OPERATIONS RESEARCH
Vol. 60, No. 6, November–December 2012, pp. 1461–1476
ISSN 0030-364X (print) | ISSN 1526-5463 (online)
http://dx.doi.org/10.1287/opre.1120.1116

Algorithmic Solutions for Envy-Free Cake Cutting

Xiaotie Deng
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom; and
Department of Computer Science, City University of Hong Kong, Hong Kong
{xiaotie@liv.ac.uk; deng@cs.cityu.edu.hk}

Qi Qi
Department of Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology, Hong Kong,
kaylaqi@ust.hk

Amin Saberi
Department of Management Science and Engineering, Stanford University, Stanford, California 94305,
saberi@stanford.edu

We study the problem of finding an envy-free allocation of a cake to $d+1$ players using d cuts. Two models are considered, namely, the oracle-function model and the polynomial-time function model. In the oracle-function model, we are interested in the number of times an algorithm has to query the players about their preferences to find an allocation with the envy less than ϵ. We derive a matching lower and upper bound of $\theta(1/\epsilon)^{d-1}$ for players with Lipschitz utilities and any $d > 1$. In the polynomial-time function model, where the utility functions are given explicitly by polynomial-time algorithms, we show that the envy-free cake-cutting problem has the same complexity as finding a Brouwer's fixed point, or, more formally, it is PPAD-complete. On the flip side, for monotone utility functions, we propose a fully polynomial-time algorithm (FPTAS) to find an approximate envy-free allocation of a cake among three people using two cuts.

Subject classifications: fair division; cake cutting; envy-free; FPTAS; fixed point; PPAD.
Area of review: Optimization.
History: Received April 2010; revisions received March 2011, September 2011; accepted January 2012.

1. Introduction

Suppose you have a cake and you would like to divide it among n persons. Each person may have a different opinion as to which part is more valuable. Is there a way of cutting a cake into n pieces with $n-1$ cuts and allocating one piece to each person so that everyone values his or her piece no less than any other piece? Is there an efficient algorithm that finds such a way, or commonly what is known as an envy-free solution?

Since the beautiful works of Banach and Knaster (see Brams and Taylor 1995) and then Steinhaus (1948, 1949), the envy-free cake-cutting problem has been a favorite among mathematicians for its properties and algorithmic solutions (see, e.g., Even and Paz 1984, Brams and Taylor 1995, Woeginger and Sgall 2007). Its core applications in operations research and related fields, including fair scheduling, resource allocation, and conflict resolution have been studied extensively, where every participant desires a solution that optimizes its own utility.

1.1. Applications in Operations Research and Economics

Many interesting operations research applications within the cake-cutting framework are presented by Thomson (2007). Some of the examples include allocating staff to time-intensive tasks such as scheduling police patrol operations and allocation of cleaning tasks to maintenance crews. Thomson also studies territory-splitting applications, for example, for shop placement, and proposes a cake-cutting protocol.

Similar territory-splitting operations are also discussed in Sherstyuk (1998) for understanding partisan gerrymandering schemes. Again, the author uses fair cake-cutting approaches to develop methods that maintain electoral fairness. It is also worth noting that the issue of connectedness of the districts, the property also enforced in our problem, is of vital importance for this application.

In the same spirit, Moulin (2007) targets the issue of fairness in scheduling and queuing decisions. Meunier (2008) studies the more abstract problem of fairly allocating a discrete object represented as a necklace. Other discrete variations of fair allocation have also attracted a lot of attention, both in algorithms (Lipton et al. 2004, Asadpour and Saberi 2007, Bansal and Sviridenko 2006) and communication networks (Kelly et al. 1998, Kumar and Kleinberg 2006, Kleinberg et al. 1999).

There is also an extensive literature in the broader context of envy-free allocation. Halland and Zeckhauser (1979) develop an implicit market procedure to elicit honest preferences from individuals' private preferences by assigning the players efficiently to goods or positions such that their different objectives are maximized within the choices. A true story example, reported in Pratt and Zeckhauser (1990), deals with the allocation of two trunks of silver left behind to eight grandchildren. The designed procedure is in accordance with a market equilibrium approach, divided in three steps: eliciting utility functions, solving for the equilibrium, and implementing a randomization step for the final integer solution. The story shows a real success of using the equilibrium procedure to solve the envy-free allocation problem. The utility function here is not an oracle, but explicitly given. That makes it more efficient than using the oracle model to query the players from time to time.

In the context of bankruptcy management, Abdelghani et al. (Elimam et al. 1996) find a solution for fair and equitable distribution of debtors' mixed assets to their respective creditors. In a different direction of firm acquisition and merging, Hirsch (1986) studies the transition from hostile ambushes to smooth transition of the power, providing incentives to sustain order within the business. While the study focuses on the social aspect of the changes, it reveals the underlying principles that have made the social change: an amiable model that tries to maximize the utility of everyone involved, a situation advocated in the envy-free cake-cutting paradigm.

1.2. Related Work

Various notions have been proposed to capture fairness. A related fairness criteria that is less demanding than envy-freeness is proportionality. A solution is proportional if each person gets a piece that he or she values more than $1/n$ of total.

For two players, finding a proportional solution is quite simple: cut and choose. One gets to cut the cake, and the other gets to choose which piece. The cut and choose protocol results in an envy-free, and hence also a proportional, solution. A generalization of this scheme for finding a proportional solution for $n > 2$ players is discovered by Banach and Knaster (see Brams and Taylor 1995) and Steinhaus (1948, 1949). Later, Even and Paz (1984) give an elegant divide-and-conquer algorithm for this problem that runs in $O(n \log n)$. Very recently, Edmonds and Pruhs prove that this algorithm is essentially optimum for deterministic algorithms (Edmonds and Pruhs 2006a) and give a randomized algorithm with running time $O(n)$ (Edmonds and Pruhs 2006b).

The problem can also be relaxed to allow more than $n-1$ cuts. In that case, there is an envy-free solution for three players found around 1960 by Selfridge and is rediscovered independently by Conway (see Brams and Taylor 1995 for details). The solution for the n-player envy-free allocation is not known until the result of Brams and Taylor in 1995 (Brams and Taylor 1995).

For the problem under our consideration, that is the n-person envy-free cut problem with exactly $n-1$ cuts, progress in complexity analysis has been limited. The existence of such a solution is proven by Stromquist (1980) with a fixed-point argument. His proof implies that an ϵ-approximation can be found in time exponential in input size $O(\log 1/\epsilon)$, where $\epsilon > 0$ can be considered as an insignificant extra amount in the value of their utility functions that the players would not care. Later, Simmons and Su (Su 1999) prove a similar result by Sperner's lemma.

Stromquist (1980) also proposes a moving knife procedure for cutting a cake for three players, that results in exactly two cuts. We will explain this procedure in details in §4. It is easy to see that Stromquist's solution requires frequent recalculations of the knife positions, and it may take an exponential time. We will show a stronger result: the particular fixed point found by Stromquist needs an exponential number of queries about the values of the players.

1.3. Our Contributions and Methodology

The main question studied in this paper is whether it is possible to find an envy-free allocation by an efficient procedure. Because utilities can be arbitrary continuous functions, it may be impossible to describe the envy-free solution with a finite number of digits. Instead, we would be interested in solutions with $\epsilon > 0$ relaxation in envy. That is, we are interested in a partition of the cake among the participants such that each would not envy another's piece more than an ϵ.

For simplicity, let $N = 1/\epsilon$ throughout our discussion. We are interested in whether there are algorithms with the time complexity polynomial in $O(\log N)$, i.e., whether there is an algorithm in time polynomial in the bit size of the approximation parameter.

The approximation approach allows us to consider the discrete version of the problem. A triangulated simplex is a very useful tool in the study of computational complexity. In a triangulation, a d-dimensional simplex is partitioned into a collections of smaller d-dimensional simplices such that any two of them have either no common element or a face of them. Such a d-dimensional simplex in the triangulation will be called a base simplex or a base cell in the following discussion.

1.3.1. The Oracle-Function Model vs. the Polynomial-Time Function Model. We are interested in two types of functions for computing the utilities or the preferences of the players. In the polynomial-time function model, the agents have to represent their utilities fully by a polynomial-time algorithm. In the oracle model, on the other hand, the algorithm gets a partial picture of the utilities of the agents after every question. Because of this, our problem is harder under the oracle-function model than under the polynomial-time function model. This is simply because the oracle-function model can admit many more functions than the polynomial-time function model.

Both models are quite natural. Oracle-function models are suitable in situations when the agents are unable or unwilling to communicate their full utility functions. This can happen because the full utility function is too long (e.g., when N is large in our context) or when it is costly for the agent to evaluate his or her utility function for every possible scenario. Oracle-function models are also used in the context of learning theory and combinatorial auctions (Blum et al. 2004, Blumrosen and Nisan 2007, Dobzinski et al. 2010).

The polynomial utility functions are suitable when the utilities can be described concisely, for example, when they are linear or well-behaved concave functions. In this model, it is assumed that the agent can represent his or her utility by a simple program or a polynomial-time function.

For both models, we prove that the time complexity is equivalent to that of finding a fixed point for general utility functions.

