Cake Cutting Algorithms for Piecewise Constant and Piecewise Uniform Valuations

Haris Aziz[1] and Chun Ye[2]

[1] NICTA and UNSW, Sydney, Australia NSW 2033
haris.aziz@nicta.com.au
[2] 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 *MCSD (Mixed Constrained Serial Dictatorship)*. CCEA is polynomial-time, robust envy-free, and non-wasteful. Then, we show that there exists an algorithm *(MEA)* that is polynomial-time, envy-free, proportional, and Pareto optimal. Moreover, we show that for piecewise uniform valuations, MEA and CCEA are group-strategyproof and are equivalent to Mechanism 1 of Chen et. al.(2013). We then present an algorithm *MCSD* and a way to implement it via randomization that satisfies strategyproofness in expectation, robust proportionality, and unanimity for piecewise constant valuations. We also present impossibility results that show that the properties satisfied by CCEA and MEA are maximal subsets of properties that can be satisfied by any algorithm.

1 Introduction

Cake cutting is one of the most fundamental topics in fair division (see e.g., [7, 17]). 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 that specifies how much that agent values a particular subinterval. 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.

In this paper, we approach the cake cutting problem from a mechanism design perspective. We assume that each cake recipient, which we will refer to as *an agent*, has a private value density function over the cake. Throughout the paper we focus on the fundamental classes of value functions called *piecewise constant value density functions*.We also consider *piecewise uniform valuations* which are a restricted class of

T.-Y. Liu et al. (Eds.): WINE 2014, LNCS 8877, pp. 1–14, 2014.

piecewise constant valuations. 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. Strategyproofness has largely been ignored in cake-cutting barring a few recent exceptions [9, 10, 14, 15, 20]. The 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. A few works that are directly relevant to this paper are [8, 9, 10, 11]. Chen et al. [9, 10] presented a deterministic, strategyproof, polynomial-time, envy-free and Pareto optimal algorithm for piecewise uniform valuations. They left open the problem of generalizing their algorithm for piecewise constant valuations. Cohler et al. [11] and Brams et al. [8] formulated linear programs to compute envy-free allocations for piecewise constant valuations. However, the algorithms are not Pareto optimal in general. Two of the algorithms in our paper rely on transforming the problem of allocating a cake to agents with piecewise constant value density functions to an equivalent problem of allocating objects to agents where each agent has a homogeneous preference for each object. The transformation is done by pre-cutting the cake into subintervals using the union of discontinuity points of the agents' valuation function. This transformation allows us to adopt certain well-known results of random assignment and market equilibrium to the problem at hand.

Drawing the connection between cake cutting and random assignment, we present *CCEA (Controlled Cake Eating Algorithm)* for piecewise constant valuations. CCEA is polynomial-time and satisfies robust envy-freeness and robust proportionality, which are stronger than the notions of fairness that have been considered in the cake cutting literature. The main idea of an allocation being *robust* envy-free/proportional is that even if an agent re-adjusts or perturbs his value density function, as long as the ordinal information of the function is unchanged, then the allocation remains envy-free/proportional.[1] CCEA depends on a reduction to the generalizations [1, 13] of the *PS (probabilistic serial)* algorithm introduced by Bogomolnaia and Moulin [4] in the context of random assignments.[2]

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 deterministic, polynomial-time Pareto optimal, envy-free, and proportional for piecewise constant valuations. The main computation of MEA lies in solving the Eisenberg-Gale convex program for market equilibrium. Although similar ideas using linear programs and market equilibria have been used explicitly to compute envy-free allocations in

---

[1] Although full information is a standard assumption in cake cutting, it can be argued that it is unrealistic that agents have exact vNM utilities for each segment of the cake. Even if they do report exact vNM utilities, they may be uncertain about these reports.

