Optimal Envy-Free Cake Cutting

Yuga J. Cohler
Harvard SEAS
ycohler@fas.harvard.edu

John K. Lai
Harvard SEAS
jklai@post.harvard.edu

David C. Parkes
Harvard SEAS
parkes@eecs.harvard.edu

Ariel D. Procaccia
Harvard SEAS
arielpro@seas.harvard.edu

Abstract

We consider the problem of fairly dividing a heterogeneous divisible good among agents with different preferences. Previous work has shown that envy-free allocations, i.e., where each agent prefers its own allocation to any other, may not be efficient, in the sense of maximizing the total value of the agents. Our goal is to pinpoint the most efficient allocations among all envy-free allocations. We provide tractable algorithms for doing so under different assumptions regarding the preferences of the agents.

1. Introduction

We study the problem of dividing an infinitely divisible resource among several agents, often interpreted intuitively as cutting a cake. Agents have valuation functions that assign a value to each piece of cake; in general agents have different valuation functions. This problem has attracted significant attention in AI (see, e.g., Chen et al. (2010) and the references therein).

The reason for this interest is twofold. First, resource allocation is considered to be a fundamental problem in the multiagent systems literature. Second, although the cake cutting problem, which dates back to the 1940's, has mainly been studied by mathematicians, economists, and political scientists, much of the literature is algorithmic in nature, and therefore calls for the input of computer scientists.

Envy-freeness is perhaps the most well-studied notion of fairness in the context of resource allocation. An *envy-free (EF)* allocation of the cake is one where each agent (weakly) prefers its own piece of cake to the piece allocated to any other agent. Not surprisingly, in order to achieve envy-freeness one must sometimes sacrifice the economic *efficiency*[1] of the solution, that is, the total value received by the agents. However, usually there are multiple EF allocations and some are more efficient than others. An allocation is *optimal* among a set of allocations if it achieves the maximum efficiency of any allocation in the set. We focus on the case where the set of allocations is *all EF allocations*.

Most of the literature on cake cutting investigates algorithms that instruct agents to perform certain operations. In contrast, we examine an alternate algorithmic model where agents report their entire valuations to the algorithm (see, e.g., Chen et al. (2010); Zivan et al. (2010)). While this approach is infeasible for general valuation functions, it is tractable for the special cases discussed in this paper. Our goal is therefore:

> *Given the valuation functions, tractably compute an optimal EF allocation.*

In some cases we relax this goal, asking only for approximate efficiency, approximate envy-freeness, or both.

Our results. Our presentation of the results progresses through three levels of generality in terms of the supported valuation functions. In Section 3 we assume that the valuation functions are piecewise constant, i.e., can be represented by a step function. We give a polynomial-time algorithm that computes optimal EF allocations via a simple linear programming approach.

In Section 4 we deal with piecewise linear valuations, a rather general class of valuation functions that contains the class of piecewise constant valuations. We first provide an algorithm that singles out an optimal EF allocation for the case of two agents. Unfortunately, we show that in this setting, and even with two agents, no tractable algorithm exists, as in some instances any optimal EF allocation must be specified using irrational numbers. We therefore leverage our (intractable) algorithm to produce a tractable algorithm for approximately EF allocations to two agents that are as efficient as the optimal EF allocation. The algorithm runs in time polynomial in $\log 1/\epsilon$, where ϵ specifies the amount of envy permitted. Technically, this is facilitated by a delicate search procedure, which in particular employs a technique of Papadimitriou (1979) for searching for rational numbers.

Section 5 deals with general valuation functions and any number of agents. We design a tractable algorithm that computes approximately optimal, approximately EF allocations by approximating the given valuation functions by piecewise constant functions, and employing the results of Section 3. Our algorithm runs in time polynomial in $1/\epsilon$, where ϵ specifies the deviation from optimality as well as the amount of envy that is permitted.

Related work. Caragiannis et al. (2009) present a frame-

[1] Throughout the paper we use the term *efficiency* in the economic sense rather than the computational sense. Our notion of efficiency should not be confused with the weaker notion of *Pareto-efficiency*.

work for quantifying the efficiency loss due to fairness requirements, including envy-freeness, under general valuation functions. Their *price of envy-freeness* is the worst-case ratio between the total utility under an (unconstrained) optimal allocation, and the total utility under an optimal EF allocation. Caragiannis et al. provide a lower bound of $\Omega(\sqrt{n})$ and a weak upper bound of $O(n)$ on the price of envy-freeness, where n is the number of agents. Note that an upper bound singles out in every instance (set of valuation functions) an allocation that achieves a certain ratio ($O(n)$ in this case), but makes no claim as to the optimality of the allocation.

