatest common denominator of the lengths of the intervals and then dividing the intervals in equally sizes subintervals. However the number of subintervals can be exponential in the input size. We use the idea of Athanassoglou and Sethuraman [1] that some houses may only be partially available.

Chen et al. [12] stated that generalizing their strategyproof algorithm for piecewise uniform valuations to the case for piecewise constant valuations as an open problem. We presented two algorithms — CCEA and MEA — that generalize Mechanism 1 of Chen et al. [11, 12]. Although they both satisfy certain desirable properties, both natural generalizations are not strategyproof.

A number of works in the cake-cutting literature reason about strategyproofness. However they refer to a very weak notion of strategyproofness which is equivalent to a maximin strategy. Apart from the papers of Chen et al. [11, 12], we are aware of no positive results regarding discrete, strategyproof, and fair algorithms even for the restricted domain of piecewise constant valuations. In this paper we present a proportional algorithm (CSD) for piecewise constant valuations that although not formally strategyproof seems less like likely to manipulate in comparison to CCEA. If we are allowed to use randomization, then we show that CSD can be adapted to be strategyproof in expectation. Notice that if we instead require our algorithm to be strategyproof always, and proportional in expectation, then CRSD would satisfy these properties. Moreover, we note that Chen et al. [11] also presented an algorithm that is randomized, envy-free and proportional, and strategyproof in expectation. However, it does not satisfy unanimity and requires much more randomization in contrast to CSD. It remains an open question whether there exists an algorithm that always returns a proportional allocation and is always immune to agent manipulation.

A difficulty that arises in coming up with strategyproof and proportional algorithm lies in that there is no restriction on the distribution of the discontinuity points of the agents’ valuation functions. To illustrate this point, suppose the algorithm designer knows that the discontinuity points of the agents’ valuation functions come from a set  $S = \{x_1, \dots, x_k\}$ , where  $0 \leq x_1 \leq \dots \leq x_k \leq 1$ . Consequently, a mechanism that partitions  $[0, 1]$  into intervals of the form  $[x_i, x_{i+1}]$  and allocates  $1/n$  of each interval to each agent would be proportional, envy-free and strategyproof. Even if the designer does not know such a  $S$ , but instead we require the minimum distance between any two consecutive discontinuity points of the agent’s valuation function to be at least some  $\epsilon > 0$ , then there exists a strategyproof and  $\delta$ -proportional algorithm for this setting.

<sup>11</sup> Even in the first paper on the PS algorithm for strict preferences, it was observed that PS can be extended to the case where there are many more objects than agents (Page 310, [3]).

As for private endowments, we are only aware of [31] where private endowment are considered in pie-cutting. However, in comparison to our positive results concerning envy-freeness, Thomson [31] presents a negative result concerning the core.

## 10 Conclusion

In this paper, we made a number of conceptual and technical contributions. It will be interesting to consider new results with respect to robust versions of fairness, endowments, and variable claims. Cake cutting is an exciting sub-field of microeconomics with numerous applications to computer science. In order for theory to be more relevant to practice, we envision exciting work in much richer models of cake cutting.

## Acknowledgements

The authors acknowledge the helpful comments of Jay Sethuraman and Simina Brânzei.

This material is based upon work supported by the Australian Government's Department of Broadband, Communications and the Digital Economy, the Australian Research Council, the Asian Office of Aerospace Research and Development through grant AOARD-12405.

## References

1. S. Athanassoglou and J. Sethuraman. House allocation with fractional endowments. *International Journal of Game Theory*, 40(3):481–513, 2011.
2. H. Aziz, F. Brandt, and P. Harrenstein. Pareto optimality in coalition formation. In G. Persiano, editor, *Proceedings of the 4th International Symposium on Algorithmic Game Theory (SAGT)*, Lecture Notes in Computer Science (LNCS), pages 93–104. Springer-Verlag, 2011.
3. A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.
4. A. Bogomolnaia and H. Moulin. Random matching and assignment under dichotomous preferences. <http://www.ruf.rice.edu/econ/papers/2001papers/03Moulin.pdf>, 2001.
5. A. Bogomolnaia and H. Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.
6. S. J. Brams. *Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures*. Princeton University Press, 2008.
7. S. J. Brams and A. D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.
8. S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.
9. S. J. Brams, M. A. Jones, and C. Klamler. Proportional pie-cutting. *International Journal of Game Theory*, 36:353–367, 2008.

10. C. Chambers. Consistency in the probabilistic assignment model. *Journal of Mathematical Economics*, 40:953–962, 2004.
11. Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia. Truth, justice, and cake cutting. In *Proc. of 24th AAAI Conference*, pages 756–761, 2010.
12. Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia. Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284–297, 2013.
13. Y. J. Cohler, J. K. Lai, D. C. Parkes, and A. D. Procaccia. Optimal envy-free cake cutting. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pages 626–631, 2011.
14. H. Crès and H. Moulin. Scheduling with opting out: Improving upon random priority. *Operations Research*, 49(4):565–577, 2001.
15. N. Devanur, C. H. Papadimitriou, A. Saberi, and V. Vazirani. Market equilibrium via a primal–dual algorithm for a convex program. *Journal of the ACM*, 55(5), 2008.
16. G. Gallo, M. Grigoriadis, and R. Tarjan. A fast parametric maximum flow algorithm and applications,. *SIAM Journal on Computing*, 18:30–55, 1989.
17. A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.
18. A.-K. Katta and J. Sethuraman. A solution to the random assignment problem on the full preference domain. *Journal of Economic Theory*, 131(1):231–250, 2006.
19. F. Kojima. Random assignment of multiple indivisible objects. *Mathematical Social Sciences*, 57(1):134–142, 2009.
20. A. Maya and N. Nisan. Incentive compatible two player cake cutting. In *Proceedings of the 8th International Workshop on Internet and Network Economics (WINE)*, pages 170–183, 2012.
21. E. Mossel and O. Tamuz. Truthful fair division. In S. Kontogiannis, E. Koutsoupias, and P. Spirakis, editors, *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT)*, volume 6386 of *Lecture Notes in Computer Science (LNCS)*, pages 288–199. Springer-Verlag, 2010.
22. H. Moulin. *Fair Division and Collective Welfare*. The MIT Press, 2003.
23. A. D. Procaccia. Cake cutting: Not just child’s play. *Communications of the ACM*, 56(7):78–87, 2013.
24. J. H. Reijnierse and J. A. M. Potters. On finding an envy-free pareto-optimal division. *Mathematical Programming*, 83:291–311, 1998.
25. J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998.
26. J. Schummer. Strategy-proofness versus efficiency on restricted domains of exchange economies. *Social Choice and Welfare*, 14:47–56, 1997.
27. H. Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.
28. W. Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8): 640–644, 1980.
29. F. E. Su. Rental harmony: Sperner’s lemma in fair division. *American Mathematical Monthly*, 10:930–942., 1999.
30. W. Thomson. Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: a survey. *Mathematical Social Sciences*, 45:249–297, 2003.
31. W. Thomson. Children crying at birthday parties. why? *Economic Theory*, 31: 501–521, 2007.