[2] The CC algorithm of Athanassoglou and Sethuraman [1] is a generalization of the EPS algorithm [13] which in turn is a generalization of PS algorithm of Bogomolnaia and Moulin [4].

cake-cutting[11, 8], they do not necessarily return a Pareto optimal allocation. In a recent paper, Tian [20] characterized a class of strategyproof and Pareto optimal mechanisms for cake cutting when agents have piecewise uniform valuation functions. The algorithm of Tian involves maximizing the sum of concave functions over the set of feasible allocations. It is worth noting that MEA when restricted to the piecewise uniform valuation setting is a special case of his algorithm. We show that for piecewise uniform valuations, CCEA and MEA not only coincide but are also group-strategyproof. Previously, Chen et al. [9, 10] presented a deterministic, strategyproof, polynomial-time, envy-free and Pareto optimal algorithm for piecewise uniform valuations. We prove that for piecewise uniform valuations, CCEA and MEA are in fact equivalent to their algorithm and are group-strategyproof instead of just strategyproof.

Although CCEA and MEA are desirable algorithms, they are not strategyproof for piecewise constant valuations. This is because the incentive of the agents for the piecewise uniform valuation setting is rather limited: each agent only cares about obtaining as much of their desired pieces of the cake as possible. On the other hand, for piecewise constant valuations, agents also care about the tradeoff in quantities of having pieces at different levels of desirability. Another difficulty of obtaining a strategyproof algorithm via the aforementioned transformation is that the discontinuity points of each agent's valuation function is private information for the agent. In particular, unlike the setting of allocating multiple homogenous objects, where it suffices for an algorithm to output the fractional amount of each object that an agent will receive, the method of *conversion* from fractions of intervals into an actual allocation in terms of the union of subintervals is also a necessary step of the algorithm, which may be subject to strategic manipulation by the agents. To drive this point further, in the paper we give an example of an algorithm that is strategyproof in the random assignment setting, but is no longer strategyproof if we implement the conversion process from fractions of intervals to the union of subintervals in a deterministic fashion.

To tackle this difficulty, we present another algorithm called *MCSD (Mixed Constrained Serial Dictatorship)* which is strategyproof in expectation, robust proportional, and satisfies unanimity. For the important case of two agents[3], it is polynomial-time, and robust envy-free. To the best of our knowledge, it is the first cake cutting algorithm for piecewise constant valuations that satisfies strategyproofness, (ex post) proportionality, and (ex post) unanimity at the same time. MCSD requires some randomization to achieve strategyproofness in expectation. However, MCSD is deterministic 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 MCSD uses some essential ideas of the well-known *serial dictatorship* rule for discrete allocation, it is significantly more involved. In contrast to serial dictatorship, MCSD achieves *ex post* fairness. Our main technical results are as follows.

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

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

[3] Many fair division problems involve disputes between two parties.

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

Algorithms	Restriction	DET	R-EF	EF	R-PROP	PROP	GSP	W-GSP	SP	PO	NW	UNAN	POLYT
CCEA	-	+	+	+	+	+	-	-	-	-	+	+	+
CCEA	pw uniform	+	+	+	+	+	+	+	+	+	+	+	+
MEA		+	-	+	-	+	-	-	-	+	+	+	+
MEA	pw uniform	+	+	+	+	+	+	+	+	+	+	+	+
MCSD	-	-	-	-	+	+	-	-	+	-	-	+	-
MCSD	pw uniform	-	-	-	+	+	-	-	+	+	-	+	-
MCSD	2 agents	-	+	+	+	+	-	-	+	-	-	+	+

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

Theorem 4. *For piecewise constant valuations, there exists a randomized implementation of an algorithm (MCSD) 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 show that the properties satisfied by CCEA and MEA are maximal subsets of properties that can be satisfied by any algorithm.

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.*

Some of our main results are also summarized in Table 1. Some of our results even extend to more general domains with variable claims and private endowments. As a consequence of CCEA and MEA, we generalize the positive results regarding piecewise uniform valuations in [9, 10] and piecewise constant valuations in [11] in a number of ways such as handling richer cake cutting settings, handling more general valuations functions, achieving a stronger fairness concept, or a stronger strategyproofness notion.