Reijnierse and Potters (1998) design a clever but involved and computationally demanding algorithm that computes a *Pareto-efficient* EF allocation, i.e., an EF allocation such that no other EF allocation is at least as good for all the agents and better for at least one agent, when agents hold piecewise constant valuations. The core of their algorithm involves a procedure for solving a market clearing problem, which can also be solved approximately using the Eisenberg-Gale convex program (Eisenberg and Gale 1959) or exactly using methods developed by Devanur et al. (2002). Crucially, it is easy to see that an optimal EF allocation is also a Pareto-efficient EF allocation. Therefore, the results we present in Section 3 provide, as a special case, a simple alternative for computing a Pareto-efficient EF allocation, also under piecewise constant valuations. Reijnierse and Potters ultimately use their algorithm to compute approximately Pareto-efficient EF allocations under general valuations; our approximation approach for general valuations, presented in Section 5, is inspired by theirs.

Zivan et al. (2010) present a way to find Pareto-efficient EF allocations that reduce untruthful manipulations, also assuming agents hold piecewise constant valuations. We do not examine strategic issues in this paper, as we discuss below.

Nuchia and Sen (2001) provide a procedure which starts from an externally given EF allocation and improves its efficiency while maintaining envy-freeness. However, this procedure is not guaranteed to produce an optimal EF allocation. In Section 4 we do provide such a guarantee by starting from an efficient allocation and improving its envy-freeness.

For general valuations, computing EF allocations is notoriously difficult (see, e.g., Procaccia (2009, Section 1)). However, there is a known approach for computing ϵ-EF allocations via Sperner's Lemma (Su 1999). If one is only interested in approximate envy-freeness, our approach is comparably simple, but significantly more general as it makes it possible to also optimize efficiency.

2. The Model

The cake is modeled as the interval $[0,1]$, and there is a set of agents $N = \{1,\ldots,n\}$. A *piece of cake* X is a finite set of disjoint subintervals of $[0,1]$. Each agent is endowed with an integrable, non-negative value density function $v_i(x)$ which defines a value for each possible piece of cake. Specifically, an agent's value $V_i(X)$ for a piece of cake X is given by $\sum_{I\in X}\int_I v_i(x)dx$. Defined in this manner, agent valuations are *additive*, i.e. $V_i(X\cup Y) = V_i(X)+V_i(Y)$ if X and Y are disjoint, and *non-atomic*, i.e., $V_i([x,x]) = 0$. Because of the latter property, we can treat open and closed intervals as equivalent. We assume that agents have equal weight, and in particular their valuation functions are normalized so that the entire cake gives each agent value one, that is, for all $i\in N$, $\int_0^1 v_i(x)dx = 1$.

An *allocation* $A = (X_1,\ldots,X_n)$ is an assignment of a piece of cake X_i to each agent, ensuring that the X_i are disjoint. Two notions of fairness have been studied in the cake cutting literature. An allocation is *proportional* with respect to $V_1,\ldots,V_n$ if $V_i(X_i)\geq 1/n$ for all $i\in N$, and *envy-free (EF)* with respect to $V_1,\ldots,V_n$ if $V_i(X_i)\geq V_i(X_j)$ for all $i,j\in N$. Envy-freeness ensures that each agent weakly prefers the piece it is given and implies proportionality when the entire interval is allocated, i.e., $\bigcup_i X_i = [0,1]$.

Because of normalization, it is meaningful to consider the sum of agent valuations and define the *efficiency* of an allocation A, denoted by $e(A)$, as the sum of agent values, i.e.,

$$e(A) = \sum_{i=1}^{n} V_i(X_i).$$

An allocation A is *optimal* amongst a set of possible allocations $\mathcal{S}$ if $e(A) = \max_{A'\in\mathcal{S}} e(A')$. In particular, we will be interested in computing an optimal allocation when $\mathcal{S}$ is the set of EF allocations.