1.4. Main Results

We establish three main results.

1. When the preferences of the players are provided as polynomial-time algorithms, we prove that our cake-cutting problem is PPAD-complete (as defined in §2.4). By proving PPAD-completeness, we establish that the complexity of finding an envy-free cake cutting is the same as finding a Brouwer's fixed point, or finding a Nash equilibrium or a market equilibrium.

2. We also study the case in which the preferences are given by an oracle function. In other words, we consider the setting in which the algorithm can query the preferences of the players by showing them a particular cake cutting and asking for their preferred piece (or pieces) in the cut. In this setting, we are interested in the minimum number of queries needed for finding an ϵ-approximate cake cutting for $d+1$ players. We derive a matching lower and upper bound of $\theta(N^{d-1})$ for the query complexity (see §2.4 for more details).

Despite a strong connection, mathematical and complexity-wise, between equilibrium computation and fixed-point computation, this is the first *matching* query complexity result for an equilibrium computation problem. For example, we know of no such result for Nash equilibrium, which is also PPAD-complete with a strong tie to fixed-point computation.

3. For the special case of monotone utility functions, we prove there is a fully polynomial-time approximation scheme (see Definition 7 in §2.2) for finding an approximate envy-free allocation of a cake among three people using only two cuts. In other words, we are able to find an approximate solution in time polynomial of the number of players and $O(\log(1/\epsilon))$.

1.4.1. PPAD Completeness. Our proof starts with the problem of finding a fully colored base simplex in a triangulated d-dimensional simplex with a valid Sperner coloring, which is a known PPAD-complete problem.

We define a discrete problem in which the goal is to find an approximate envy-free cut. Let N be a large integer. Consider all the d-cuts represented by barycentric coordinates $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_d)$ with $\alpha_0 = X_0/N$, $\alpha_1 = X_1/N, \ldots, \alpha_d = X_d/N$, where X_is are integers between 0 and N and $\sum X_i = N$. Note that $(\alpha_0, \alpha_1, \ldots, \alpha_d)$ is a point on the standard d-dimensional simplex. We may use $\mathbf{X} = (X_0, X_1, \ldots, X_d)$ as the integer representation of a d-cut.

To prove that our problem is in PPAD, we use Kuhn's triangulation (Kuhn 1960) to triangulate the simplex so that the corner points of the base cells represent the discrete cuts defined above for some N. Then, the reduction is done by a two-stage process: labeling and coloring.

The usefulness of Kuhn's triangulation in efficient implementation has also been noted by Su and used in his implementation, the Fair Division Calculator (at Su's webpage http://www.math.hmc.edu/su/fairdivision/calc/), a very interesting applet that does fair division of a homotopy algorithm.

Let V be the set of vertices of the triangulation. We find a labeling $\mathscr{L}$: $V \to \{0, 1, \ldots, d\}$ such that $\forall \mathbf{X}, \mathbf{Y} \in V$ on the same base cell, $\mathscr{L}(\mathbf{X}) \neq \mathscr{L}(\mathbf{Y})$. Then for any labeled vertex $\mathbf{X}$, we define a coloring $\mathscr{C}$: $V \to \{0, 1, \ldots, d\}$ such that $\mathscr{C}(\mathbf{X}) = i$ if player $\mathscr{L}(\mathbf{X})$ prefers the ith piece of the cut $\mathbf{X}$. By Sperner's lemma, the resulting triangulation has a fully colored base that corresponds to an approximate envy-free solution. This reduces the problem to finding a fully colored base in a Sperner simplex and therefore establishes that the problem is in PPAD.

On the other hand, we design a reduction based on the BROUWER problem (Brouwer 1910; Papadimitriou 1994; Chen and Deng 2008, 2009a, b) for its proof of PPAD-hardness.

1.4.2. Matching Lower and Upper Bound in the Oracle-Function Model. We derive a $\theta(N^{d-1})$ time matching bound for the query complexity of cake cutting for $d+1$ players with Lipschitz utilities. The tight upper bound requires a binary search method that finds a balanced cut of the simplex. It is made possible by Kuhn's triangulation and our labeling rule that allows an efficient parity checking of the index of the boundary simplices of $(d-1)$-dimension. Take the case of three players as an example: because two cuts are possible to obtain an envy-free cutting, we have two variables. Therefore, the problem can be decided on a two-dimensional space. We need to find a triangle of all three colors on its vertices on the Kuhn's triangulation over a two-dimensional space. To find one, we first divide the triangulated Sperner simplex (with a valid coloring of $\{0, 1, 2\}$) into two polygons P_1 and P_2. By the standard degree theory, the number of fully colored edges by $\{0, 1\}$ on the boundary has the same parity as the number of fully colored triangles by $\{0, 1, 2\}$ in the polygon. We then count the numbers of edges on the boundary that are colored by both 0 and 1 for both polygons and denote them as N_1 and N_2, respectively. This can be done in time $O(N)$. By Sperner's lemma, the number of triangles with all three colors in the original simplex is odd. Therefore, either N_1 or N_2 is odd. Without loss of generality, we assume N_1 is odd. Then there must be a fully colored triangle in P_1. The size of the problem is reduced by 1/2. By doing the above steps recursively, we can find one fully colored triangle in time $O(N) + O(N/2) + \cdots + O(1) = O(N)$. The above method can be generalized to high-dimensional problems. By using the same argument, we only need to examine the edges on the boundary, which reduces the complexity on the exponent from d to $d-1$.

For the lower bound, the results are obtained by a reduction from BROUWER (Deng et al. 2011).

1.4.3. FPTAS for Three Players with Monotone Utility Functions. For the special case of monotone utility functions, we are able to utilize the monotonicity to construct an ϵ-approximate envy-free solution in time polynomial in $\log(N)$ for three players.

We rely on the general approach of binary search on the parity but exploit the monotonicity of the best choice along certain lines in the simplex. We will sketch the main idea of the approach in the next two paragraphs and leave the details to the last section.

Any player with a monotone utility function would prefer A to B for $B \subseteq A$. Therefore, when one cut is fixed, one of the three pieces is fixed. Any player's preference on the other two pieces will change monotonically as another cut changes from left to right. We use the coordinate $\mathbf{X} = (X_0, X_1, X_2)$, along the line of fixed X_0, and let X_1 increase from 0 to $N - X_0$. The preference of a player's choice will start with the last piece, to the first piece, and to the middle piece (one or two of them may be missing). Similar monotonicity holds when X_2 is fixed. For each player, we can find the break point of the choices by binary search in $O(\log N)$ time.

Because of the monotonicity along those two directions, we can calculate the indices of cut edges efficiently. Therefore, the indices of the two regions split by the cut can be decided quickly. We will stay on the region with an odd index so that we should terminate with one that is a diamond-shaped polygon consisting of at most two base cells. By parity, one of the two cells is a fully colored base triangle. The overall query complexity and running time will be of $O(\log^2 N)$.

1.5. Organization

The rest of the paper is organized as follows. We first introduce the relevant concepts, the mathematical and computational models in §2.

In §3, we present the complexity results for finding an envy-free cake-cutting solution, with a matching algorithmic bound under the oracle-function model, as well as a PPAD-completeness proof for the polynomial function model.

In §4, we show that for three players there is an algorithm for finding an ϵ-approximate envy-free solution in time polynomial in $\log(N)$ under certain conditions. In comparison, we prove that Stromquist's solution demands an exponential number of queries.

Finally, we conclude our work with discussion and open problems in §5.

2. Definitions and Basic Concepts

In this section, we will briefly review the basic definitions and results in combinatorial topology and computational complexity that will be used in the proof.

2.1. Sperner's Lemma

In general, a d-simplex $\Delta^d(\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_d)$ is the convex hull of $d+1$ independent vectors $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_d$, i.e., $\Delta^d = \{\mathbf{v} \mid \mathbf{v} = \sum_{i=0}^{d} \alpha_i \mathbf{X}_i, \alpha_i \geqslant 0, \sum_{i=0}^{d} \alpha_i = 1\}$, where $(\alpha_0, \alpha_1, \ldots, \alpha_d)$ are the *barycentric coordinates* of $\mathbf{v}$. For $\mathbf{v} \in \Delta^d$, define $\chi(\mathbf{v}) = \{i \mid \alpha_i > 0\}$.

A face of a simplex $\Delta^d(\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_d)$ is any simplex $\Delta^k(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \ldots, \mathbf{X}_{i_k})$ where $0 \leqslant i_1 < i_2 < \cdots < i_k \leqslant d$.

Triangulation of a simplex is a procedure that partitions the original d-simplex Δ^d into several small d-simplices $\Delta_1, \ldots, \Delta_m$ such that for $\forall\, i, j$, $\Delta_i \cap \Delta_j$ is either empty or a common face.

Let V be the union of the vertices of all Δ_is. Sperner's lemma is defined on V, the vertices of a triangulated simplex with Sperner coloring.

Definition 1 (Sperner Coloring). For a triangulated simplex Δ with vertex set V, a coloring

$$\lambda\colon \mathbf{v} \to \{0, 1, \ldots, d\} \text{ is called Sperner coloring}$$

$$\text{if } \forall\, \mathbf{v} \in V\colon\ \lambda(\mathbf{v}) \in \chi(\mathbf{v}).$$

A Sperner colored triangulated simplex has a special property that is best explained by the concept of index.