2 Preliminaries

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 $N = \{1, \ldots, n\}$. Each agent has a piecewise continuous *value density function* $v_i : [0, 1] \to [0, \infty)$. 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.

Preference functions. In this paper we will only consider *piecewise uniform* and *piecewise constant* valuation 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 $\{d_1, ..., d_m\}$ that represents the consecutive discontinuity points 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.

Properties of allocations. An *allocation* is a partition of the interval $[0, 1]$ into a set $\{X_1, \ldots, X_n, W\}$, where 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. Within the cake cutting literature, the most important criteria of a fair allocation are *envy-freeness* and *proportionality*. In an *envy-free allocation*, we have $V_i(X_i) \geq V_i(X_j)$ for each pair of agent $i, j \in N$, that is every agent considers his allocation at least as good as any other agent's allocation. In a *proportional* allocation, we have $V_i(X_i) \geq \frac{1}{n} V_i([0, 1])$, that is, each agent gets at least $1/n$ of the value he has for the entire cake. Envy-freeness implies proportionality provided that every desirable part of the cake is allocated. An even stronger condition that envy-freeness is *equitability* which requires that each agent is indifferent between his allocation and the allocations of other agents.

An allocation 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 exists 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 $i \in N$. In the case where Pareto efficiency cannot be satisfied, we also consider a weaker notion of efficiency called non-wastefulness. 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.

For fairness, we do 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*. 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 $v_i' \in \hat{V}_i$, $\int_{X_i} v_i'(x)dx \geq \int_{X_j} v_i'(x)dx$. The main idea of an allocation being robust envy-free is that even if an agent re-adjusts or perturbs 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. It also addresses a criticism in cake cutting models that an agent has the ability to ascribe an exact number to each tiny segment of the cake.[4] Note that even equitability does not imply robust envy-freeness because by perturbing the valuation function, equitability can easily be lost.

Let us fix a piecewise constant value density function v. Let $(I_1, I_2, \ldots, I_k)$ be the *positively* valued intervals induced by the discontinuity points of the value function sorted in the order of decreasing preference, that is, $v(x)$ is higher on I_i than it is on I_j if $i < j$. Let x and x' be two allocation vectors whose i-th component specifies the length of I_i that is allocated to the agent, then we say that x stochastically dominates x' with respect to the preference ordering if $\sum_{i=1}^{j} x_i \geq \sum_{i=1}^{j} x_i'$ for every $j = 1, \ldots, k$. It can be shown that an allocation x for the agent with valuation function v is robust envy-free if and only if it stochastically dominates any other allocation x' with respect to the preference ordering. Moreover, it can be shown that an allocation x is robust proportional if and only if $\sum_{i=1}^{j} x_i \geq 1/n \sum_{i=1}^{j} |I_i|$ for every $j = 1, \ldots, k$. Both robust envy-freeness and robust proportionality require each agent to get a piece of cake of the same length if every agent desires the entire cake.

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 an allocation that satisfies 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 well-established in social choice and much stronger than the notion of 'strategyproofness' used in some of the cake-cutting literature (see e.g., [6]), where truth-telling is a maximin strategy and it need not be dominant strategy incentive compatible. 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 at least 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 utility that he receives under a profile

[4] Let us say that a cake is part chocolate and part vanilla. An agent may easily state that chocolate is more preferable than vanilla but would require much more effort to say that if the vanilla piece is α times bigger than the chocolate piece then he would prefer both pieces equally.

where he misreports while fixing the other agents' reports. We say that a cake cutting algorithm satisfies *unanimity*, if when each agent has positive valuation for at most $1/n$ of the cake and the interval of the cake for which he has positive valuation does not intersect with intervals of the cake for which other agents have positive valuation, then each agent is allocated the whole interval for which he has positive valuation.

Relation between the properties of cake cutting algorithms. We highlight some important relations between the main properties of cake cutting algorithms.

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 $\implies$ unanimity; *g*) two agents, proportionality $\implies$ envy-freeness; *h*) two agents, robust proportionality $\implies$ robust envy-freeness.