In our algorithmic model, agents report their valuations $V_1,\ldots,V_n$ to the algorithm, which makes an allocation based on these reports. These reports may differ from the agent's true valuations. We do not consider truthful algorithms and so our algorithms can be thought to compute optimal EF allocations with respect to reported valuations, or one can assume agents are truthful. We ignore this distinction in the sequel and adopt value density v_i and value V_i in describing our algorithms.

3. Piecewise Constant Valuations

The first family of valuation functions that we consider is the family of *piecewise constant* valuations. A valuation function is piecewise constant if the associated value density function is piecewise constant, that is, the cake can be partitioned into a finite number of subintervals such that the density function is constant on each interval.

Our purpose in studying piecewise constant valuation functions is twofold. First, there are realistic situations that are captured by such valuation functions. For example, think of the cake as time, in the context of advertising or access to a shared resource. Agents may be interested only in specific time slots (e.g., during the commercial break of a specific program), but are indifferent between different parts of the desired slot. Second, the results of this section will be leveraged in Section 5 to address general valuations.

The main result of this section is a simple polynomial-time algorithm for finding an optimal EF allocation when agents have piecewise constant valuations. In order to discuss computational complexity, we must describe the representation of the input to the algorithm. While general valuation functions do not admit concise descriptions, piecewise

Algorithm 1

```
1. Mark the boundaries of the reported intervals of all agents,
   as well as 0 and 1.
2. Let $\mathcal{J}$ be the set of subintervals of $[0,1]$ formed by
   consecutive marks.
3. Solve the following linear program:
       max   $\sum_{i=1}^{n} \sum_{I \in \mathcal{J}} x_{iI} V_i(I)$                          (1)
       s.t.  $\sum_{i=1}^{n} x_{iI} \le 1 \; \forall I \in \mathcal{J}$                          (2)
             $\sum_{I \in \mathcal{J}} x_{iI} V_i(I) \ge \sum_{I \in \mathcal{J}} x_{jI} V_i(I) \; \forall i,j \in N$   (3)
             $x_{iI} \ge 0 \; \forall i \in N, I \in \mathcal{J}$                          (4)
4. Return an allocation which for all $i \in N$ and $I \in \mathcal{J}$
   allocates an $x_{iI}$ fraction of subinterval $I$ to agent $i$.
```

constant functions have a conveniently simple representation. Each agent reports the boundaries of the inclusion-maximal intervals on which the agent's density function is constant, along with the value of the density function on each of these intervals. The size of the input is the number of bits required to report these parameters. We assume that the boundaries of intervals as well as the value of the density function can be expressed as *k-bit rationals*, i.e., rational numbers of the form a/b where a and b are k-bit integers.

Our procedure for finding an optimal EF allocation is formally given as Algorithm 1. Step 1 of the algorithm is illustrated in Figure 1.

The linear program (LP) in Step 3 has variables x_{iI} for each $i \in N$ and $I \in \mathcal{J}$ (where $\mathcal{J}$ is defined in Step 2), which represent the fraction of interval I given to agent i. Crucially, the value density functions of all agents are constant on each interval $I \in \mathcal{J}$, hence the value of each agent $i \in N$ for a fraction x_{iI} of interval I is $x_{iI}V_i(I)$, and the agent's value for its piece is $\sum_{I \in \mathcal{J}} x_{iI}V_i(I)$. The objective function (1) then simply gives the efficiency of the allocation. The first constraint (2) ensures that the allocation of each interval in $\mathcal{J}$ is valid, while the second constraint (3) is simply the envy-freeness constraint. We have the following result.

Theorem 1. *Assume that there are n agents with piecewise constant valuation functions. Then Algorithm 1 computes an optimal EF allocation in polynomial time.*

Interestingly, setting the variables x_{iI} to $1/n$ for every $i \in N$ and $I \in \mathcal{J}$—allocating to each agent a $1/n$-fraction of each interval in $\mathcal{J}$—produces an allocation where $V_i(X_j) = 1/n$ for every $i, j \in N$; this is what Chen et al. (2010) call a *perfect allocation*. A perfect allocation is in particular EF. So, under piecewise constant valuation functions finding an EF allocation is trivial, and computing an optimal EF allocation only slightly less so.