32. V. V. Vazirani. Combinatorial algorithms for market equilibria. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10, pages 103–134. Cambridge University Press, 2007.
33. D. Weller. Fair division of a measurable space. *Journal of Mathematical Economics*, 14(1):5–17, 1985.
34. O. Yilmaz. Random assignment under weak preferences. *Games and Economic Behavior*, 66(1):546–558, 2009.
35. H. P. Young. *Equity: in Theory and Practice*. Princeton University Press, 1994.
36. L. Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135, 1990.
37. R. Zivan, M. Dudík, S. Okamoto, and K. Sycara. Reducing untruthful manipulation in envy-free pareto optimal resource allocation. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 391–398, 2010.



## Proof of Theorem 5

*Proof.* Consider the following two-agent profile.

Agent 1:

$$v_1(x) = v_a^1 \text{ for } x \in [0, 0.5], \quad v_1(x) = v_b^1 \text{ for } x \in (0.5, 1].$$

Agent 2:

$$v_2(x) = v_a^2 \text{ for } x \in [0, 0.5], \quad v_2(x) = v_b^2 \text{ for } x \in (0.5, 1].$$

Choose  $v_a^1, v_b^1, v_a^2, v_b^2 > 0$  in such a way that  $v_a^1 > v_b^1$  and  $v_a^2 > v_b^2$  and  $\frac{v_a^1}{v_b^1} > \frac{v_a^2}{v_b^2}$ . By either robust envy-freeness or robust proportionality, the mechanism must make the following allocation  $x$  where  $x_a^1 = x_b^1 = x_a^2 = x_b^2 = 0.25$ . On the other hand, in order for the mechanism to be Pareto efficient,  $x$  must be an element of  $P^1 = \{x | x_b^1 = 0 \text{ or } x_a^2 = 0\}$ . Hence, we have reached an impossibility.  $\square$

## Proof of Theorem 6

*Proof.* For cake cutting with piecewise constant valuations and  $n \geq 2$ , it follows from (Theorem 3, [26]) that the only type of strategyproof and Pareto optimal mechanisms are dictatorships. Consequently, there exists no strategyproof and Pareto optimal mechanism that is also proportional or symmetric.  $\square$

## Proof of Theorem 7

*Proof.* The following example shows that there exists no cake-cutting algorithm that is strategyproof, robust proportional, and non-wasteful.

Profile 1:

$$1 : v_1(x) = a \text{ if } x \in [0, 0.5], \quad v_1(x) = b \text{ if } x \in [0.5, 1]$$

$$2 : v_2(x) = a \text{ if } x \in [0, 0.5], \quad v_2(x) = b \text{ if } x \in [0.5, 1]$$

for some  $a > b > 0$ .

Since the algorithm is robust proportional, it must be the case that each agent receive  $1/2$  of  $[0, 0.5]$  and  $1/2$  of  $[0.5, 1]$ . This is because the fractional parts of  $[0, 0.5]$  and  $[0.5, 1]$  that each agent receives must stochastically dominate the uniform allocation, otherwise the allocation would not satisfy proportionality for some valuation function in the ordinal equivalence class.

Without loss of generality (up to reordering and regrouping of the cake), we may assume that agent 1 receives  $[0, 0.25] \cup [0.5, 0.75]$  and agent 2 receives  $[0.25, 0.5] \cup [0.75, 1]$ .

Now consider profile 2:

$$1 : v_1(x) = a \text{ if } x \in [0, 0.25], v_1(x) = b \text{ if } x \in [0.5, 0.75], v_1(x) = 0 \text{ otherwise}$$

$$2 : v_2(x) = a \text{ if } x \in [0, 0.5], v_2(x) = b \text{ if } x \in [0.5, 1]$$

By SP, agent 1 must again receive  $[0, 0.25] \cup [0.5, 0.75]$  and agent 2 receives  $[0.25, 0.5] \cup [0.75, 1]$ . If agent 1 receives anything less in profile 2, then he would deviate from profile 2 to profile 1. If agent 1 receives anything more in profile 2, then he would deviate from profile 1 to profile 2.

Now consider profile 3:

$$1 : v_1(x) = a \text{ if } x \in [0, 0.25], v_1(x) = b \text{ if } x \in [0.5, 0.75], v_1(x) = 0 \text{ otherwise}$$

$$2 : v_2(x) = a + \epsilon \text{ if } x \in [0, 0.25], v_2(x) = a \text{ if } x \in [0.25, 0.5], v_2(x) = b \text{ if } x \in [0.5, 1]$$

By robust proportionality, both agent 1 and 2 must receive  $1/2$  of  $[0, 0.25]$ . By non-wastefulness, agent 2 must receive  $[0.25, 0.5]$  and  $[0.75, 1]$  since agent 1 has a utility of 0 on these intervals. Hence, agent 2 in profile 2 would misreport so that he receives the allocation in profile 3.

□

## Proof of Proposition 1

*Proof (Sketch).* First of all, CCEA is non-wasteful because an agent is never allowed to eat a piece of the cake that he has no desire for. On the other hand, the algorithm terminates only when every portion of the cake that is desired by at least one agent is completely consumed by some agent who desires it.

Next, we show that the algorithm is robust envy-free. Consider a fractional assignment  $p$  returned by the CC algorithm. Without private endowments CC is equivalent to the EPS algorithm of Katta and Sethuraman [18]. Assignment  $p$  satisfies justified envy-freeness in presence of variable eating rates:  $u_i(p_i) \geq u_i(p_j)$  for all utilities  $u$  consistent with preferences of  $i$  over the houses. The intuition is that at any point during the running of CC, an agent  $i$  will be ‘eating’ his most favoured object(s) at the same rate as any other agent  $j$  even if  $j$  is also the eating the same object(s). Hence, for all  $v'_i \in \hat{V}_i$ , it is the case that for  $j \neq i$ ,  $\int_{X_i} v'_i(x) dx \geq \int_{X_j} v'_i(x) dx$ .