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 and efficiency. The free disposal assumption is necessary to ensure the algorithm of Chen et al. [10], which is a special case of two of our algorithms, to be strategyproof for piecewise uniform valuations. The existence of a non-free disposal algorithm that satisfies all of the desirable properties in the piecewise uniform setting remains an open question. Now we are ready to present our algorithms.

3 CCEA — Controlled Cake Eating Algorithm

CCEA (Controlled Cake Eating Algorithm) is based on CC (Controlled Consuming) algorithm of Athanassoglou and Sethuraman [1]. Since the original PS algorithm is also known as the simultaneous eating algorithm, we give our algorithm 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. In an assignment problem, we have a set of agents and a set of houses. Each agent has a preference ordering over the set of houses. Given two houses h and h', we will use the notation $h \succsim_i h'$ to indicate that agent i prefers h over h'. In our case, the preferences of agents over the houses are naturally induced by the relative height of the piecewise constant function lines in the respective intervals. The technical heart of the algorithm is in CC (Controlled Consuming) algorithm of Athanassoglou and Sethuraman [1]. Note that even though in the standard assignment problem, each house has a size of one and each agent has a demand of one house, the CC algorithm still applies in the case where the intervals corresponding to the houses have different lengths and there is no constraint on the total length of an agent's allocation. 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 $\mathcal{J} = \{J_1, \ldots, J_m\}$ be the set of subintervals of $[0, 1]$ formed by consecutive discontinuity points.
2 Consider $(N, H, \succsim, size(\cdot))$ where
   - $H = \{h_1, \ldots, h_m\}$ where $h_i = J_i$ for all $i \in \{1, \ldots, m\}$
   - $\succsim$ is defined as follows: $h \succsim_i h'$ if and only if $v_i(x) \geq v_i(y)$ for $x \in h$ and $y \in h'$;
   - $size(h_j) = 1$ for $h_j \in \arg\max_{j \in \{1,\ldots,m\}}(len(J_j))$;
     $size(h_j) = \frac{len(J_j)}{(\max_{j \in \{1,\ldots,m\}}(len(J_j)))}$ for all $h_j \notin \arg\max_{j \in \{1,\ldots,m\}}(len(J_j))$;
3 Discard the houses that give every agent a utility of zero from H to obtain H'.
4 $p \longleftarrow CC(N, H', \succsim, size(\cdot))$
5 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 $J_j = [a_j, b_j]$, then one possibility of J_j^i can be $[a_j + \sum_{n=1}^{i-1} p_{ih_j}/size(h_j) \times len(J_j),\ a_j + \sum_{n=1}^{i} p_{ih_j}/size(h_j) \times len(J_j)]$.
   $X_i \longleftarrow \bigcup_{j=1}^{m} J_j^i$ for all $i \in N$
6 **return** $X = (X_1, \ldots, X_n)$

Algorithm 1. CCEA (Controlled Cake Eating Algorithm)

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

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

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, we show that CCEA is not strategyproof even for two agents. Later (in Section 5), 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 restrict the preferences to piecewise uniform valuations, 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 [9].

Lemma 1. *For piecewise uniform value functions, CCEA is equivalent to Mechanism 1 of Chen et al. [9].*

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]). We show that that CCEA and hence Mechanism 1 of Chen et al. [9] is group strategyproof for piecewise uniform valuations.[5]

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

For piecewise uniform valuations, CCEA is also Pareto optimal. The result follows from Lemma 1 and the fact that Mechanism 1 of Chen et al. [9] is Pareto optimal.

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

4 MEA — Market Equilibrium Algorithm

In the previous section we presented CCEA which is not Pareto optimal for piecewise constant valuations. If we relax the robust notion of fairness to envy-freeness, then we can use fundamental results in general equilibrium theory to formulate a convex program that always returns an envy-free and Pareto optimal allocation as its optimal solution. For each valuation profile, let $J = \{J_1, \ldots, J_m\}$ 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. Once we determine the length of J_j to be allocated to an agent, we allocate any subinterval of that length to the agent. 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. [12].