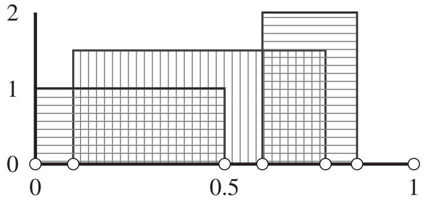

Figure 1: An illustration of piecewise contant value density functions, where $n = 2$ and the area under the density function of agent 1 (resp., agent 2) is filled with horizontal (resp., vertical) lines. Marks made by Algorithm 1 are represented by white circles on the horizontal axis. Note that both value density functions are constant between every pair of consecutive marks.

4. Piecewise Linear Valuations

Piecewise linear valuations are a significantly more general family of valuation functions that includes piecewise constant valuations. An agent's valuation function is *piecewise linear* if its value density function is piecewise linear. Piecewise linear functions offer added expressiveness, yet can still be concisely represented. The agent's valuation function can be pinned down by breaking down $[0, 1]$ into a set of subintervals on which the agent's value density function has constant slope. The agent then specifies the boundaries of each of these intervals as well as the slope and intercept of the density function on the interval.

While Algorithm 1 exactly solves the piecewise constant case, it is not directly generalizable to the piecewise linear case. The algorithm relied on the fact that we could split $[0, 1]$ into a finite number of intervals on which agent value densities were constant. This allowed us to focus only on the fraction of each interval given to each agent rather than the specific part of the interval. With piecewise linear valuations, it is not longer possible to split $[0, 1]$ into a finite number of intervals on which value densities are constant.

The main result of this section is an algorithm that finds an optimal EF allocation for two agents when valuations are piecewise linear.[2] We first outline an abstract algorithm for handling these valuation functions. We then prove an impossibility result that an exact implementation of this abstract algorithm is intractable. We conclude by sketching an approximate implementation of the algorithm. An algorithm for any number of agents is left open.

An abstract algorithm

At a high level, the algorithm starts with an optimal (not necessarily EF) allocation, and transfers pieces to the envious agent until the agent is no longer envious. The crux of the procedure lies in the choice of which pieces are given to the envious agent. A key notion will be that of the ratio between the density functions of agent 1 and agent 2.

Definition 2. Given $x \in [0, 1]$ where $v_2(x) \neq 0$, the *value ratio at x* is $R(x) = v_1(x)/v_2(x)$.

[2] Envy-freeness and proportionality are equivalent in the case of two agents, so the algorithm equivalently finds an optimal proportional allocation.

Algorithm 2

```
1. If $V_1(Y_{1\geq 2}) \geq 1/2$ and $V_2(Y_{2\geq 1}) \geq 1/2$, give agent 1
   $Y_{1>2}$, agent 2 $Y_{2>1}$.
   (a) If $V_1(Y_{1>2}) \geq 1/2$, give $Y_{1=2}$ to agent 2.
   (b) Otherwise, divide $Y_{1=2}$ so that agent 1 receives value
       exactly 1/2.
2. Without loss of generality, assume $V_1(Y_{1\geq 2}) < 1/2$. Give
   $Y_{1\geq 2}$ to agent 1. Let $r^*$ be the maximal $r$ such that
   $V_1(Y_{1\geq 2} \cup Y_{\geq r}) \geq 1/2$. Give $Y_{>r^*}$ to agent 1, and
   divide $Y_{=r^*}$ so that agent 1 receives exactly value 1/2.
```

Notationally, for $i, j \in \{1, 2\}$ let $Y_{i\ op\ j} = \{x \in [0, 1] : v_i(x)\ op\ v_j(x)\}$, where $op \in \{>, \geq, =\}$. For instance, $Y_{1\geq 2} = \{x \in [0, 1] : v_1(x) \geq v_2(x)\}$. Let

$$Y_{op\ r} = \{x : (v_1(x) < v_2(x)) \wedge (R(x)\ op\ r)\},$$

where $op \in \{>, \geq, =\}$. Using these notations we can present our algorithm, given as Algorithm 2. In the rest of this subsection we prove the following theorem.

Theorem 3. *Assume that there are two agents with piecewise linear valuations. Algorithm 2 finds an optimal EF allocation.*

Before proving Theorem 3, we establish a few useful lemmas.