Definition 2. For a d-dimensional simplex with vertices assigned all colors $\{0, 1, \ldots, d\}$, we define its index as 1. Otherwise, it is defined to be zero. For a triangulated polygon in d-space, its index is defined to be the sum of the indices of its base simplices modulo two.

For a polygon P, denote by ∂P the boundaries of P. Note that the triangulation of P induces a triangulation of ∂P into $(d-1)$-dimensional simplices. Naturally, we define $index_{d-1}(\partial P, \lambda) = \sum_{\delta_{d-1} \in \partial P} index_{d-1}(\delta_{d-1}, \lambda)$. Note that a $(d-1)$ dimension cell has index 1 if its vertices have all colors in $\{0, 1, \ldots, d-1\}$, 0 otherwise. The following result is useful (see, e.g., Todd 1976).

Proposition 1. $index(P, \phi) \equiv index_{d-1}(\partial P, \phi) \bmod 2$.

With the restriction of Sperner coloring, we immediately obtain the following:

Theorem 1 (Sperner's Lemma, 1928 (Sperner 1928)). *For a simplex that is triangulated and colored by Sperner coloring, the number of simplices in the subdivision with index* 1, *i.e., its vertices all colored differently, is odd.*

Sperner's lemma states that a triangulated simplex with a Sperner coloring has a base cell such that all its vertices have different colors. Such a simplex is commonly referred to as a Sperner Simplex.

2.2. The Model and Envy-Free Solutions

Following the cake-cutting literature, we represent the cake as a line segment $L = [0, 1]$. To model the preferences, we assign a utility function u_i defined over each line segment on L to each player $i \in I$. We may more generally define the utility function over the Borel space of the line segment $L = [0, 1]$. However, as we are considering a d-cut solution for $d+1$ players, a function defined on each subsegment of L would be enough for our study.

Our utility functions are required to satisfy the following two conditions:

• Nonnegativity condition: $u_i(\varnothing) = 0$ and $u_i(\neq \varnothing) > 0$.

• Lipschitz condition: For any pair of intervals A, $B \subseteq L$, where $A = [a_1, a_2]$ and $B = [b_1, b_2]$, $|u_i(A) - u_i(B)| \leqslant K \times \mu(A \ominus B)$ for a fixed constant K, where $\mu(\cdot)$ is the Borel measure and $A \ominus B = (A \backslash B) \cup (B \backslash A)$.

The first condition, which is also called the "hungry condition," is the main requirement in determining a continuous solution. The second condition opens up a possibility to discretize the function for an approximate solution.

A cut in the middle will be along the point $1/2$, which can be denoted by $(1/2, 1/2)$, meaning that the first part is the left half of the whole, and the second part is the right half of the whole. In general for a set $I = \{0, 1, \ldots, d\}$ of $d+1$ players, a cut into $d+1$ pieces through a d-cut can be represented by a $(d+1)$-component vector $(\alpha_0, \alpha_1, \ldots, \alpha_d)$ with the property that $\sum_{i=0}^{d} \alpha_i = 1$. Here again, the d-cut splits the cake, from left to right, pieces in fractions α_j, $j = 0, 1, \ldots, d$, of the whole.

By the nonnegativity condition of utility functions, every player will strictly prefer the nonzero segments to the zero segments. More formally, consider any boundary point $\mathbf{p}$ represented as $(\alpha_0, \alpha_1, \ldots, \alpha_d)$ such that $\alpha_j = 0$ but $\alpha_i > 0$. For the d-cut defined by $\mathbf{p}$, every player strictly prefers the ith segment to the jth.

It is easy to see that the set of all possible d-cuts form a d-dimensional simplex Δ^d. As a side note, consider any triangulation of such a simplex and color the vertices of the triangulation by the preferences of one of the players. Because of the nonnegativity property of the utility functions, the resulting coloring would be a Sperner coloring.

We may define an envy-free solution for the cake-cutting problem as follows:

Definition 3 (An Envy-Free Cut). A partition or a d-cut S is envy-free if different players prefer different segments of S, i.e., there is a permutation π: $I \to S$ such that for all $j \neq i$,

$$u_i(\pi(i)) \geqslant u_i(\pi(j)).$$

That is, for each player, its assigned piece of the cake is the best among all the $d+1$ pieces, according to its own utility function.

Theorem 2 (Stromquist 1980, Su 1999). *There is an envy-free cake-cutting solution for $d+1$ players that uses only d cuts.*

2.3. Simmons-Su's Method for Finding Approximate Solutions

Su (1999) explicitly described the first computational procedure for finding an approximate envy-free cake-cutting solution, by a labeling process on a triangulated simplex, citing Simmons as the one who first outlined these ideas using Sperner's Lemma (Simmons 1980).

Simmons-Su's method is based on a discretization and triangulation of the simplex corresponding to a d-cut. It first labels all vertices of the triangulated d-simplex with $d+1$ numbers such that each base d-simplex is labeled on its vertices with all $d+1$ numbers. We will refer it to as a $(d+1)$-labeling of the triangulated simplex.

Definition 4 (Simmons-Su Labeling). Given a triangulated simplex, a labeling of the vertices is a Simmons-Su labeling if, on each base simplex, every vertex is labeled differently.

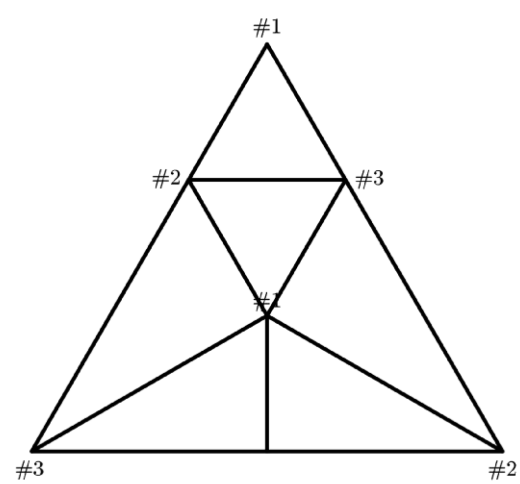
Recall that the Sperner's lemma says for any triangulated simplex with Sperner coloring, the number of the fully colored base cells is odd. If a triangulated simplex is labeled by a Simmons-Su labeling, then all base cells are labeled differently. Note that any triangulated simplex can be colored by Sperner coloring, but not every triangulation admits a Su's labeling. See Figure 1 as an example.

Definition 5 (Simmons-Su Simplex). Given a triangulation with a Sperner coloring and a Su's labeling, a base simplex is called a Su's simplex if it is fully labeled *and* colored, i.e., all its vertices are *labeled differently* and *colored differently*.

Clearly, a Simmons-Su labeling together with a Sperner coloring admit a Simmons-Su simplex by Sperner's lemma.

Theorem 3. *For a triangulated simplex that is labeled by Simmons-Su labeling and colored by Sperner coloring, there exists at least one Simmons-Su simplex.*

Figure 1. An example of triangulation not admitting any Simmons-Su labeling scheme.



This immediately leads to an approximate solution concept for cake cutting. For a vertex u of label i, it is colored by j if player i prefers the jth piece on cut u. The nonnegative property of the utility functions guarantees the colors are Sperner colors. Therefore, there exists a Su's simplex in the triangulation.

On a Simmons-Su simplex, the $d+1$ labels with $d+1$ different colors indicate that the optimal choices of $d+1$ players are different. Even though those preferences are expressed for different d-cuts, the corresponding d-cuts are close enough to form an approximate solution when the base simplices are sufficiently small.

By refining the triangulation, one can use the above argument to find a continuous fixed point and therefore, an envy-free solution.

Su (1999) used the barycentric triangulation of a simplex. At each level, a simplex is triangulated into $(d+1)!$ simplices recursively in dimensions as follows. For dimension $d=1$, a simplex is a line segment, and its barycentric triangulation is a partition of the line segment into two by inserting the center point of the line segment. A $d>1$ dimensional simplex has $d+1$ faces. Each face is partitioned into a $(d-1)$-dimensional barycentric triangulation (with a total of $d!$ cells of $(d-1)$-dimension). Each cell forms a d-dimensional simplex with the center of the original simplex of d-dimension. With a total of $d+1$ faces, there are a total of $(d+1)d! = (d+1)!$ simplices of d-dimension in this triangulation.

It continues in levels after levels until the desired granularity is reached. The barycentric triangulation can be labeled by a Simmons-Su labeling and therefore it can prove the existence of the envy-free solution. However, from a complexity perspective, barycentric triangulations are problematic because they create simplices with large aspect-ratios that make the process converge rather slowly. The distance between two points in the same base simplex in the barycentric triangulation will remain relatively long with respect to the level of triangulation.

In §3, we will adapt the Simmons-Su approach for Kuhn's triangulations. By giving polynomial-time algorithms for producing the triangulation, and showing that a Simmons-Su labeling is possible for Kuhn's triangulation, we come up with hardness proofs both for the polynomial-function model and the oracle model. We will define these complexity models in the next subsection.