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

We mention here that the connection between cake cutting and computing market equilibria is not completely new: Reijnierse and Potters [16] 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 [21]. MEA requires the machinery of convex programming. It remains open whether MEA can be implemented via linear programming. Cohler et al. [11] 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. Similarly, Brams et al. [8] used a similar connection to a linear Fisher market as MEA to compute a maxsum envy-free allocations. However the allocation they compute may not be Pareto optimal even for piecewise uniform valuations (Theorem 7, [8]). Although MEA is not robust envy-free like CCEA, it is Pareto optimal because it maximizes the Nash product. What is interesting is that

[5] Chen et al. [9] had shown that their mechanism for piecewise uniform valuations is strategyproof.

Input: Cake-cutting problem with piecewise constant valuations.
Output: A proportional, envy-free, and Pareto optimal allocation.

1 Let $J = \{J_1, \ldots, J_m\}$ be the intervals whose endpoints are consecutive points in the union of break points of the agents' valuation functions. Discard any interval that gives an utility of zero to every agent. Let x_{ij} be the length of any subinterval of J_i that is allocated to agent j.

2 $l_i \longleftarrow len(J_i)$

3 Solve the following convex program.

$$\min \quad -\sum_{j=1}^{n} log(u_j)$$

$$\text{s.t.} \quad u_j = \sum_{i=1}^{m} v_{ij}x_{ij} \quad \forall j = 1, \ldots, n; \qquad \sum_{j=1}^{n} x_{ij} \leq l_i \quad \forall i = 1, \ldots, m; \qquad x_{ij} \geq 0 \quad \forall i, j.$$

4 Let $u_j^\star$, $x_{ij}^\star$ 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 $x_{ij}^\star$.

5 $Y_j \longleftarrow \cup_{i=1}^m J_i^j$ be the allocation of each $j = 1, \ldots, n$.

6 **return** $Y = (Y_1, \ldots, Y_n)$.

Algorithm 2. MEA (Market Equilibrium Algorithm)

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. [9] 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 generalizations, we lose strategyproofness.

5 MCSD — Mixed Constrained Serial Dictatorship Algorithm

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 that satisfies all of the properties that CCEA and MEA satisfy? The answer is no, as there is no algorithm that is both Pareto optimal and robust proportional (Theorem 5). Similarly, there is no algorithm that satisfies the properties MEA satisfies along with strategyproofness (Theorem 6). Lastly, there is no algorithm that satisfies the properties CCEA satisfies plus strategyproofness (Theorem 7). 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.

We saw that CCEA and MEA are only strategyproof for piecewise uniform valuations. In light of the impossibility results established, it is reasonable to ask what other property along with strategyproofness can be satisfied by some algorithm. It follows from (Theorem 3, [19]) that the only type of strategyproof and Pareto optimal mechanisms are dictatorships. Chen et al. [10] raised the question whether there exists a strategyproof and proportional algorithm for piecewise constant valuations. The algorithm MCSD answers this question partially.

Before diving into the MCSD algorithm, it is worth noting that there is a fundamental difference between the setting of assignment objects to agents where the agents have homogeneous preferences over the objects and the cake cutting setting. In the former 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. In order to illustrate this difficulty, consider the uniform allocation rule. The uniform allocation rule (that assigns $1/n$ of each house) 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 converting $1/n$ of each interval (induced by the agent valuations) to actual subintervals is done deterministically. Chen et al. [10] resorted to randomizing the conversion process from fractions of intervals to subintervals in order to make the uniform allocation rule strategyproof in expectation.