Lemma 4. *Suppose that agent $i \in \{1, 2\}$ receives a piece of cake X_i, with $V_i(X_i) \geq 1/2$. Agent i will not envy the other agent.*

Proof. By additivity, $V_i(X_i) + V_i([0, 1] \setminus X_i) = 1$. The proposition follows by observing that the other agent receives at most $[0, 1] \setminus X_i$ if agent i receives X_i. □

Lemma 5. *In any optimal EF allocation, $V_1(X_1) \geq 1/2$ and $V_2(X_2) \geq 1/2$.*

Proof. To prove this lemma, we first show that any optimal EF allocation allocates all intervals on which some agent has strictly positive value. Suppose for contradiction that there is some optimal EF allocation X_1, X_2 that does not allocate an interval I where $v_1(I) > 0$ or $v_2(I) > 0$. We can augment X_1, X_2 with an allocation of I that maintains envy-freeness while improving efficiency. Indeed, assume without loss of generality that $V_1(I) > 0$. Divide I into two subintervals I', I'' such that $V_1(I') = V_1(I'')$. Allocate to agent 2 the subinterval with higher value according to V_2, and give the remaining subinterval to agent 1. Efficiency is improved because agent 1 receives strictly greater value and agent 2 receives weakly greater value. Envy-freeness is maintained since agent 1 is indifferent between the two pieces, and agent 2 prefers the additional piece it receives.

Thus, all desired intervals are allocated to one of the agents, so for $i \in \{1, 2\}$, $V_i(X_1) + V_i(X_2) = 1$, and envy-freeness requires that $V_i(X_i) \geq 1/2$. □

We are now ready to prove Theorem 3.

Proof of Theorem 3. Consider each of the cases specified by Algorithm 2.

Case 1: $V_1(Y_{1\geq 2}) \geq 1/2, V_2(Y_{2\geq 1}) \geq 1/2$. Algorithm 2 allocates $Y_{1>2}$ to agent 1 and $Y_{2>1}$ to agent 2. The allocation made by Algorithm 2 is always efficient, since the agent who strictly prefers an interval always receives it. What is left to be shown is that the allocation is EF.

Case 1(a): $V_1(Y_{1>2}) \geq 1/2$. Algorithm 2 gives $Y_{1=2}$ to agent 2. $V_2(Y_{2\geq 1}) \geq 1/2$ by assumption. Both agents have value at least 1/2, and by Lemma 4 are not envious.

Case 1(b): $V_1(Y_{1>2}) < 1/2$. Algorithm 2 splits $Y_{1=2}$ so that agent 1 receives value exactly 1/2 after adding in $Y_{1>2}$. This must be possible since $V_1(Y_{1\geq 2}) \geq 1/2$. Agent 1 is not envious by Lemma 4. Let X_2 be the piece given to agent 2 (the remaining portion of $Y_{1=2}$ along with $Y_{2>1}$). Algorithm 2 allocates the entire interval, so by additivity, $V_1(X_2) = 1/2$. However, the piece X_2 consists only of intervals where $v_2(x) \geq v_1(x)$, so $V_2(X_2) \geq V_1(X_2) = 1/2$.

Case 2: $V_1(Y_{1\geq 2}) < 1/2$. First, note that Algorithm 2 finds an EF allocation X_1, X_2. Indeed, as before, agent 1 is not envious as $V_1(X_1) = 1/2$. Since agent 2 is given all the intervals not given to agent 1, $V_1(X_2) = 1/2$. The piece X_2 consists only of intervals where $v_2(x) > v_1(x)$, so $V_2(X_2) \geq V_1(X_2) = 1/2$.

Because $V_1(Y_{1\geq 2}) < 1/2$, envy-freeness requires us to sacrifice efficiency since we need to give agent 1 some intervals that are strictly preferred by agent 2. To show that X_1, X_2 is an optimal EF allocation, let X_1', X_2' be any optimal EF allocation. Define the following three pieces of cake:

$$\begin{aligned}
A &= X_1 \cap X_1' \cap Y_{2>1},\\
B &= (X_1 \setminus X_1') \cap Y_{2>1},\\
C &= (X_1' \setminus X_1) \cap Y_{2>1}.
\end{aligned}$$

A gives the intervals where both allocations lose efficiency due to giving piece preferred by agent 2 to agent 1. B gives the intervals where X_1, X_2 loses efficiency, and C gives the intervals where X_1', X_2' loses efficiency.