2.4. Complexity Models and Classes

We will study the complexity of the envy-free cake-cutting problem for two types of models: the oracle-function model and the polynomial-time function model.

In the oracle-function model, we are interested in the minimum number of times the algorithm has to query the players about their preferences. A query is defined as the following operation: the algorithm presents a d-cut to a player and asks for the index (or indices) of the most desirable piece(s) in the cut.

If we assume that utility functions are arbitrary and continuous, it may be impossible to present the envy-free solution with a finite number of digits. Because of that, we introduce the concept of approximate envy-free solutions:

Definition 6 (An ϵ-Approximate Envy-Free Cutting). A partition is ϵ-envy-free if different players ϵ-approximately prefer different segments in S, i.e., there is a permutation $\pi\colon I \to S$ such that for all $j \neq i$,

$$u_i(\pi(i)) + \epsilon \geqslant u_i(\pi(j)).$$

Therefore, the interesting question becomes how many queries or operations are needed for obtaining an ϵ-approximate solution. The polynomial-time function model captures the latter criterion. In this model, we assume that the utilities of the players are given explicitly as functions or polynomial-time algorithms. Then we are interested to see if there is a polynomial-time algorithm in the number of players and the number of bits of the error (i.e., $\log(1/\epsilon)$) for finding an ϵ-approximate envy-free cut.

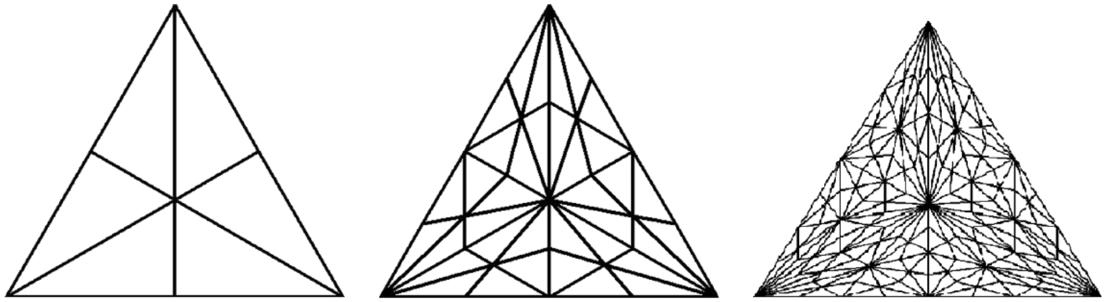
Definition 7 (PTAS and FPTAS). Given a problem G, an algorithm $\mathscr{A}$ is called a polynomial-time approximate scheme (PTAS for short) if, for any $\epsilon > 0$, it finds an ϵ-approximate solution in polynomial time in the original input size and $1/\epsilon$. It is further called an FPTAS if it is polynomial in the original input size and $\log 1/\epsilon$.

2.4.1. Relevant Complexity Classes. Our results for the oracle-function model are unconditional: we give explicit (and tight) bounds on the number of queries needed to compute an approximate envy-free allocation as a function of number of players and the error. However, in the polynomial-time function model, our results will build on the extensive literature in computational complexity and consists of proving inclusion or completion for certain complexity classes.

Problems such as finding envy-free cake cutting or approximate envy-free cake cutting in our discussion are always guaranteed to have a solution and they are defined to belong to the class *TFNP*, defined by Megiddo and Papadimitriou (1991). More formally, a binary relation $F(\cdot,\cdot)$ is in the class *TFNP* if, for any x, there exists a y, $|y|$ bounded by a polynomial in $|x|$ such that $P(x,y)$ holds, and $P(x,y)$ can be verified to hold in polynomial time. The problem is to find out such a y in polynomial time.

An interesting subclass of *TFNP*, the class *PPAD*, is based on an exponential size graph, consisting of directed paths and cycles. For each instance of the problem, one starting vertex is given for a directed path. We are required to find out the end vertex of any directed path or the starting vertex of another directed path different from the given one. Here such a solution is guaranteed to exist because the graph consists of directed paths and cycles, with at least a direct path (for the given starting vertex). Note that as a further requirement to *PPAD*, given a vertex, the incoming edge and outgoing edge can be computed by a given polynomial-time algorithm. Such a model is called the polynomial-time function model.

Figure 2. Barycentric triangulation on a two-dimensional triangle.



(a) 1st iteration (b) 2nd iteration (c) 3rd iteration

Alternatively, in the oracle model the function values are returned by an oracle function, which, given input x, outputs the function value $f(x)$ in one oracle time. Note that polynomial-time algorithms are a small faction of all the algorithms. Therefore, the oracle function will cover more functions than the polynomial-time model. One unit of oracle time may in general translate into much longer computer time if it is fully implemented. We also define the query complexity of an algorithm to be the number of times a function value is queried in the oracle model. Our discussion in the oracle model will focus on the query complexity.

3. Complexity Analysis for Discrete Cake Cutting

For the purpose of complexity analysis, let us introduce a definition for a discrete envy-free cake-cut for $d+1$ players using d cuts.

For the discrete version, we will use X_i, $i=0,1,\ldots,d$ for the ith barycentric coordinate of a cut in its integer representation. That is, we partition the cake (as a line $[0,1]$) into N equal segments. In the following parts, the cake is treated as $[0,N]$ and only integral cut $\mathbf{X}$ is considered. Therefore, $X_i \in \{0,1,\ldots,N\}$, as opposed to $0 \leqslant \alpha_i \leqslant 1$ in the previous discussion.

We extend a utility function to the barycentric coordinate as follows:

Definition 8. Given a d-cut represented by the barycentric coordinate $\mathbf{X}$, let $c_j(\mathbf{X}) = (1/N)\sum_{i=0}^{j} X_i$, $j=0,1,\ldots,d$ with the convention that $c_{-1}(\mathbf{X})=0$. Let the corresponding intervals be $I_j(\mathbf{X}) = [c_{j-1}(\mathbf{X}),\ c_j(\mathbf{X})]$, $j=0,1,\ldots,d$. We extend a utility function u on the intervals to the barycentric coordinate: $u(\mathbf{X}) = \max\{u(I_j(\mathbf{X})), j=0,1,\ldots,d\}$.

Two d-cuts $\mathbf{X}$ and $\mathbf{Y}$ are adjacent to each other if $\forall\, i \in \{0,1,\ldots,d\}$: $|X_i - Y_i| \leqslant 1$.

Definition 9 (Discrete Cut). A discrete cake cut is defined to be a set $\{\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(d)}\}$ of $d+1$ d-cuts such that for each pair of j and k, the two d-cuts $\mathbf{X}^{(j)}$ and $\mathbf{X}^{(k)}$ are adjacent.

When we choose finer and finer scale with larger and larger N, the *discrete cut* structure becomes finer and finer (their nodes become closer and closer to each other). The discrete cut here is defined to be a set of cuts that are adjacent to each other, i.e., any two cuts are very close to each other. This requirement is the essential point for the complexity analysis of the approximate cake cutting, which overcomes the problem of large aspect-ratio in Su's barycentric triangulation.

Definition 10 (Discrete Envy-Free Cake Cut). A discrete cake cut is an envy-free solution if there is a permutation π of $\{0,1,\ldots,d\}$ such that player i prefers the $\boldsymbol{\pi}(i)$th segment for some d-cut $\mathbf{X}^{(j)}$ in the set. We denote it by $P_i(\mathbf{X}^{(j)}) = \boldsymbol{\pi}(i)$.

Intuitively, in the limit point along a convergent sequence, the preference of players will be on different segments of the limit point, representing a d-cut, and therefore results in an envy-free solution. The convergence will be guaranteed when the utility functions are Lipschitz.

Corollary 1. *If utilities are Lipschitz with the Lipschitz constant K, then any cut in the DISCRETE ENVY-FREE CAKE CUT set is an ϵ-approximate envy-free solution for $\epsilon = O(1/N)$.*

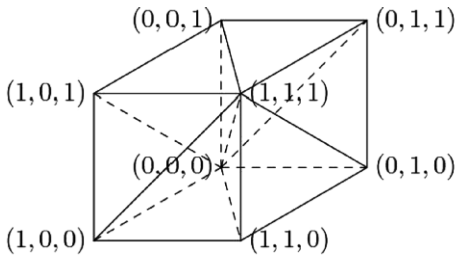
Proof. Assume $X, Y \in$ DISCRETE ENVY-FREE CAKE CUT set. Note that $I_j(\mathbf{X}) = [c_{j-1}(\mathbf{X}), c_j(\mathbf{X})]$, we have for any j, $|u(I_j(\mathbf{X})) - u(I_j(\mathbf{Y}))| \leqslant K \times \mu(I_j(\mathbf{X}) \ominus I_j(\mathbf{Y})) \leqslant K \times (2/N)$ by Definition 9. The statement follows.

The rest of this section is organized as follows. We start by introducing the Kuhn's triangulation. We will show that Kuhn's triangulation is balanced and admits Simmons-Su labeling, i.e., a DISCRETE ENVY-FREE CAKE CUT corresponds to an ϵ-approximate envy-free solution for a small ϵ. Then we derive the complexity of finding a DISCRETE ENVY-FREE CAKE CUT for Kuhn's triangulations in both oracle and polynomial-time function models.