□

## Proof of Proposition 2

*Proof (Sketch).* For a cake cutting instance  $I$ ,  $|I| = nm$  is the input size where  $n$  is the number of agents and  $m$  is the number of relevant subintervals. Once the lengths of the subintervals in  $\{J_1, \dots, J_m\}$  are computed, the size of each house can be computed in

linear time. The number of houses in CCEA is  $m$ . We now analyse the running time of CC on  $n$  agents and  $m$  houses (Section 3.5, [1]). In the CC algorithm, the flow network consists of  $|V|$  vertices and  $|E|$  arcs where  $|V| = O(n^2)$  and  $|E| = O(n^2m)$ . The number of parametric flow problems needed to be solved is  $O(nm)$ . A parametric flow network problem can be solved in time  $O(|V||E| \log(|V|^2/|E|))$  due to Gallo et al. [16]. Hence, the running time of CCEA is  $O(nm(n^2)(n^2m) \log(n^4/n^2m)) = O(n^5m^2 \log(n^2/m))$ .  $\square$

### Proof of Proposition 3

*Proof (Sketch).* CCEA is not strategyproof even if all the piecewise intervals are of equal length, and there are no private endowments or variable claims, and agents have strict preferences over the intervals. In this case CCEA is equivalent to the classic PS algorithm. It is known that PS is not strategyproof even for strict preferences when there are more objects than agents [19].  $\square$

### Proof of Lemma 1

*Proof (Sketch).* In the absence of private endowments and variable claims, CCEA can be solved by invoking EPS instead of CC but with the slight modification that in the corresponding flow network of EPS, the capacity of each arc  $(h, t)$  is set to  $size(h)$  in step 2 of EPS (Algorithm 1, [18]). Let us refer to this simplified CCEA as SimpleCCEA. When SimpleCCEA is run, it invokes EPS and solves repeated parametric network flow problems (Step 3, Algorithm 1, [18]). In the step, EPS computes a bottleneck set of agents and houses at each break-point. SimpleCCEA computes bottleneck sets in the same way as Mechanism 1 of Chen et al. [11] and then allocates the resources in the bottleneck sets to the agents in the bottleneck set. The flow networks of the slightly modified EPS (Figure 2, [18]) and that of Mechanism 1 (Figure 2, [11]) are identical with only two insignificant differences namely that in the flow network of Mechanism 1 of Chen et al. [11] i) the source and target are swapped and all the arcs are inverted and ii) the size of the houses/intervals is not normalized. However, the eventual allocations are same.  $\square$

### Proof of Proposition 4

*Proof.* In light of lemma 1, it suffices to show that SimpleCCEA is GSP. We begin with some notations.

Let  $len(X)$  denote the length of  $X \subseteq [0, 1]$ . Since the utility function is piecewise uniform, it suffices to consider the length of pieces of the cake that are desired by each agent.

Let  $S \subseteq N$  be a coalition of agents who misreport their true preference.

Let  $I$  denote the instance where every agent reports truthfully and  $I'$  denote the instance where agents in  $S$  misreport.

Let  $D_1, \dots, D_n \subseteq [0, 1]$  denote the pieces of cake that are truly desired by each agent.

Let  $D'_1, \dots, D'_n \subseteq [0, 1]$  denote the desired pieces of cake that are reported by each

agent.

Let  $A_1, \dots, A_n \subseteq [0, 1]$  denote the allocation received by each agent under truthful reports.

Let  $A'_1, \dots, A'_n \subseteq [0, 1]$  denote the allocation received by each agent when the agents in  $S$  misreport.

Let  $X_1, \dots, X_k$  be the bottleneck sets of agents with respect to the true preferences of agents arranged in the order that they are being allocated by the algorithm in instance  $I$ .

Let  $X'_1, \dots, X'_p$  be the bottleneck sets of agents with respect to the true preferences of agents arranged in the order that they are being allocated by the algorithm in instance  $I'$ .

Let  $\text{len}(X_i)$  denote the length of the allocation each agent receives in the bottleneck set  $X_i$ . Let

$$\tilde{X}_l = \{i \in X_l \mid \text{len}(A'_i \cap D_i) \geq \text{len}(A_i \cap D_i) = \text{len}(A_i)\}^{12}$$

$$\hat{X}_l = \{i \in X_l \mid \text{len}(A'_i \cap D_i) \leq \text{len}(A_i)\}$$

In other words  $\tilde{X}_l$  is the subset of agents of  $X_l$  who weakly gain in utility when the agents in  $S$  misreport, and  $\hat{X}_l$  is the subset of agents of  $X_l$  who weakly lose in utility when the agents in  $S$  misreport. We will show that for all  $l = 1, \dots, k$ ,  $X_l = \hat{X}_l$ . This would then directly imply that  $S$  must be empty since no coalition  $S$  can exist such that by misreporting, everyone in the coalition is weakly better off and at least one agent in the coalition is strictly better off.

We will prove this result via induction on  $k$ . In order to carry on with the induction, we will show that no agent in  $X_1$  appears in the coalition  $S$ . We begin with a lemma.

**Lemma 2.** *It is the case that  $X_1 = \tilde{X}_1$ . In other words, no agent in  $X_1$  is strictly worse off when some subset of agents misreport their preference.*

It is clear that  $\tilde{X}_1 \subseteq X_1$ . Suppose the claim does not hold, then there exists some agent  $i \in X_1 \setminus \tilde{X}_1$ . In other words, agent  $i$  is strictly worse off due to the misreports. Since an agent would only misreport his preference if misreporting weakly improves his utility, we deduce that  $i \notin S$ , which implies that  $D_i = D'_i$ . Consequently, the following set of inequalities hold for agent  $i$ :

$$\text{len}(A'_i) = \text{len}(A'_i \cap D'_i) = \text{len}(A'_i \cap D_i) < \text{len}(A_i),$$

where the first equality follows from the fact that  $A'_i \subseteq D'_i$  by the way the algorithm allocates the pieces of cake to agents. The second equality follows from  $D_i = D'_i$ . And the last inequality follows from  $i \in X_1 \setminus \tilde{X}_1$ .

We claim that since  $X_1$  is the first bottleneck set with respect to the true preference, it cannot be the case that  $\text{len}(X'_1) = \text{len}(A'_i) < \text{len}(A_i) = \text{len}(X_1)$ , where  $X'_1$  is the bottleneck set that  $i$  belongs to in the instance  $I'$ . Suppose this is the case, then we have

<sup>12</sup> The fact that  $\text{len}(A_i \cap D_i) = \text{len}(A_i)$  makes use of the free disposal property, i.e. the allocation that the mechanism gives agent  $i$  is a subset of the pieces desired by agent  $i$  under truthful reports.

that  $\text{len}(X'_1) \leq \text{len}(X'_I) < \text{len}(X_1)$ .<sup>13</sup> It is clear that  $X'_1$  cannot contain an agent who misreports in  $I'$ , since a misreporting agent in  $X'_1$  only receives a piece of cake with length  $\text{len}(X'_I) < \text{len}(X_1)$ , which is strictly less than what he would've gotten had he reported truthfully. Hence, every agent in  $X'_1$  must report his true preference in  $I'$ . Since  $X'_1$  is the first bottleneck set in  $I'$ , we have that

$$\begin{aligned} \text{len}(X'_1) &= \frac{\text{len}((\cup_{j \in X'_1} D'_j) \cap [0, 1])}{|X'_1|} = \frac{\text{len}((\cup_{j \in X'_1} D_j) \cap [0, 1])}{|X'_1|} \\ &\geq \frac{\text{len}((\cup_{j \in X_1} D_j) \cap [0, 1])}{|X_1|} = \text{len}(X_1), \end{aligned}$$

where the first and third equalities follows from the way the algorithm makes an allocation, the second equality follows from the fact that the agents in  $X'_1$  have the same reports in  $I'$  as in  $I$ , and the inequality follows from the fact that  $X_1 \in \arg \min_{S \subseteq N} \frac{\text{len}((\cup_{j \in S} D_j) \cap [0, 1])}{|S|}$ . This contradicts the fact that  $\text{len}(X'_1) < \text{len}(X_1)$ .