Let $V_1(Y_{1\geq 2}) = 1/2 - \epsilon, \epsilon > 0$. Note that $A \cap B = \emptyset, A \cap C = \emptyset$, and $A \cup B = X_1 \cap Y_{2>1}, A \cup C = X_1' \cap Y_{2>1}$. Algorithm 2 gives agent 1 exactly value 1/2 yielding:[3]

$$V_1(A) + V_1(B) = \int_A v_1(x)dx + \int_B v_1(x)dx = \epsilon. \quad (5)$$

Similarly, Lemma 5 says that since X_1', X_2' is an optimal EF allocation, agent 1 must receive value at least ϵ from its allocation of $Y_{2>1}$:

$$V_1(A) + V_1(C) = \int_A v_1(x)dx + \int_C v_1(x)dx \geq \epsilon. \quad (6)$$

Combining (5) and (6) yields

$$\int_C v_1(x)dx - \int_B v_1(x)dx \geq 0. \quad (7)$$

[3] We slightly abuse notation and take the integral over A, B, C to signify the sum of integrals over inclusion-maximal subintervals of A, B, C respectively.

Let $\ell(X_1, X_2)$ denote the difference between the efficiency of the optimal allocation (not necessarily EF) and the efficiency of X_1, X_2.

$$\ell(X_1, X_2) = \int_A (v_2(x) - v_1(x))dx + \int_B (v_2(x) - v_1(x))dx$$
$$\ell(X_1', X_2') \geq \int_A (v_2(x) - v_1(x))dx + \int_C (v_2(x) - v_1(x))dx$$

The loss for X_1', X_2' is an inequality because while Algorithm 2 gives all of $Y_{1>2}$ to agent 1, X_1', X_2' need not and may lose efficiency from those intervals as well.

To complete the proof, recall how Algorithm 2 constructs X_1. Let r^* be the value computed in Step 2 of Algorithm 2. By definition of Algorithm 2, $X_1 \cap Y_{2>1}$ consists of all points with $R(x) > r^*$ and some or all of the points with $R(x) = r^*$. Therefore, if $x \in B$ then $R(x) \geq r^*$, and if $x \in C$ then $R(x) \leq r^*$. We conclude that

$$\begin{aligned}
&\ell(X_1', X_2') - \ell(X_1, X_2) \\
&\geq \int_C (v_2(x) - v_1(x))dx - \int_B (v_2(x) - v_1(x))dx \\
&= \int_C \left(\frac{v_1(x)}{R(x)} - v_1(x)\right) dx - \int_B \left(\frac{v_1(x)}{R(x)} - v_1(x)\right) dx \\
&\geq \left(\frac{1}{r^*} - 1\right)\left(\int_C v_1(x)dx - \int_B v_1(x)dx\right) \\
&\geq 0,
\end{aligned}$$

where the last inequality follows from (7). □

Interestingly, Algorithm 2 does not make specific use of the piecewise linearity assumption. In theory, it can be applied to more general classes of valuation functions, provided that the sets $Y_{1>2}, Y_{1=2}, Y_{2>1}, Y_{\geq r}$ correspond to legal pieces of cake. However, we do use the piecewise linearity assumption in the next subsection.

Implementing Algorithm 2

To discuss implementation details, we need to discuss the representation of the input. As alluded to earlier, piecewise linear valuations can be represented by asking agents to partition $[0, 1]$ into intervals on which their value density functions have constant slope. The agents then report the boundaries of the intervals as well as the slope and intercept of the density function on each interval. We assume that all points can be specified with k-bit rationals. Since slopes and intercepts can be negative, in this section we take k-bit rationals to include numbers of the form $-a/b$ where a, b are k-bit rationals.

While it is tempting to apply Algorithm 2 to produce an optimal EF allocation, there is a barrier to this approach. Even when the inputs are k-bit rational numbers, the r^* defined in Step 2 of Algorithm 2 and the boundaries of the resulting allocation may be irrational. In fact, this limitation is not specific to Algorithm 2. There are cases where the allocation computed by Algorithm 2 is the *unique* optimal EF allocation and has irrational boundaries. The proof is omitted due to lack of space.