Proposition 8. *No deterministic algorithm that implements the uniform allocation rule can be strategyproof if it also satisfies the free disposal property.*

In order to motivate MCSD, we first give a randomized algorithm that is strategyproof and robust proportional in expectation. The algorithm is a variant of random dictatorship. Under random dictatorship, if the whole cake is acceptable to each agent, then each time a dictator is chosen, he will take the whole cake which is unfair ex post. 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. This algorithm *Constrained Random Serial Dictatorship (CRSD)* draws a random permutation of the agents. It then makes the allocation to agents in the order that the lottery is drawn. Every time that 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 (breaking 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. We will see, through the proof of Proposition 10, that CRSD is robust proportional in expectation.

MCSD is an algorithm that derandomizes CRSD by looking at its allocation for all $n!$ different permutations and aggregating them in a suitable manner. The algorithm is formally presented as Algorithm 3. Although MCSD does not necessarily require $n!$ cuts of the cake, it is #P-complete to implement [3, 18] and may take exponential time if the number of agents is not a constant.

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

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

for each $\pi \in \Pi^N$ **do**
  $C \longleftarrow [0, 1]$ (intervals left)
  for $i = 1$ to n **do**
    $X^{\pi}_{\pi(i)} \longleftarrow$ maximum preference cake piece of size $1/n$ from C
    $C \longleftarrow C - X^{\pi}_{\pi(i)}$; $i \longleftarrow i + 1$.
  end for
end for
Construct a disjoint and exhaustive interval set $\mathscr{J}'$ induced by the discontinuity points in agent valuations and the cake cuts in the $n!$ cake allocations.
$Y_i \longleftarrow$ empty allocation for each $i \in N$.
for each $J_j = [a_j, b_j] \in \mathscr{J}'$ **do**
  for each $i \in N$ **do**
    Let $p_{ij} = \frac{count(i,J_j)}{n!}$ where $count(i, J_j)$ is the number of permutations in which i gets J_j.
    Generate $A_{ij} \subseteq J_j$, which is of length $p_{ij}|J_j|$ according to some subroutine.
    $Y_i \longleftarrow Y_i \cup A_{ij}$
  end for
end for
return $Y = (Y_1, \ldots, Y_n)$

Algorithm 3. MCSD (Mixed Constrained Serial Dictatorship)

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

Unlike CRSD, MCSD interprets the probability of allocating each interval to an agent as allocating 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 MCSD because a deviating agent is unable to properly evaluate his allocation against his true valuation function in the reported profile. Contrary to intuition, MCSD may or may not be strategyproof depending on how the fractional parts of each interval are allocated.

Remark 2. MCSD is not strategyproof if the fraction of each interval of $\mathscr{J}'$ is allocated deterministically.

In light of this difficulty, we will implement a method (Algorithm 5) that randomly allocates the fractions of intervals to agents. For every interval, the method chooses a starting point in the interval uniformly at random to make the cut. Agent 1 receives the left-most subinterval from the starting point with length dictated by MCSD, followed by agent 2, so on and so forth. To generate this starting point for all subintervals, it suffices to use a single bit of randomness together with the proper dilation and translation for every interval. With this implementation, MCSD is strategyproof in expectation.

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

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

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

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

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

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

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

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

Moreover, for cake cutting with piecewise uniform valuations, MCSD is not weakly group-strategyproof since RSD is not for random assignment for dichotomous preferences [5]. On the fairness front, even though MCSD satisfies both proportionality and symmetry, it does not satisfy the stronger notion of envy-freeness.

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

Another drawback of MCSD is that it is not Pareto optimal for piecewise constant valuations. However for two agents, it is robust envy-free and polynomial-time.

Proposition 14. *For two agents and piecewise constant valuations, MCSD is (ex post) robust envy-free, and polynomial-time but not Pareto optimal.*

6 Conclusion

We presented three deterministic cake-cutting algorithms — CCEA, MEA, and MCSD. We then proposed a specific randomized version of MCSD that is truthful in expectation. All the algorithms have their relative merits. Some of our results also extend to the more general cake-cutting setting in which agents do not have uniform claims to the cake or when agents are endowed with the cake pieces [2]. Cake cutting is a fundamental problem with numerous applications to computer science. In order for theory to be more relevant to practice, we envision exciting work in richer models of cake cutting.