**Lemma 3.** *It is the case that  $X_1 = \hat{X}_1 = \tilde{X}_1$ . In other words, no agent in  $X_1$  is strictly better off when some subset of agents misreport their preference.*

Since we have established that  $X_1 = \tilde{X}_1$ , and  $\hat{X}_1 \subseteq X_1$ , it suffices to show that  $\tilde{X}_1 \subseteq \hat{X}_1$ . Suppose not, then for all  $i \in X_1$ , we have that  $\text{len}(A'_i \cap D_i) \geq \text{len}(A_i)$  and there exists some  $j \in X_1$  such that  $\text{len}(A'_j \cap D_j) > \text{len}(A_j)$ . Summing over  $i \in X_1$ , we get that

$$\text{len}(\cup_{i \in X_1} (A'_i \cap D_i)) = \sum_{i \in X_1} \text{len}(A'_i \cap D_i) > \sum_{i \in X_1} \text{len}(A_i) = \text{len}(\cup_{i \in X_1} A_i) = \text{len}(\cup_{i \in X_1} D_i),$$

where the first two equalities follow from the fact that the  $A_i$ 's and  $A'_i \cap D_i$ 's are disjoint subsets and the third equality follows from the way the algorithm allocates to the agents in the smallest bottleneck set. But this set of inequalities contradict the fact that  $\cup_{i \in X_1} (A'_i \cap D_i) \subseteq \cup_{i \in X_1} D_i$ , which implies that  $\text{len}(\cup_{i \in X_1} (A'_i \cap D_i)) \leq \text{len}(\cup_{i \in X_1} D_i)$ . Hence, it must be the case that for every  $i \in X_1$ , we have that  $\text{len}(A'_i \cap D_i) = \text{len}(A_i)$ , which implies that  $i \in \hat{X}_1$ .

**Lemma 4.** *No agent in  $X_1$  appears in the coalition  $S$  and  $X_1$  is also the first bottleneck set for  $I'$ .*

By the previous lemma, no agent in  $\hat{X}_1$  is strictly better off by misreporting his preference. Thus, any agent in  $X_1$  would potentially be in  $S$  if by misreporting, he makes himself no worse off and simultaneously make some other agent in  $S$  strictly better off. The only way that this can happen is by misreporting, the agents in  $X_1$  make their collective claim over their desired pieces smaller, so that agents in later bottleneck set can claim some of their desired pieces. On the other hand, the following inequality

$$\text{len}(\cup_{i \in X_1} D'_i) \geq \text{len}(\cup_{i \in X_1} A'_i) \geq \text{len}(\cup_{i \in X_1} (A'_i \cap D_i)) = \text{len}(\cup_{i \in X_1} D_i)$$

<sup>13</sup> It is shown in [12] that the length of allocation of agents weakly increases with respect to the index of bottleneck sets.

implies that if any subset of agents of  $X_1$  wants to misreport so they are not worse off, then collectively, they must over-report their preference to obtain allocations that together is weakly larger in total than the allocations they would get had they reported truthfully. Thus, having a subset of agents in  $X_1$  misreport will not benefit the other agents in the coalition  $S$ . Hence, we may conclude that no agent in  $X_1$  appears in the coalition  $S$ . Provided that every agent in  $X_1$  also reports truthfully in  $I'$ , there is no incentive for an agent that belongs to a subsequent bottleneck set in  $I$  to misreport and prevent  $X_1$  from being the first bottleneck set in  $I'$  since that would make the misreporting agent strictly worse off, as in doing so, he needs to create a bottleneck set  $X'$  such that  $\text{len}(X') < \text{len}(X)$  and he would consequently receive an allocation of  $\text{len}(X')$ .

Since no agent in  $X_1$  appears in the coalition  $S$  and  $X_1$  is also the first bottleneck set for  $I'$ , we can remove  $X_1$  from  $N$  and  $\cup_{i \in X_1} A_i$  from  $[0, 1]$  and do induction on the set of remaining agents  $N \setminus X_1$  and the remaining piece of cake  $[0, 1] \setminus \cup_{i \in X_1} A_i$  to be allocated and the proof is complete by invoking the inductive hypothesis with  $X_2$  being the first bottleneck set in the new instance.  $\square$

## Proof of Proposition 6

*Proof.* Consider the following math program

$$\begin{aligned}
& \max \quad \prod_{j=1}^n u_j \\
& \text{s.t.} \quad u_j = \sum_{i=1}^k v_{ij} x_{ij} \quad \forall j = 1, \dots, n \\
& \quad \sum_{j=1}^n x_{ij} \leq l_i \quad \forall i = 1, \dots, k \\
& \quad x_{ij} \geq 0 \quad \forall i, j.
\end{aligned}$$

where  $l_i = \text{len}(J_i)$ ,

Notice that the feasible region of the math program contains all feasible allocations. An optimal solution given by the LP is not Pareto dominated by any other feasible allocation because that would contradict the optimality of the solution. Hence, it is Pareto efficient.

To see that the optimal solution of the math program is also an envy free allocation, if we instead view  $x_{ij}$  as the fractional amount of  $J_i$  that is allocated to agent  $j$ , then scaling the  $v_{ij}$ 's appropriately (i.e. setting  $v'_{ij} = v_{ij} l_i$ ), then solving the math program

stated in the proposition is equivalent to solving the following math program.

$$\begin{aligned}
& \max \quad \prod_{j=1}^n u_j \\
& \text{s.t.} \quad u_j = \sum_{i=1}^k v'_{ij} x_{ij} \quad \forall j = 1, \dots, n \\
& \quad \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i = 1, \dots, k \\
& \quad x_{ij} \geq 0 \quad \forall i, j.
\end{aligned}$$

which in turn equivalent to solving

$$\begin{aligned}
& \max \quad \sum_{j=1}^n \log u_j \\
& \text{s.t.} \quad u_j = \sum_{i=1}^k v'_{ij} x_{ij} \quad \forall j = 1, \dots, n \\
& \quad \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i = 1, \dots, k \\
& \quad x_{ij} \geq 0 \quad \forall i, j.
\end{aligned}$$

Notice that the above math program is a convex program since we are maximizing a concave function (or equivalently minimizing a convex function) subject to linear constraints.

In (pp 105-107, [32]), Vazirani invites us to consider a market setting of buyers (agents) and divisible goods (intervals). Each good is assumed to be desired by at least one buyer (i.e. for every good  $i$ ,  $v_{ij} > 0$  for some buyer  $j$ ). There is a unit of each good and each buyer is given the same amount of money say 1 dollar, for which he uses to purchases the good(s) that maximizes his utility subject to a set of given prices. The task is to find a set of equilibrium prices such that the market clears (meaning all the demands are met and no part of any item is leftover) when the buyers seek purchase good(s) to maximize their utility given the equilibrium prices.