Theorem 6. *There exist piecewise linear valuations whose interval boundaries, slopes, and intercepts are all rational numbers yet whose optimal EF allocations can only be specified with irrational numbers.*

As a result, it is necessary to resort to approximation. Indeed, we relax envy-freeness by considering approximately EF allocations. Specifically, an allocation is *ϵ-EF* if for all $i, j \in N$, $V_i(X_i) \geq V_i(X_j) - \epsilon$ (see, e.g., Lipton et al. (2004)). The following theorem formally presents our approximation guarantees.

Theorem 7. *Assume that there are two agents with piecewise linear valuations. For any $\epsilon > 0$ there is an algorithm that runs in time polynomial in the input and $\log(1/\epsilon)$, and finds an ϵ-EF allocation A' such that $e(A') \geq e(A)$, where A is an optimal EF allocation.*

The theorem's proof is omitted due to lack of space, but we give a very brief sketch. In the case considered in Step 1 of Algorithm 2, we would like to find a point x^* such that

$$V_1(([0, x^*] \cap Y_{1=2}) \cup Y_{1>2}) = 1/2.$$

Using binary search over $[0, 1]$, we find a point x that is smaller but very close to x^*. It is then possible to bound the envy, while the resulting allocation is at least as efficient as the optimal EF allocation. In Step 2 of Algorithm 2, we need to search for a ratio r close to r^*. This is more subtle, because very small differences in $|r - r^*|$ can lead to significant differences in the derived value when there is a long interval with constant value ratio. Fortunately, in this problematic case it can be shown that r^* is a rational, and hence it is sufficient to find the rational r closest to r^*. This can be done using a delicate search over rationals, via techniques due to Papadimitriou (1979).

5. General Valuations

In this section we give a method for handling general valuation functions (under some mild conditions) and for any number of agents. We approximate general valuation functions with piecewise constant valuations and leverage Algorithm 1. We construct an allocation that is ϵ-EF and whose efficiency is within ϵ of the optimal EF allocation. Our central observation is the following lemma, whose proof is omitted due to lack of space.

Lemma 8. *Given $\epsilon > 0$ and value density functions $v_1, \ldots, v_n$, suppose that $v_1', \ldots, v_n'$ are piecewise constant value density functions such that for all $i \in N$,*

$$v_i(x) \leq v_i'(x) \leq v_i(x) + \epsilon/2. \tag{8}$$

Let $A = (X_1, \ldots, X_n)$ be an optimal EF allocation with respect to valuations V_i (induced by v_i), and let $A' = (X_1', \ldots, X_n')$ be an $\epsilon/2$-EF allocation with respect to valuations V_i' (induced by v_i'). Then A' is ϵ-EF and $e(A') \geq e(A) - \epsilon/2$.

Given piecewise constant value density functions $v_1', \ldots, v_n'$ that satisfy (8), it is easy to find an $\epsilon/2$-EF allocation A' by applying Algorithm 1 to these valuations, where the envy-freeness constraint (3) is relaxed by $\epsilon/2$.

To find $v'_1, \ldots, v'_n$ as required by Lemma 8, we assume that $v_1, \ldots, v_n$ are K-Lipschitz, i.e., for all $x, y \in [0, 1]$,

$$|v_i(x) - v_i(y)| \leq K \cdot |x - y|.$$

Now, split $[0, 1]$ into $\lceil (4K)/\epsilon \rceil$ intervals of size at most $\epsilon/(4K)$. Let $S = \{k/2^p : k \in [0, M2^p]\}$, where M is an upper bound on $v_i(x)$ for all $i \in N$ and $x \in [0, 1]$, and p will be specified later. For each interval I and agent i, let $v^*(I) = \max_{x \in I} v_i(x)$. For all $x \in I$ let $v'_i(x) = s^*(I)$, where $s^*(I) = \min\{s \in S : s \geq v^*(I)\}$.

The K-Lipschitz condition ensures that the density function varies by at most $\epsilon/4$ on each interval. Letting $p = \lceil 2 + \log(1/\epsilon) \rceil$, $s^*(I) - v^*(I) \leq \epsilon/4$, so v'_i satisfies condition 8.

While the K-Lipschitz condition rules out valuation density functions with discontinuities, our results extend to valuation density functions with a finite number of discontinuities that are K-Lipschitz on each continuous subinterval. In particular, we can use the described procedure separately on each continuous subinterval to find v'_i that satisfy (8). We have the following theorem.