3.1. Kuhn's Triangulation

Kuhn's triangulation was first defined on a unit hypercube in d dimensions. It partitions the d-cube into $d!$ simplices. Specifically, let $\mathbf{V}^0 = (0,0,\ldots,0)_{1\times d}$ be one corner

Figure 3. An example of Kuhn's triangulation on a unit cube in three dimension.



of the cube. The diagonal vertex to it would be $\mathbf{V}^{d+1} = (1, 1, \ldots, 1)_{1\times d}$. Let $\pi := (\pi(1), \pi(2), \ldots, \pi(d))$ be any permutations of the integers $1, 2, \ldots, d$. Each permutation π corresponds to one base simplex Δ_π^d whose vertices are given by $\mathbf{v}_\pi^i = \mathbf{v}_\pi^{i-1} + \mathbf{e}_{\pi(i)}$, where $\mathbf{v}_\pi^0 = \mathbf{V}^0$ and $\mathbf{e}_i$ is a unit vector with ith component being 1. These simplices all have disjoint interiors, and their union is the d-cube. Therefore, the unit hypercube is triangulated to $d!$ base simplices.

For example, consider a unit 3-cube. Let $(0, 0, 0)$ be the base point. By Kuhn's triangulation, the unit cube is partitioned into six tetrahedrons according to the different permutations of 1, 2, 3. See Figure 3.

$$\pi^1 = (1,2,3)\colon\ \Delta_{\pi^1} = \{(0,0,0), (1,0,0), (1,1,0), (1,1,1)\};$$
$$\pi^2 = (1,3,2)\colon\ \Delta_{\pi^2} = \{(0,0,0), (1,0,0), (1,0,1), (1,1,1)\};$$
$$\pi^3 = (2,1,3)\colon\ \Delta_{\pi^3} = \{(0,0,0), (0,1,0), (1,1,0), (1,1,1)\};$$
$$\pi^4 = (2,3,1)\colon\ \Delta_{\pi^4} = \{(0,0,0), (0,1,0), (0,1,1), (1,1,1)\};$$
$$\pi^5 = (3,1,2)\colon\ \Delta_{\pi^5} = \{(0,0,0), (0,0,1), (1,0,1), (1,1,1)\};$$
$$\pi^6 = (3,2,1)\colon\ \Delta_{\pi^6} = \{(0,0,0), (0,0,1), (0,1,1), (1,1,1)\}.$$

Next, let us extend Kuhn's triangulation to a simplex. Let N be an integer bigger than 1. Take a unit d-cube and use parallel cuts of equal distance to partition it into N^d smaller d-cubes of side length $1/N$. Then, partition each small cube into $d!$ simplices by Kuhn's triangulation. Now, observe that the unit cube can also be partitioned into $d!$ big simplices first, and each big simplex contains N^d smaller simplices or base cells. The proof of the consistency of the two processes is discussed as follows.

We scale the unit cube into one of side length N. We call it the big cube, and each of the N^d subcubes the small cubes.

It is enough to show that for a simplex in the refined grid starting from any grid point $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_d^*)$ to $\mathbf{y}^* = (x_1^*+1, x_2^*+1, \ldots, x_d^*+1)$, defined by a permutation γ, all of its $d+1$ vertices are in the same big simplex derived by some permutation ρ with base point $(0, 0, \ldots, 0)$ to the point $(N, N, \ldots, N)$.

Let $\mathbf{x}^0 = \mathbf{x}^*$, and $\mathbf{x}^i$ is the same as $\mathbf{x}^{i-1}$ except for its $\gamma(i)$th coordinate $x_{\gamma(i)}^i$ which is $x_{\gamma(i)}^{i-1} + 1$. Then we have

1. If $x_i^* > x_j^*$ (which equivalent to $x_i^* \geqslant x_j^* + 1$), $x_i^k \geqslant x_j^k$ for all $k = 1, 2, \ldots, d$.

2. If $x_i^* = x_j^*$ and $\gamma^{-1}(i) < \gamma^{-1}(j)$, $x_i^k \geqslant x_j^k$ for all $k = 1, 2, \ldots, d$.

Hence, there is a permutation ρ such that $x_{\rho(1)}^k \geqslant x_{\rho(2)}^k \geqslant \cdots \geqslant x_{\rho(d)}^k$ for all $k = 0, 1, 2, \ldots, d$, which guarantees the base simplex inside the large simplex derived by permutation ρ with base point $(0, 0, \ldots, 0)$ to the point $(N, N, \ldots, N)$.

Based on the equivalency of the two partitioning processes, we choose one of the big simplices, for example the one corresponding to $\pi = (1, 2, \ldots, d)$, named Δ^d, and triangulate it by Kuhn's triangulation. Note that a vertex $\mathbf{x} \in \Delta^d$ can also be represented by barycentric coordinates $(\alpha_0, \alpha_1, \ldots, \alpha_d)$. For simplicity of presentation, we use the integers $\mathbf{X} = (X_0, X_1, \ldots, X_d) = (N\alpha_0, N\alpha_1, \ldots, N\alpha_d)$ to be a base point. Here we have $\sum_{i=0}^d X_i = N$. Set $\mathbf{a}_i = (a_{i0}, a_{i1}, \ldots, a_{id})$ where $a_{ij} = 0$ for all $i \neq j-1, j$ and $a_{ii} = -1$, $a_{i,\,i+1} = 1$ for $\forall\, i \in \{0, \ldots, d-1\}$. Then the vertices of the base simplex according to permutation π based on $\mathbf{X}$ are given by $\mathbf{v}_\pi^i = \mathbf{v}_\pi^{i-1} + \mathbf{a}_{\pi(i)-1}$ for $\forall\, i \in \{1, \ldots, d\}$ and $\mathbf{v}_\pi^0 = \mathbf{X}$. For example, consider a simplex with $N = 1$ based on vertex $\mathbf{x} = (0, 0, \ldots, 0)_d$ and permutation $(1, 2, \ldots, d)$. We first set $\mathbf{v}^0 = (1, 0, \ldots, 0)_{d+1}$, then by the above transformation $\mathbf{v}^1 = (0, 1, \ldots, 0)_{d+1}$, $\mathbf{v}^2 = (0, 0, 1, \ldots, 0)_{d+1}$. Please refer to Scarf (1977, p. 42) for more details.

The main advantage of Kuhn's triangulation is that, unlike the barycentric triangulation considered by Su, it is balanced. In other words, the corner points of every base cell in this triangulation are close to each other.

Lemma 1. *For given* $\mathbf{X} = (X_0, X_1, \ldots, X_d)$ *and* $\mathbf{Y} = (Y_0, Y_1, \ldots, Y_d)$, *define*

$$\delta_{\mathbf{X}-\mathbf{Y}} = \max_{\forall i \in \{0,\,1,\,\ldots,\,d\}} \{|X_i - Y_i|\}.$$

If $\mathbf{X}$ *and* $\mathbf{Y}$ *are in the same base cell in Kuhn's triangulation, then* $\delta_{\mathbf{X}-\mathbf{Y}} = 1$.

Proof. Without loss of generality, assume the base cell corresponds to a permutation π and $\mathbf{Y} = \mathbf{X} + \sum_{i=l}^k \mathbf{a}_{\pi(i)-1}$ for some $0 \leqslant l \leqslant k \leqslant d$. Fix i, because $a_{ii} = -1$, $a_{i,\,i+1} = 1$ are the only nonzero coordinates in a_i, and let $\mathbf{b} = (b_0, b_1, \ldots, b_d) = \sum_{i=l}^k \mathbf{a}_{\pi(i)-1}$, then $\forall\, i$, $b_i \in \{-1, 0, 1\}$. Therefore, $\delta_{\mathbf{X}-\mathbf{Y}}$ equals to either 0 or 1 and because $\mathbf{X}$, $\mathbf{Y}$ are two different vertices, we must have $\delta_{\mathbf{X}-\mathbf{Y}} = 1$.

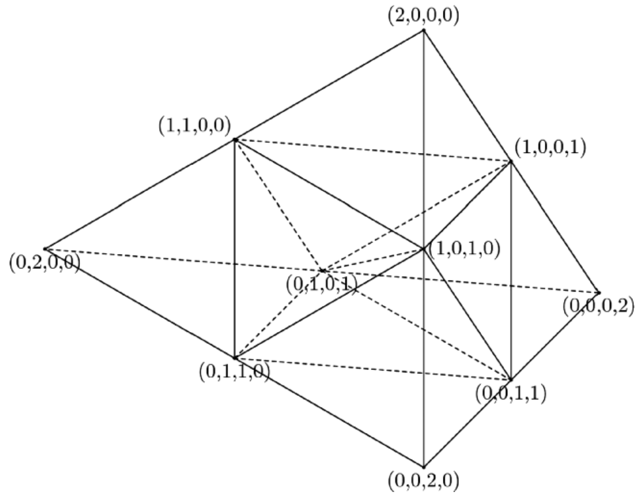
Lemma 1 implies that the vertices of any base cell of Kuhn's triangulation are adjacent to each other and thus:

Corollary 2. *Any base cell of Kuhn's triangulation forms a discrete cut.*

3.2. Simmons-Su Simplex in Kuhn's Triangulation

We can now present an algorithm for finding a DISCRETE ENVY-FREE CAKE CUT in a Kuhn's triangulation. We will crucially use the fact that Kuhn's triangulation admits a Simmons-Su labeling. The proof will be given later in this section.