Using duality theory, one can interpret the dual variable  $p_i$  associated with the constraints  $\sum_{j=1}^n x_{ij} \leq 1$  as the price of consuming a unit of good  $i$ . By invoking the KKT conditions, Vazirani [32] shows the prices given by the optimal dual solution is a unique set of equilibrium prices. Moreover, the primal optimal solution for each buyer  $j$  is precisely the quantity of good(s) that the buyer ends up purchasing that maximizes

his utility given the equilibrium prices.

Now we can argue as to why the optimal primal solution is an envy free allocation. Because given the equilibrium prices, if a buyer desires another buyer's allocation, since he has the same purchasing power as any other buyer, he would instead use his money to obtain the allocation of the buyer that he envies. This would result in some surplus and deficit of goods, contradicting the fact that the given prices are equilibrium prices.  $\square$

## Proof of Proposition 7

*Proof.* For piecewise uniform valuations, it is known that CCEA is equivalent to mechanism 1 in [12], which we will refer to as *SimpleCCEA*. The remainder of the proof will focus on showing the equivalence between SimpleCCEA and the convex program for piecewise uniform valuations. To do so, given an allocation of SimpleCCEA, which is a feasible solution of the convex program, we will find a set of prices corresponding to the allocation and show that the prices are in fact the equilibrium prices defined by Vazirani on pages 105-107 of [32]. Moreover, this allocation would be an allocation that maximizes the agents' utility given the equilibrium prices.

Using the same notations as those in [12], given a valuation profile, let  $B$  be the set of buyers/agents and  $G$  be the set of goods or intervals. Let  $S_i$  be the  $i$ -th bottleneck set computed by SimpleCCEA, i.e.  $S_i = S_{\min} \in \arg \min_{S' \subseteq S} \text{avg}(S', X_i)$  in the  $i$ -th iteration of the subroutine of SimpleCCEA.

Let  $G_i$  be the set of goods that are distributed amongst the buyers in  $S_i$ . In the convex program, since each buyer is endowed with 1 dollar and every buyer in  $S_i$  receives  $\text{avg}(S_i, X_i)$  units of good(s), it is natural to define the price of a unit of each good  $k \in G_i$  to be

$$p_k = \frac{1}{\text{avg}(S_i, X_i)} = \frac{|S_i|}{\text{len}(D(S_i, X_i))}.$$

Notice that the prices for each good is well defined (i.e. each good has exactly one nonnegative price). This follows from the following observations:

- (i)  $\cup G_i = G$  (or every good has at least one price). This follows from the assumption that every good is desired by at least one agent, which means that SimpleCCEA will allocate all of the goods.
- (ii)  $G_i \cap G_j = \emptyset$  for all  $i \neq j$  (or every good has at most one price). This follows from the fact that no fractional parts of any good is allocated to agents from two or more bottleneck sets, which another algorithmic property of SimpleCCEA.

To show that the  $p_k$ 's form a set of equilibrium prices, we will show that given the  $p_k$ 's, the buyers in every  $S_i$  will choose to purchase *only* goods from  $G_i$  to maximize their utility function. We will do induction on the number of bottleneck sets. Consider the first bottleneck set  $S_1$ , we will show that

**Lemma 5.**  $G_1$  are the only items that are desirable by buyers in  $S_1$ .



*Proof.* SimpleCCEA finds  $S_1$  by solving a parametrized max flow problem on a bipartite network. The network has a node for every buyer and a node for every good. There is a directed edge from a buyer  $i$  to a good  $j$  with infinite capacity if good  $j$  is desired by buyer  $i$ . In addition, there is a source  $s$  and a sink  $t$ . There is a directed edge from  $s$  to each buyer  $i$  with capacity  $\lambda$  and a directed edge from each good  $j$  to  $t$  with capacity equaling the quantity of good  $j$ .  $\lambda$  is set to 0 initially so that the unique min cut in the network for  $\lambda = 0$  is  $\{s, B \cup G \cup t\}$ .  $\lambda$  is gradually raised until  $\{s, B \cup G \cup t\}$  is no longer the unique min cut, at which point  $S_1$  and  $G_1$  are found by looking for another min cut of the form  $\{s \cup S_1 \cup G_1, B \setminus S_1 \cup G \setminus G_1 \cup t\}$ . Since  $\{s \cup S_1 \cup G_1, B \setminus S_1 \cup G \setminus G_1 \cup t\}$  is a min cut, it must be the case that  $G_1 = nbr(S_1)$  in the network, which proves the lemma. (If  $nbr(S_1) \setminus G_1 \neq \emptyset$ , then there will be an infinite capacity edge crossing the cut. On the other hand, if  $G_1 \setminus nbr(S_1) \neq \emptyset$ , then replacing  $G_1$  with  $nbr(S_1)$  on the  $s$  side of the cut would give a cut with a smaller capacity.)  $\square$

Since the agents have piecewise uniform valuation, in the setting of the convex program, each buyer in  $S_1$  has the same utility for the items that he desire in  $G_1$ . Moreover, given that the prices of goods are identical in  $G_1$ , each buyer is indifferent between choosing among his desirable items for the best item in terms of bang per buck. Hence, for all buyers in  $S_1$ , the allocation given by SimpleCCEA maximizes buyer's utility given the prices  $p_k$ 's. Moreover, notice that the money of buyers in  $S_1$  and goods in  $G_1$  are exhausted by the allocation given by SimpleCCEA.

After the goods in  $G_1$  are allocated to the buyers in  $S_1$  we repeat the same argument for the remaining buyers and goods and the inductive hypothesis allows to conclude that for every  $i$  and for all buyers in  $S_i$ , the allocation given by SimpleCCEA maximizes buyer's utility given the prices  $p_k$ 's. Moreover, the money of buyers in  $S_i$  and goods in  $G_i$  are exhausted by the allocation given by SimpleCCEA. There is a slight difference between the inductive step and the base case, as it is possible that some buyer  $b$  in  $B \setminus S_i$  also desires certain goods in  $G_j$  for some  $j < i$ . However, Chen et al. [12] state that  $avg(S_i, X_i)$  is a weakly increasing function of  $i$ , which means that  $p_i = 1/avg(S_i, X_i)$  is a weakly decreasing function of  $i$ . Since we are dealing with piecewise uniform valuations, the utility of a buyer over his desirable goods are identical, this means that for any buyer in  $S_i$ , the goods that maximize his bang for buck are in  $G_i$ .

Putting everything together and we have shown that  $p_k$ 's constitute the set of equilibrium prices and SimpleCCEA gives an equilibrium allocation, which is an optimal solution to the convex program.  $\square$

## Proof of Proposition 8

Consider the following profile of two agents.