Acknowledgments. The authors acknowledge the helpful comments of Simina Brânzei and Jay Sethuraman. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. Chun Ye acknowledges NSF grants CMMI-0916453 and CMMI-1201045.

References

1. Athanassoglou, S., Sethuraman, J.: House allocation with fractional endowments. International Journal of Game Theory 40(3), 481–513 (2011)
2. Aziz, H., Ye, C.: Cake cutting algorithms for piecewise constant and piecewise uniform valuations. Technical Report 1307.2908, arXiv.org (2013)
3. Aziz, H., Brandt, F., Brill, M.: The computational complexity of random serial dictatorship. Economics Letters 121(3), 341–345 (2013)
4. Bogomolnaia, A., Moulin, H.: A new solution to the random assignment problem. Journal of Economic Theory 100(2), 295–328 (2001)
5. Bogomolnaia, A., Moulin, H.: Random matching under dichotomous preferences. Econometrica 72(1), 257–279 (2004)
6. Brams, S.J.: Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures. Princeton University Press (2008)
7. Brams, S.J., Taylor, A.D.: Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press (1996)
8. Brams, S.J., Feldman, M., Morgenstern, J., Lai, J.K., Procaccia, A.D.: On maxsum fair cake divisions. In: Proc. of 26th AAAI Conference, pp. 1285–1291. AAAI Press (2012)
9. Chen, Y., Lai, J.K., Parkes, D.C., Procaccia, A.D.: Truth, justice, and cake cutting. In: Proc. of 24th AAAI Conference, pp. 756–761 (2010)
10. Chen, Y., Lai, J.K., Parkes, D.C., Procaccia, A.D.: Truth, justice, and cake cutting. Games and Economic Behavior 77(1), 284–297 (2013)
11. Cohler, Y.J., Lai, J.K., Parkes, D.C., Procaccia, A.D.: Optimal envy-free cake cutting. In: Proc. of 25th AAAI Conference, pp. 626–631 (2011)
12. Devanur, N., Papadimitriou, C.H., Saberi, A., Vazirani, V.: Market equilibrium via a primal–dual algorithm for a convex program. Journal of the ACM 55(5) (2008)
13. Katta, A.-K., Sethuraman, J.: A solution to the random assignment problem on the full preference domain. Journal of Economic Theory 131(1), 231–250 (2006)
14. Maya, A., Nisan, N.: Incentive compatible two player cake cutting. In: Goldberg, P.W. (ed.) WINE 2012. LNCS, vol. 7695, pp. 170–183. Springer, Heidelberg (2012)
15. Mossel, E., Tamuz, O.: Truthful fair division. In: Kontogiannis, S., Koutsoupias, E., Spirakis, P.G. (eds.) SAGT 2010. LNCS, vol. 6386, pp. 288–299. Springer, Heidelberg (2010)
16. Reijnierse, J.H., Potters, J.A.M.: On finding an envy-free Pareto-optimal division. Mathematical Programming 83, 291–311 (1998)
17. Robertson, J.M., Webb, W.A.: Cake Cutting Algorithms: Be Fair If You Can. A. K. Peters (1998)
18. Saban, D., Sethuraman, J.: The complexity of computing the random priority allocation matrix. In: Chen, Y., Immorlica, N. (eds.) WINE 2013. LNCS, vol. 8289, p. 421. Springer, Heidelberg (2013)
19. Schummer, J.: Strategy-proofness versus efficiency on restricted domains of exchange economies. Social Choice and Welfare 14, 47–56 (1997)
20. Tian, Y.: Strategy-proof and efficient offline interval scheduling and cake cutting. In: Chen, Y., Immorlica, N. (eds.) WINE 2013. LNCS, vol. 8289, pp. 436–437. Springer, Heidelberg (2013)
21. Zivan, R., Dudík, M., Okamoto, S., Sycara, K.: Reducing untruthful manipulation in envy-free pareto optimal resource allocation. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 391–398 (2010)