Figure 4. Kuhn's triangulation for a simplex with $N = 2$.



For the simplicity of exposition, consider the problem for four players: A, B, C, D. In this case, the closed set of all possible cuts is a simplex of side length 1. To obtain an approximate discrete envy-free cut, we partition the simplex by Kuhn's triangulation with $N = 2$ and obtain $N^d = 8$ base simplices (see Figure 4).

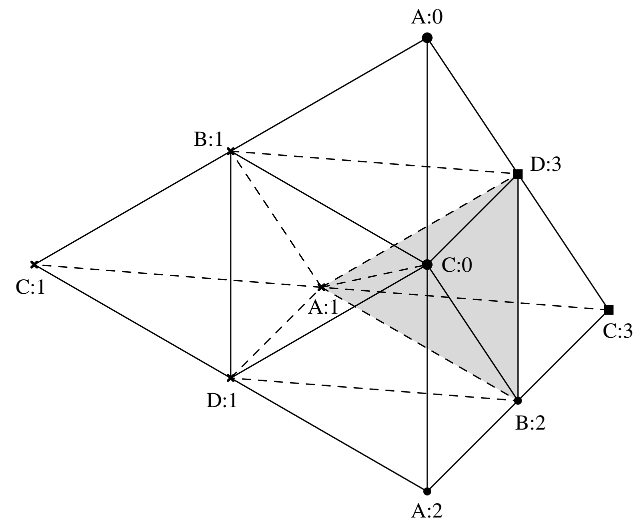
Let V be the set of vertices of all base cells, i.e., $V = \{(X_0, X_1, X_2, X_3): X_0, X_1, X_2, X_3 \in \mathbb{Z}_+, X_0 + X_1 + X_2 + X_3 = 2\}$, where $\mathbb{Z}_+$ is the nonnegative integer set.

The algorithm has two stages: labeling and coloring. We partition V into four control subsets V_0, V_1, V_2, V_3, each corresponding to one player. Starting by assigning $(2, 0, 0, 0)$ to V_0, $(1, 1, 0, 0)$ to V_1, $(1, 0, 1, 0)$ to V_2, and $(1, 0, 0, 1)$ to V_3. The rest of V is partitioned in such a way that the four vertices of each base simplex belong to different subsets V_ts.

Next, we should color each vertex in V. For a vertex $\mathbf{X}$ labeled by t, we let player t choose, among four segments, $[0, X_0/N]$, $[X_0/N, (X_0 + X_1)/N]$, $[(X_0 + X_1)/N, (X_0 + X_1 + X_2)/N]$ and $[(X_0 + X_1 + X_2)/N, 1]$ of I, one that maximizes its utility. For simplicity of presentation, we should assume a nondegenerate condition that the choice is unique. The general case can be handled with a careful tie-breaking rule. If its optimal segment is the one of length X_s/N, $0 \leqslant s \leqslant 3$, we assign color s to the vertex $\mathbf{X} = (X_0, X_1, X_2, X_3)$. See Figure 5 for details of the example, where A: 0 means player A prefers the first piece at that cut.

Because of the nonnegativity property of the utility functions, the above coloring is a valid Sperner coloring. The four vertices of each base simplex belong to four different subsets, or we say four different players. If we find a fully colored base simplex, then on the four vertices of this simplex, different players prefer different pieces (see the shadowed base simplex in Figure 5). It results in a Simmons-Su Simplex, which is a DISCRETE ENVY-FREE CAKE CUT under our definition.

Figure 5. Four-players envy-free cake cutting with Kuhn's triangulation.



The case of $d + 1$ players for any d can be handled in a similar way if there exists a valid Simmons-Su labeling for the triangulated d-simplex. We start by proving that Kuhn's triangulation does admit a Simmons-Su labeling for all dimensions.

Claim 1. *The Kuhn's triangulation of a simplex admits a Simmons-Su labeling.*

Proof. For a Kuhn's triangulated simplex Δ^d with vertex set V, a weight function W is defined as:

$$W(\mathbf{X}) = \sum_{i=0}^{d} iX_i,$$

where $\mathbf{X} = (X_0, X_1, \ldots, X_d) \in V$.

Now, consider a labeling rule $\mathscr{L}$:

$$\mathscr{L}(\mathbf{X}) = V_t, \quad \text{if } W(\mathbf{X}) \equiv t (\operatorname{mod} d + 1).$$

Consider a base simplex Δ with base point $\mathbf{X}$ and permutation π.

Recall that the vertices of Δ can be represented as $\mathbf{v}_\pi^i = \mathbf{v}_\pi^{i-1} + \mathbf{a}_{\pi(i)-1}$ for $\forall\, i \in \{1, \ldots, d\}$, where $a_{ij} = 0$ for all $i \neq j-1, j$ and $a_{ii} = -1$, $a_{i,\,i+1} = 1$ for $\forall\, i \in \{0, \ldots, d-1\}$ and $\mathbf{v}_\pi^0 = \mathbf{X}$. Then,

$$\begin{aligned} W(\mathbf{v}_\pi^i) &= W(\mathbf{v}_\pi^{i-1}) + \big(((\pi(i) - 1) + 1) - (\pi(i) - 1)\big) \\ &= W(\mathbf{v}_\pi^{i-1}) + 1. \end{aligned}$$

Therefore, all vertices of Δ are labeled differently.

3.3. Complexity in the Polynomial Function Model

The problem of finding a Sperner Simplex (a fully colored base cell) in a Sperner coloring simplex is called SPERNER.

Lemma 2 (Papadimitriou 1994). *SPERNER is PPAD-complete.*

Labeling of every vertex of the Kuhn's triangulation can be done in polynomial time, and so is the coloring, because the utility functions are given by a polynomial-time algorithm according to the polynomial-function model. Each base cell can be constructed and their colors can be verified in polynomial time. In addition, all $(d+1)$ vertices of every base cell in Kuhn's triangulation have different labels. A fully colored base cell in this triangulation will be a solution to the envy-free cake-cutting solution. Therefore, the problem reduces to SPERNER. Hence, we have the following lemma:

Lemma 3. *Finding a discrete d-cut set for envy-free cake-cutting problem for $d+1$ people is in PPAD.*

In the rest of the section, we will prove that the envy-free cake-cutting problem is in fact PPAD-complete. This is done by applying a reduction from Brouwer's fixed-point theorem (Daskalakis et al. 2006, Chen and Deng 2009a). Let us start by defining the problem.

Consider a d-dimensional hypergrid of side length N and denote its node set by:

$$V_N^d = \{\mathbf{p} = (p_1, p_2, \ldots, p_d) \in \mathbb{Z}^d \mid \forall\, i\colon 0 \leqslant p_i \leqslant N\}.$$

Its boundary B_N^d consists of point $\mathbf{p} \in V_N^d$ with $p_i \in \{0, N\}$ for some i:

$$B_N^d = \{\mathbf{p} \in V_N^d\colon p_i \in \{0, N\} \text{ for some } i\}.$$

For each $\mathbf{p} \in \mathbb{Z}^d$, let

$$\mathcal{K}_{\mathbf{p}} = \{\mathbf{q}\colon q_i \in \{p_i, p_i + 1\}\}.$$

Define $m(\mathbf{p}) = 0$ if $\forall\, i, p_i > 0$; and $m(\mathbf{p}) = \min\{i\colon p_i = 0\}$, otherwise. A coloring function $g\colon V_N^d \Rightarrow \{0, 1, \ldots, d\}$ is valid if $g(\mathbf{p}) = m(\mathbf{p})$, $\forall\, \mathbf{p} \in B_N^d$.

Definition 11 (BROUWER). Let g be a valid coloring on V_N^d. Given an algorithm that outputs $g(p)$ for every $p \in V_N^d$, find a point $\mathbf{p} \in V_N^d$ such that $\mathcal{K}_{\mathbf{p}}$ is fully colored, that is, $\mathcal{K}_{\mathbf{p}}$ has all $d+1$ colors.

Lemma 4 (Daskalakis et al. 2006, Chen and Deng 2009a). *BROUWER is PPAD-complete.*

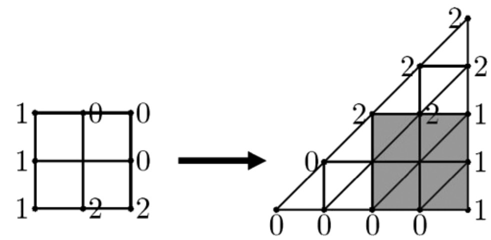
We are now ready to prove the theorem.

Theorem 4. *Finding an approximate solution for envy-free cake-cutting for $d+1$ people with d cuts is PPAD-complete.*

Proof. Lemma 3 shows that the problem is in PPAD. We will prove the PPAD-hardness through a reduction from BROUWER.