Profile 1:

$$\begin{aligned} v_1(x) &= 1 \text{ if } x \in [0, 0.2], \quad v_1(x) = 0 \text{ if } x \in (0.2, 1]. \\ v_2(x) &= 0 \text{ if } x \in [0, 0.6], \quad v_2(x) = 1 \text{ if } x \in (0.6, 1]. \end{aligned}$$

The uniform allocation rule gives us the allocation:

$$Y_1 : \{\frac{1}{2}[0, 0.2], \frac{1}{2}(0.6, 1]\}.$$

$$Y_2 : \{\frac{1}{2}[0, 0.2], \frac{1}{2}(0.6, 1]\}.$$

Let  $A \subset (0.6, 1]$  be the allocation that agent 2 receives in this case.

Now consider profile 2:

$$\begin{aligned} v_1(x) &= 1 \text{ if } x \in [0, 0.2], \quad v_1(x) = 0 \text{ if } x \in (0.2, 1]. \\ v_2(x) &= 0 \text{ if } x \in [0, 0.1] \setminus A, \quad v_2(x) = 1 \text{ if } x \in A. \end{aligned}$$

The uniform allocation rule gives us the allocation:

$$Y_1 : \{\frac{1}{2}[0, 0.2], \frac{1}{2}A\}.$$

$$Y_2 : \{\frac{1}{2}[0, 0.2], \frac{1}{2}A\}.$$

Hence, agent 2 in profile 2 would misreport so that the reported profile is profile 1.

## Proof of Proposition 9

*Proof.* We first prove that CSD is well-defined and results in a feasible allocation in which each agent gets  $1/n$  size of the cake. Let  $\mathcal{J}' = \{J_1, \dots, J_\ell\}$  be a partitioning of the interval  $[0, 1]$  induced by the discontinuities in agent valuations and the cake cuts in the  $n!$  cake allocations. We make a couple of claims about  $\mathcal{J}'$  that following from the way  $\mathcal{J}'$  is constructed.

*Claim.* An agent is completely indifferent over each subinterval in  $\mathcal{J}'$ .

*Claim.* Let  $X_i^\pi$  denote a maximum preference cake piece of size  $1/n$  chosen by agent  $i$  in the serial order  $\pi$ . For each  $J \in \mathcal{J}'$  either  $X_i^\pi$  contains  $J$  completely or it does not contain any part of  $J$ .

Now consider a matrix of dimension  $n! \times \ell$ :  $B = (b_{ij})$  such that  $b_{ij} = 1$  if  $J_j \subset X_i^\pi$  and  $b_{ij} = 0$  if  $J_j \not\subset X_i^\pi$ . Since for each  $\pi \in \Pi^N$ , each agent  $i \in N$  gets  $1/n$  of the cake in  $X_i^\pi$ , then it follows that  $\sum_{i=1}^{n!} \sum_{j=1}^{\ell} b_{ij} \text{len}(J_j) = n!/n$ . Hence,

$$\sum_{j=1}^{\ell} \sum_{i=1}^{n!} b_{ij} \text{len}(J_j) / n! = 1/n.$$

Also consider a matrix of dimension  $n \times \ell$ :  $M = (m_{ij})$  such that  $m_{ij}$  denotes the fraction of  $J_j$  that agent  $i$  gets in  $Y_i$ . From the algorithm CSD, we know that  $m_{ij} = \frac{\text{count}(i, J_j)}{n!} J$  where  $\text{count}(i, J_j)$  is the number of permutations in which  $i$  gets  $J_j$ . It is immediately seen that each column sums up to 1. Hence each  $J_j$  is complete allocated

to the agents. We now prove that each agent gets a total cake piece of size  $1/n$ . We do so by showing that  $\sum_{j=1}^{\ell} m_{ij} \text{len}(J_j) = 1/n$ .

$$\begin{aligned} 1/n &= \sum_{j=1}^{\ell} \sum_{i=1}^{n!} b_{ij} \text{len}(J_j)/n! = \sum_{j=1}^{\ell} \left( \sum_{i=1}^{n!} b_{ij} \right) \text{len}(J_j)/n! = \sum_{j=1}^{\ell} (\text{count}(i, J_j)) \text{len}(J_j)/n! \\ &= \sum_{j=1}^{\ell} \left( \frac{\text{count}(i, J_j)}{n!} \right) \text{len}(J_j) = \sum_{j=1}^{\ell} m_{ij} \text{len}(J_j). \end{aligned}$$

Hence  $Y = (Y_1, \dots, Y_n)$  the allocation returned by CSD is a proper allocation of the cake in which each agent gets a total cake piece of size  $1/n$ .  $\square$

## Proof of Proposition 10

*Proof.* We first argue for proportionality of CSD. In the case where all agents have the same valuations as the valuation of  $i$ ,  $i$  is guaranteed  $1/n$  of the value of the whole cake because of anonymity of CSD. First note that for each  $\pi \in \Pi^N$  and preferences  $V_{-i}$  of all agents other than  $i$ ,  $V_i(\text{CSD}^\pi(V_i, V_{-i})) \geq V_i(\text{CSD}^\pi(V_i, (V_i, \dots, V_i)))$ .

The reason is that when valuations are not identical, predecessors of  $i$  in  $\pi$  leave weakly better cake for  $i$  as when their valuations are same as agent  $i$ . Hence,  $V_i(\text{CSD}(V_i, V_{-i})) \geq V_i(\text{CSD}(V_i, (V_i, \dots, V_i))) = V_i([0, 1])/n$ .

Finally, note that when an agent selects his best possible cake piece in each permutation, the exact height of the valuation function is not relevant and only the relative height matters. Hence, CSD in fact satisfies robust proportionality. Symmetry for CSD follows directly from symmetry for RSD.  $\square$

## Proof of Remark 2

CSD would not be strategyproof if the fraction of each subinterval of  $\mathcal{J}'$  is allocated deterministically and the allocation is made public information before the agents submit their valuation function. To see this, consider the following example of two agents.

Profile 1:

$$\begin{aligned} v_1(x) &= 1 \text{ if } x \in [0, 0.5], \quad v_1(x) = 0 \text{ if } x \in (0.5, 1]. \\ v_2(x) &= 1 \text{ if } x \in [0, 0.5], \quad v_2(x) = 0 \text{ if } x \in (0.5, 1]. \end{aligned}$$

Running CSD gives us the allocation:

$$Y_1 : \{\tfrac{1}{2}[0, 0.5], \tfrac{1}{2}(0.5, 1]\}.$$

$$Y_2 : \{\tfrac{1}{2}[0, 0.5], \tfrac{1}{2}(0.5, 1]\}.$$

Profile 2:

$$\begin{aligned} v_1(x) &= 1 \text{ if } x \in [0, 0.5], \quad v_1(x) = 0 \text{ if } x \in (0.5, 1]. \\ v_2(x) &= 0 \text{ if } x \in [0, 0.25], \quad v_2(x) = 1 \text{ if } x \in (0.25, 0.75], \quad v_2(x) = 0 \text{ if } x \in (0.75, 1]. \end{aligned}$$

Running CSD gives us the allocation:

$$Y_1 = \{[0, 0.25], \frac{1}{2}(0.25, 0.5], \frac{1}{2}(0.75, 1]\}.$$

$$Y_2 = \{\frac{1}{2}[0.25, 0.5], (0.5, 0.75], \frac{1}{2}(0.75, 1]\}.$$

Now it is possible that the CSD mechanism decides to give  $[0, 0.25] \cup (0.75, 1]$  to agent 1 and  $[0.25, 0.75]$  to agent 2 in profile 1. Consequently, knowing this, agent 2 in profile 2 would misreport his valuation function to be  $v_2(x) = 1$  if  $x \in [0, 0.5]$ ,  $v_2(x) = 0$  if  $x \in (0.5, 1]$  in order to receive the allocation given in profile 1 and gain utility in doing so.