Theorem 9. *Assume that there are n agents with value density functions $v_1, \ldots, v_n$ that have a finite number of discontinuities, are K-Lipschitz on each continuous subinterval, and have maximum value M. For any $\epsilon > 0$, there is an algorithm that runs in time polynomial in $n, \log M, K, 1/\epsilon$ and computes an ϵ-EF allocation whose efficiency is within ϵ of the optimal EF allocation.*

Given the rather strong Theorem 9, one may wonder in what way the results of Section 4 are superior. In fact, for the (interesting, we believe) case of two agents with piecewise linear valuations, the method of Section 4 has two technical advantages. First, it produces an ϵ-EF allocation that is as efficient as the optimal EF allocation. Second, Theorem 7 provides running time that is polynomial in the representation and therefore logarithmic in the slope of the valuation functions, as the slope is specified by $O(k)$-bit rationals. In contrast, the running time in Theorem 9 is polynomial in the slope (since the maximum slope determines the Lipschitz constant), and hence exponential in the representation. Finally, note that piecewise linear (rather than constant) valuation functions can in theory be used to approximate *general* valuations, making it possible to relax the assumptions of Theorem 9 (when there are only two agents).

6. Discussion

Interestingly, envy-freeness can be replaced with the weaker notion of proportionality in all of our results. As mentioned in Section 2, any EF allocation is proportional, and for the case of two agents the two notions coincide. Using the last observation, the results of Section 4 immediately hold for proportionality. The results of Section 3 can easily be adapted by modifying (3), implying that the results of Section 5 hold as well. The purpose of our focus on envy-freeness is to simplify the exposition.

In Sections 3 and 4 we assume, as in Chen et al. (2010), that agents report their entire valuation function. This is possible because piecewise constant and piecewise linear valuations are concisely representable. Of course, in Section 5 we cannot adopt this model. However, notice that Theorem 9 merely requires finding values that are close to $v_i(x)$ for a polynomial number of points $x \in [0, 1]$; an implicit, reasonable assumption is that the valuation information at these points can be elicited from agents.

In contrast to Chen et al. (2010), we do not attempt to design *truthful* algorithms. The algorithms of Chen et al. are truthful and yield EF allocations, but these allocations may be highly inefficient compared to other EF allocations. Furthermore, it is known that a truthful algorithm cannot produce efficient allocations (Thomson 2007). A natural direction for future research is quantifying how much efficiency must be sacrificed to obtain truthfulness, in the spirit of recent work on approximate mechanism design without money (Procaccia and Tennenholtz 2009).

References

Caragiannis, I.; Kaklamanis, C.; Kanellopoulos, P.; and Kyropoulou, M. 2009. The efficiency of fair division. In *Proc. of 5th WINE*, 475–482.

Chen, Y.; Lai, J. K.; Parkes, D. C.; and Procaccia, A. D. 2010. Truth, justice, and cake cutting. In *Proc. of 24th AAAI*, 756–761.

Devanur, N. R.; Papadimitriou, C. H.; Saberi, A.; and Vazirani, V. V. 2002. Market equilibrium via a primal-dual-type algorithm. In *Proc. of 43rd FOCS*, 389–395.

Eisenberg, E., and Gale, D. 1959. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics* 30(1):165–168.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proc. of 6th EC*, 125–131.

Nuchia, S. W., and Sen, S. 2001. Improving optimality of n agent envy-free divisions. In *Proc. of 8th ATAL*, 277–289.

Papadimitriou, C. 1979. Efficient search for rationals. *Information Processing Letters* 8:1–4.

Procaccia, A. D., and Tennenholtz, M. 2009. Approximate mechanism design without money. In *Proc. of 10th EC*, 177–186.

Procaccia, A. D. 2009. Thou shalt covet thy neighbor's cake. In *Proc. of 21st IJCAI*, 239–244.

Reijnierse, J., and Potters, J. 1998. On finding an envy-free Pareto-optimal division. *Mathematical Programming* 83:291–311.

Su, F. E. 1999. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly* 106(10):930–942.

Thomson, W. 2007. Children crying at birthday parties. Why? *Journal of Economic Theory* 31:501–521.

Zivan, R.; Dudík, M.; Okamoto, S.; and Sycara, K. P. 2010. Reducing untruthful manipulation in envy-free Pareto optimal resource allocation. In *Proc. of 10th IAT*, 391–398.