Let us start with the two-dimension case. Given an input of 2D BROUWER, a grid (G, V_N^2), and a coloring function $g\colon N \times N \to \{0, 1, 2\}$, we embed it into a

Figure 6. 2D reduction.



Kuhn's triangulated 2-simplex defined by three vertices: $\langle (0,0), (2N, 0), (2N, 2N) \rangle$ by the mapping: $T(x, y) = (2N - x, N - y)$.

We define the preference functions for $i = 0, 1, 2$ as follows: $P_i(2N - x, N - y) = g(x, y)$ for $0 \leqslant x, y \leqslant N$; $P_i(x, y) = 0$ for $0 \leqslant x < N$; $P_i(x, y) = 2$ for $N < y \leqslant 2N$. Therefore, the preference functions form a DISCRETE ENVY-FREE CAKE-CUT problem. From the origin, along the boundary in counterclockwise direction, we will walk through three pieces of consecutive vertices of all 0, 1, and 2 back to the origin. See Figure 6.

Therefore, there is a Sperner base cell in the triangulated Kuhn's triangle $\langle (0, 0); (2N, 0); (2N, 2N) \rangle$, which is at the same time a ENVY-FREE CAKE CUT by our choices of preference functions. Therefore ENVY-FREE CAKE CUT does have a solution. Once we find one, it must be in the region bounded by $N \leqslant x \leqslant 2N$, $0 \leqslant y \leqslant N$. The inverse mapping of T will give us the required BROUWER's solution.

The above argument generalizes to higher dimensions as follows. Let $\mathbf{e}_i^d$ denote the i-coordinate unit vector in d-dimensions. Let $\mathbf{0}^d$ be a d-dimensional vector of all 0s. Define $\mathbf{X}_i^d = N(\mathbf{e}_1^d + \mathbf{e}_2^d + \cdots + \mathbf{e}_i^d)$ for $i = 1, 2, \ldots, d$, and $\mathbf{X}_0^d = \mathbf{0}^d$. Consider a Kuhn's triangle K^d with vertices generated by the identity permutation. Then K^d is the convex hull of $\mathbf{X}_0^d, \mathbf{X}_1^d, \ldots, \mathbf{X}_d^d$.

Now, we need to show that a d-dimensional BROUWER H^d can be embedded into K^d, and the coloring of a valid H^d can be extended to a Sperner coloring of vertices in the Kuhn's triangulated simplex. This is done by the following two claims, which are both proved by induction.

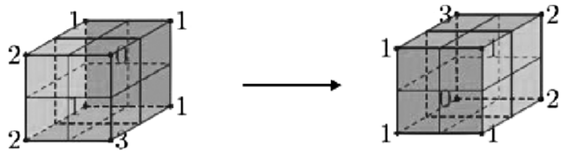
In the following discussion, we use the convention that, for a set S, $tS = \{tx\colon x \in S\}$ for any number t and $S \subseteq R^n$.

Claim 2. *A hypercube H^d of side length $N/2^{d-1}$ can be completely embedded into a Kuhn's triangulated simplex K^d.*

Proof of Claim 2. We prove it by induction. When $d = 1$, the line segment $[0, N]$ itself is a hypercube of side length N in dimension 1. Therefore, the claim holds.

Suppose the claim holds for d and consider $K^{d+1} = \{\mathbf{X}_0^{d+1}; \mathbf{X}_1^{d+1}; \ldots; \mathbf{X}_d^{d+1}; \mathbf{X}_{d+1}^{d+1}\}$ in $(d+1)$ dimensions. Note that K^{d+1} is the convex combination of the hyperplane $(K^d, 0)$ and the point X_{d+1}^{d+1}. By the induction hypothesis, there is a hypercube $(H^d, 0)$ of side length $N/2^{d-1}$

Figure 7. Step I: Color reorienting.



in $(K^d, 0)$. Then the convex combination $(1/2)\mathbf{X}_{d+1}^{d+1} + 1/2(H^d, 0)$ denoted by H_1 is a hypercube of side length $N/(2*2^{d-1})$, which is also contained in the simplex K^{d+1}. We can rewrite H_1 as

$$H_1 = \tfrac{1}{2}(\mathbf{X}_d^d, N) + \tfrac{1}{2}(H^d, 0) = \tfrac{1}{2}(\mathbf{X}_d^d + H^d, N) \subset K^{d+1}.$$

On the other hand, because $(H^d, 0) \subset (K^d, 0)$ and $(\mathbf{X}_d^d, 0) \in (K^d, 0)$, the convex combination

$$H_3 = \tfrac{1}{2}(H^d, 0) + \tfrac{1}{2}(\mathbf{X}_d^d, 0) = \tfrac{1}{2}(\mathbf{X}_d^d + H^d, 0) \subset (K^d, 0) \subset K^{d+1}.$$

Note that H_3 is also a hypercube of side length $N/(2*2^{d-1})$.

Let $H_2 = (1 - 1/2^{d-1})H_1 + (1/2^{d-1})H_3 = (1/2)(\mathbf{X}_d^d + H^d, (1 - 1/2^{d-1})N)$. Then $H_2 \in K^{d+1}$. Let

$$H^{d+1} = \alpha H_1 + (1-\alpha)H_2, \quad 0 \leqslant \alpha \leqslant 1.$$

Then $H^{d+1} \subset K^{d+1}$, and it is a hypercube of side length $N/(2*2^{d-1})$.

The claim follows.

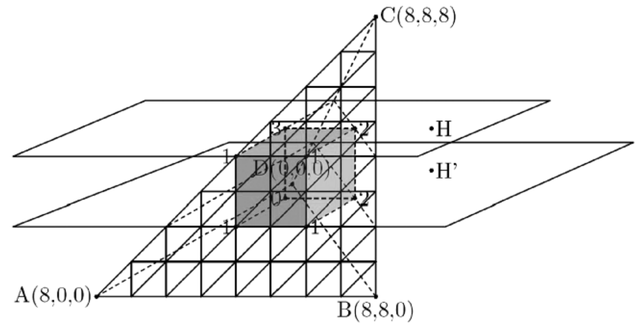
To complete the reduction, we should extend the colors of the hypergrid H^d to a Sperner coloring of vertices in the Kuhn's triangulated simplex K^d.

Claim 3. *It is possible to extend the colors of an embedded BROUWER hypergrid H^d to a Sperner coloring Kuhn's triangulated simplex K^d such that there is no fully colored base cell outside H^d.*

Proof of Claim 3. We consider the problem for the d-dimension ($d \geqslant 2$) case and apply induction to reduce it to the $d-1$-dimensional problem.

Recall in the BROUWER coloring, a node on a boundary face at $x_i = 0$ is colored i and on its opposite face is colored 0. According to the above rule, the color of a node on the boundary of a boundary face would be chosen from more than one color, but its final color is determined by the order of colors: $1 \prec 2 \prec 3 \prec \cdots \prec d \prec 0$. That is, if $i \prec j$ and they are colored on the same node, only i is the color placed at that node. All colors have priority over 0. For simplicity of the proof, we first reorient the hypercube and its colors: changing x to $N - x$ (see Figure 7). Therefore, now a node on a boundary face at $x_i = 0$ is colored 0.

Figure 8. Step II: Embedding and partition.



A boundary face is called N_i if its interior nodes are colored $i \neq 0$. A boundary face is called Z_i if its interior nodes are colored 0 while its opposite face is colored i.

Recall that a Kuhn's simplex K^d is the convex hull of $\mathbf{X}_0^d, \mathbf{X}_1^d, \ldots, \mathbf{X}_d^d$. By the proof of Claim 2, we embed the hypercube H^d with side length n into K^d by placing the face N_d on the hyperplane H: $x_d = h$ and the opposite face Z_d on the hyperplane H': $x_d = h' = h - n$. Note that N_d, Z_d are parallel to K^{d-1}. We color vertices $\mathbf{x} \in K^d \cap H^d$ with their original colors in H^d. (See Figure 8.)

We will expand the colors on the boundary of H^d to all the remaining vertices in the K^d as follows. First we cut K^d into three parts by two parallel hyperplanes H, H'.

For a vertex $\mathbf{x} \in K^d$,

1. $\mathbf{x} \in$ Part I, if $x_d > h$.
2. $\mathbf{x} \in$ Part II, if $h \geqslant x_d \geqslant h'$.
3. $\mathbf{x} \in$ Part III, if $x_d \leqslant h'$.

All vertices in Part I are colored d.

For vertices in Part II, we color them inductively on the hyperplane parallel to K^{d-1} that contains them. On the hyperplane H', the colors of vertices will be determined by the $(d-1)$-dimensional solution obtained in recursion by the embedding of Z_d, which is a BROUWER hypercube in $d-1$-dimension, into $(d-1)$-dimensional simplex. The vertices $\mathbf{x}$ with $h > x_d > h'$ can be colored in a similar way. For the layer H, it corresponds to the embedding of N_d into the $d-1$-dimension simplex. The color d on the boundary of N_d plays a role similar to 0 in the lower-dimensional case; therefore, vertices on this layer can also be colored recursively.