## Proof of Proposition 11

Consider the profiles  $P$  and  $P^i$ , where  $P$  is a profile where every agent reports truthfully and  $P^i$  is a profile where agent  $i$  misreports while fixing every other agent's report to be the same as that in  $P$ . Let  $\sigma$  denote a permutation of  $[n] = \{1, \dots, n\}$  and let  $S$  denote the set of all permutations of  $[n]$ . Let  $J_1, \dots, J_k$  denote the intervals whose fractional allocations are specified to each agent by CSD in profile  $P$  and  $J'_1, \dots, J'_{k'}$  denote the intervals whose fractional allocations are specified to each agent by CSD in profile  $P^i$ . Let  $V_i(J)$  denote agent  $i$ 's total utility derived from receiving the interval  $J$  and  $V_i(A(\sigma))$  and  $V_i(A'(\sigma))$  denote the agent  $i$ 's total utility derived from his allocated pieces when the serial ordering of the agents is  $\sigma$  in profile  $P$  and  $P^i$  respectively. Let  $p_{ij}$  denote the probability that interval  $J_j$  is assigned to agent  $i$ . Since random serial dictatorship is strategyproof in expectation, we have that

$$\sum_{j=1}^k p_{ij} V_i(J_j) = \sum_{\sigma \in S} \frac{1}{n!} V_i(A(\sigma)) \geq \sum_{\sigma \in S} \frac{1}{n!} V_i(A'(\sigma)) = \sum_{j=1}^{k'} p_{ij} V_i(J'_j)$$

Now CSD views  $p_{ij}$  as allocating a  $p_{ij}$  fraction of interval  $J_j$  to agent  $i$ . In order for a deviating agent to properly evaluate the utility derived from his allocation in the deviating profile  $P^i$ , we have to come up with an allocation rule that actually *attains* the utility  $\sum_{j=1}^{k'} p_{ij} V_i(J'_j)$  for agent  $i$  (either deterministically or in expectation) when the profile of reports is  $P^i$ . In particular, say if we want to allocate a subinterval of  $J_i$  with length  $p_{ij}$  times that of  $J_i$  to agent  $i$  at random, then this random allocation rule must satisfy the property that  $\sum_{j=1}^{k'} E[V_i(A_{ij})] = p_{ij} V_i(J'_j)$ , where  $A_{ij} \subset J'_j$  such that  $|A_{ij}| = p_{ij}|J'_j|$ . In order to do so, we will show that the randomized allocation rule for CSD satisfy the property that every agent  $i$  and interval  $J_j$ , we have that  $E[V_i(A_{ij})] = p_{ij} V_i(J'_j)$  for all valuation functions  $V_i$ , where  $A_{ij} \subset J'_j$  such that  $|A_{ij}| = p_{ij}|J'_j|$ .

Notice that  $\text{mod}(U_j + \sum_{k=1}^{i-1} p_{nj}(b_j - a_j))$  is uniformly distributed on  $[a_j, b_j]$  and  $A_{ij}$  has length  $p_{ij}(b_j - a_j)$ . The fact that  $E[V_i(A_{ij})] = p_{ij} V_i(J'_j)$  follows from the following lemma.

**Lemma 6.** Let  $U$  be uniformly distributed on the interval  $[a, b]$  and let  $0 \leq \alpha \leq 1$ . Let  $A = [U, U + \alpha(b - a)]$  if  $U + \alpha(b - a) \leq b$  and  $A = [a, U - (1 - \alpha)(b - a)] \cup [U, b]$  if  $U + \alpha(b - a) > b$ , then we have that  $E_U[V_i(A)] = \alpha V_i([a, b])$ , where  $V_i(X) = \int_X v_i(x) dx$  for any integrable function  $v_i$ .

*Proof.* Define  $\bar{v}_i = v_i$  for  $a \leq x \leq b$  and  $\bar{v}_i(x) = \bar{v}_i(x + (b - a))$  for  $x$  outside  $[a, b]$ . Since  $\bar{v}_i$  is periodic, then it suffices to show that

$$E_U[V_i(A)] = \int_a^b \int_x^{x+\alpha(b-a)} \bar{v}_i(y) dy \frac{1}{b-a} dx = \alpha \int_a^b \bar{v}_i(x) dx = \alpha V_i([a, b]).$$

By drawing a picture of the region that we evaluate the integral over and due to the fact that  $\bar{v}_i$  is periodic, we have that

$$\frac{1}{b-a} \int_a^b \int_x^{x+\alpha(b-a)} \bar{v}_i(y) dy dx = \frac{1}{b-a} \int_a^b \bar{v}_i(y) \int_{y-\alpha(b-a)}^y dx dy = \alpha \int_a^b \bar{v}_i(x) dx,$$

which proves the lemma.  $\square$

## Proof of Proposition 12

*Proof.* Let  $a = [0, 0.25]$ ,  $b = (0.25, 0.5]$ ,  $c = (0.5, 0.75]$ ,  $d = (0.75, 1]$ . Consider the following two profiles of valuations.

Profile 1:

$$\begin{aligned} v_1(x) &= 4 \text{ if } x \in a, \quad v_1(x) = 3 \text{ if } x \in b, \quad v_1(x) = 2 \text{ if } x \in c, \quad v_1(x) = 1 \text{ if } x \in d. \\ v_2(x) &= 3 \text{ if } x \in a, \quad v_2(x) = 4 \text{ if } x \in b, \quad v_2(x) = 1 \text{ if } x \in c, \quad v_2(x) = 2 \text{ if } x \in d. \end{aligned}$$

Running CSD gives us the allocation:

$$Y_1 = \{\frac{1}{2}a, \frac{1}{2}b, \frac{1}{2}c, \frac{1}{2}d\}.$$

$$Y_2 = \{\frac{1}{2}a, \frac{1}{2}b, \frac{1}{2}c, \frac{1}{2}d\}.$$

Profile 2:

$$\begin{aligned} v_1(x) &= 4 \text{ if } x \in a, \quad v_1(x) = 2 \text{ if } x \in b, \quad v_1(x) = 3 \text{ if } x \in c, \quad v_1(x) = 1 \text{ if } x \in d. \\ v_2(x) &= 2 \text{ if } x \in a, \quad v_2(x) = 4 \text{ if } x \in b, \quad v_2(x) = 1 \text{ if } x \in c, \quad v_2(x) = 3 \text{ if } x \in d. \end{aligned}$$

Running CSD gives us the allocation:

$$Y_1 = \{a, c\}.$$

$$Y_2 = \{b, d\}.$$

Hence, agents with true valuation in profile 1 would misreport together to profile 2, which means that CSD is not group strategyproof for 2 agents.  $\square$

### Proof of Proposition 13

*Proof.* There are three agents, each with piecewise uniform valuation function.  
 $v_1(x) = 1.5$  for  $x \in [0, 2/3]$  and 0 otherwise.  
 $v_2(x) = 1.5$  for  $x \in [0, 1/3] \cup (2/3, 1]$  and 0 otherwise.  
 $v_3(x) = 1.5$  for  $x \in (1/3, 1]$  and 0 otherwise.

Let  $a = [0, 1/3]$ ,  $b = (1/3, 2/3]$ ,  $c = (2/3, 1]$ .

We adopt the following implementation of CSD: when it is agent  $i$ 's turn to pick, out of the pieces of the remaining cake that he likes, he takes the *left-most* such piece with length  $1/n$ , where  $n$  is the number of agents.

If the priority ordering were 1, 2, 3, then a feasible assignment that respects the preferences is  $1 \leftarrow a$ ,  $2 \leftarrow c$ ,  $3 \leftarrow b$ .

If the priority ordering were 1, 3, 2, then a feasible assignment that respects the preferences is  $1 \leftarrow a$ ,  $3 \leftarrow b$ ,  $2 \leftarrow c$ .

If the priority ordering were 2, 1, 3, then a feasible assignment that respects the preferences is  $2 \leftarrow a$ ,  $1 \leftarrow b$ ,  $3 \leftarrow c$ .

If the priority ordering were 2, 3, 1, then a feasible assignment that respects the preferences is  $2 \leftarrow a$ ,  $3 \leftarrow b$ ,  $1 \leftarrow c$ .

If the priority ordering were 3, 1, 2, then a feasible assignment that respects the preferences is  $3 \leftarrow b$ ,  $1 \leftarrow a$ ,  $2 \leftarrow c$ .

If the priority ordering were 3, 2, 1, then a feasible assignment that respects the preferences is  $3 \leftarrow b$ ,  $2 \leftarrow a$ ,  $1 \leftarrow c$ . Then, the CSD allocation is as follows.

$$\begin{aligned} Y_1 &= \left\{ \frac{1}{2}[0, 1/3], \frac{1}{6}(1/3, 2/3], \frac{1}{3}(2/3, 1] \right\} \\ Y_2 &= \left\{ \frac{1}{2}[0, 1/3], \frac{1}{2}(2/3, 1] \right\} \\ Y_3 &= \left\{ \frac{5}{6}(1/3, 2/3], \frac{1}{6}(2/3, 1] \right\} \end{aligned}$$

Clearly, agent 1 envies agent 3 in this case.  $\square$

### Proof of Proposition 15

*Proof (Sketch).* Consider a fractional assignment  $p$  returned by the CC algorithm. Without private endowments CC is equivalent to the EPS algorithm of Katta and Sethuraman [18]. Assignment  $p$  satisfies justified envy-freeness in presence of variable eating rates:  $u_i(p_i) \geq (c_i/c_j)u_i(p_j)$  for all utilities  $u$  consistent with preferences of  $i$  over the houses. The informal intuition is that at any point during the running of CC, an agent  $i$  with a higher eating rate than  $j$  will be 'eating' his most favoured object(s) faster than  $j$  even if  $j$  is also has the eating the same object(s). Hence, for all  $v'_i \in \hat{V}_i$ , it is the case that for  $j \neq i$ ,  $\int_{X_i} v'_i(x)dx \geq (c_i/c_j) \int_{X_j} v'_i(x)dx$ .  $\square$

## Proof of Proposition 16

*Proof (Sketch).* Given a variable claim instance  $I$  of  $n$  agents where agent  $i$  has claim rate  $c_i$  for  $i = 1, \dots, n$ . We may assume without loss of generality that the claim rates are integral. If they are not, then we can simply multiple each claim rate  $c_i$  by a common denominator to make each  $c_i$  integral. Doing so will not change the allocation given by the algorithm since only relative claim rates matter to the algorithm.

Now consider a cake cutting instance  $I'$  of  $\sum_{i=1}^n c_i$  agents, where the agents have piece-wise uniform utility function and there are no private endowments or variable claims. Moreover, for every  $i = 1, \dots, n$ , there are  $c_i$  agents in  $I'$  each of whom has the same utility function as that of agent  $i$  in  $I$ . It is not difficult to see that if one aggregates the allocation that the  $c_i$  agents in  $I'$  who share agent  $i$ 's valuation in  $I$ , then one would get an equivalent allocation (in terms of utility) to agent  $i$ 's allocation.

Suppose for the sake of contradiction that CCEA is not GSP for the case of variable claims, then in some instance  $I$ , there exists some coalition  $S$  of the agents that weakly gains in utility by misreporting their preference. Now consider the equivalent instance  $I'$  with no variable claims under the aforementioned transformation, then there exists some coalition  $S'$  of the agents in  $I'$  that weakly gains in utility by misreporting their preference, which implies that CCEA is not group-strategyproof for the no variable claims case, contradicting the result of Proposition 4.  $\square$

## Proof of Proposition 17

*Proof.* The allocation can be obtained by solving the following convex program.

$$\begin{aligned} \min \quad & - \sum_{j=1}^n c_j \log(u_j) \\ \text{s.t.} \quad & u_j = \sum_{i=1}^k v_{ij} x_{ij} \quad \forall j = 1, \dots, n \\ & \sum_{j=1}^n x_{ij} \leq l_i \quad \forall i = 1, \dots, k \\ & x_{ij} \geq 0 \quad \forall i, j. \end{aligned}$$

The proof of the desired properties is similar to the case where  $c_j = 1$ .  $\square$

## Proof of Proposition 18

*Proof (Sketch).* Consider a fractional assignment  $p$  returned by the CC algorithm. We know that  $p$  satisfies justified envy-freeness for the random/fractional assignment problem (Prop. 4, [1]). If  $p_i \succeq_i^{SD} p_j$ , then  $\forall v'_i \in \hat{V}_i: \int_{X_i} v'_i(x) dx \geq \int_{X_j} v'_i(x) dx$ . If  $\neg(p_i \succeq_j^{SD} e_j)$ , then  $\exists v'_j \in \hat{V}_j: \int_{X_i} v'_j(x) dx < \int_{\omega(j)} v'_j(x) dx$ . Hence  $X$  satisfies justified envy-freeness for private endowments.  $\square$