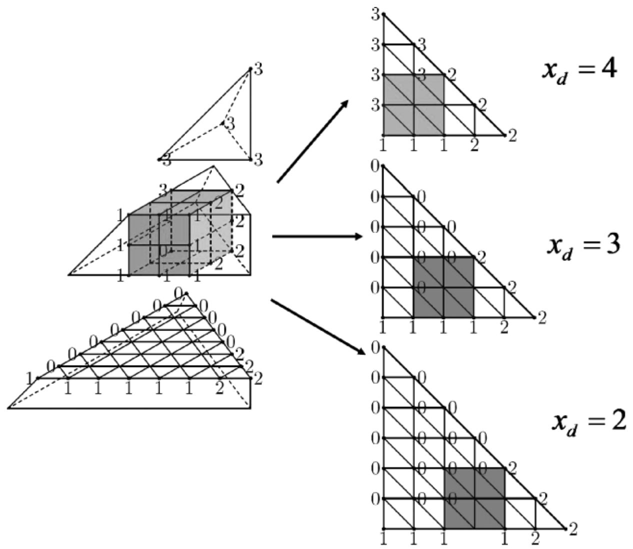
For vertices in Part III, the colors on the boundary of the $d-1$-dimension simplex on the hyperplane $0 \leqslant x_d < h'$ follow the colors on the boundary of the layer $x_d = h'$. All the remaining vertices in Part III are colored 0. (See Figure 9.)

When $d = 1$, we simply extend the colors of the extremes points of H^1 away from the BROUWER line segment to other vertices of K^1 and obtain a Sperner coloring.

Next we prove that the above coloring rule is a proper Sperner coloring and does not add any new fully colored base cell outside H^d.

In general, at dimension d, the boundary face Z_d of H^d is a $(d-1)$-dimensional BROUWER hypercube, and none of

Figure 9. Step III: Coloring.



its vertices are colored d. Similarly, the opposite boundary face N_d has no color 0. By using hyperplanes parallel to Z_d between Z_d and N_d to cut H^d, the boundaries of all the obtained $(d-1)$-dimensional hypercube have the same colors as the boundary of Z_d.

By inductive hypothesis, the vertices of the $d-1$-dimension simplex on any hyperplane between H and H' are SPERNER colored. The boundary vertices of $x_d = 0$ are colored the same as the boundary vertices on $x_d = h'$, therefore, $\mathbf{X}_i^d$ is colored i for $0 \leqslant i \leqslant d-1$. The vertex $\mathbf{X}_d^d$ is a vertex in Part I that is colored d.

To prove that none of the base cubes outside H^d has all the colors, we note that outside H^d vertices with color d only exist in Part I and on the hyperplane H. On the other hand, all the vertices in Part I are colored d, and there is no color 0 on the hyperplane H. Therefore, the claim follows.

Thus, we reduce a given BROUWER instance $(G, V(H))$ to one of finding Sperner's simplex in Kuhn's triangulated simplex. Because Kuhn's triangulated simplex is $(d+1)$-labeled, this gives us a Simmons-Su simplex.

By assigning each player the same preference function as the given Sperner coloring for a Sperner input, we reduce BROUWER to ENVY-FREE CAKE CUTTING.

3.4. Complexity in the Oracle Function Model

Similarly, under the oracle model for utility functions, we should show ENVY-FREE CAKE CUTTING has the same complexity as BROUWER.

THEOREM 5. *Solving the ENVY-FREE CAKE-CUT problem for $d+1$ players for the oracle functions requires time complexity $\theta(N^{d-1})$.*

PROOF. In the proof of Lemma 3, the Kuhn's triangulation in the last section reduces the problem of finding a Simmons-Su simplex to the problem of finding a Sperner simplex. Because a Sperner simplex can be found in $O(N^{d-1})$ time (Deng et al. 2011), this reduction allows us to find a Simmons-Su simplex in time $O(N^{d-1})$. Therefore, the solution corresponds to a DISCRETE ENVY-FREE CAKE CUT.

We note here that the algorithm is based on a binary search (on index) paradigm in fixed-point computation (Chen and Deng 2008). This framework, with further insight with respect to monotone utility function allows us to develop the FPTAS for three players discussed in the next section.

To prove the lower bound, we apply the same reduction as the one in the last section. The lower bound $\Omega(N/2^d)^{d-1}$ of ENVY-FREE CAKE CUT follows immediately from a lower bound $\Omega(N^{d-1})$ of BROUWER of size length N (Deng et al. 2011). Because d is a constant here, we obtain the lower bound $\Omega(N^{d-1})$ for ENVY-FREE CAKE CUT.

4. An FPTAS for Three Players with Monotone Utility Functions

For general oracle utility functions, the above results imply a $\theta(1/\epsilon)$ matching bound when the number of players is three. Further improvement can only be possible when we have further restrictions on the utility functions. In this section, we assume the utility functions in addition to satisfying the nonnegativity condition and the Lipschitz condition are also monotone.

Even in this case, the celebrated Stromquist's moving knife (Stromquist 1980) for envy-free cake cutting of three players can be shown to have an exponential lower bound. The main result here is to give an algorithm with a running time polynomial in the number of bits of $1/\epsilon$. Here a matching lower bound can be derived from very simple utility functions.

4.1. Complexity of Finding Stromquist's Solution

The celebrated Stromquist's solution (Stromquist 1980) involves a referee who moves her sword from left to right. The three players each has a knife at the point that would cut the right piece to the sword in half, according to their own valuation. Although the referee's sword moves right, the three knives all move right in parallel but possibly at different speeds. At all times, each player evaluates the piece to the left of the sword, and the two pieces that would result if the middle knife cuts. If any of them sees the left piece of the sword is the largest, he would shout "cut." Then the sword cuts. The leftmost piece is assigned to the player who shouted. For the two players who did not shout, the one whose knife is to the left of the middle knife receives the middle piece, and the one whose knife is on the right of the middle knife receives the rightmost piece.

We will show that Stromquist's moving knife procedure cannot be turned into a polynomial-time algorithm for finding an ϵ-envy-free solution. We will do this by showing that the particular fixed point found by the Stromquist's procedure cannot be found with polynomial number of queries. We assume a query can ask about the valuation of a player for an interval (x, y).

Suppose we have three players A, B, and C. The cake is represented as an interval $[0, 1]$. Players A and B have the same utility function. Their utility functions can be described as follows (we should normalize them for consistency but did not for simplicity of presentation).

$$u_A(x, y) = u_B(x, y) = \begin{cases} 0 & \text{for } x = 0,\ y = 1/10 \\ 2 & \text{for } x = 1/10,\ y = 3/10 \\ 100 & \text{for } x = 3/10,\ y = 4/10 \\ 2 & \text{for } x = 4/10,\ y = 8/10 \\ 100 & \text{for } x = 8/10,\ y = 9/10 \\ 0 & \text{for } x = 9/10,\ y = 1. \end{cases} \quad (1)$$

The value of both players for any other interval can be computed, assuming that their valuation is uniform across all of the above intervals. For example $u_A(1/20, 3/20) = 0.5$. u_C can be described in a similar way as follows:

$$u_C(x, y) = \begin{cases} 100 - \delta & \text{for } x = 0,\ y = 1/10 \\ 2 & \text{for } x = 1/10,\ y = 3/10 \\ 98 & \text{for } x = 3/10,\ y = 4/10 \\ 2 & \text{for } x = 4/10,\ y = 5/10 \\ 0 & \text{for } x = 5/10,\ y = 1. \end{cases} \quad (2)$$

Assume that the value of δ is very small. Now consider the Stromquist's moving knives. Suppose the referee starts by moving her knife from 0 towards 1. A and B have the same utility function, so their knives are going to be at the same place. Moreover, the "middle knife" will be always A's or B's. Observe that when the referee's knife reaches point $1/10$, the middle knife is at point $4/10$. No one will shout cut yet. Now, observe that because according to A and B the density of the interval $(1/10, 3/10)$ is twice the interval $(4/10, 5/10)$, the speed of the knives of A and B will be exactly the same as the referee's knife. For a similar reason, according to C the value of the middle piece (from referee's knife to the knife of A or B) will remain just δ above the value of the leftmost piece until the referee's knife goes slightly beyond $3/10$. At that time, C will shout cut and receive the leftmost piece. Players A and B happily split the rest of the cake equally.

Now, we will slightly perturb the utility function of C. At a point $1/10 \leqslant x \leqslant 3/10$ perturb the utility function of C in the following way: increase the utility of C for the interval $(x - \delta, x)$ by $\delta/2$ and decrease his utility for the interval $(x, x + \delta)$ by the same amount. Call the player with this perturbed utility function C_x.

Now, it is easy to see that if we are running the Stromquist's method, player C_x will shout cut at the time the referee's knife reaches x. It is also not hard to see that any value query asking for the value of an interval cannot distinguish C from C_x unless one end point of the interval for which it is querying the value is in $(x - \delta, x + \delta)$. Therefore, it is not possible to distinguish C_x from C with queries of order polylogarithmic in $1/\delta$.

4.2. An FPTAS for Three Players

THEOREM 6. *When the utility functions satisfy the above three conditions: nonnegativity condition, Lipschitz condition, and monotonicity, an ϵ envy-free solution can be found in time $O(\log 1/\epsilon)^2$ when the number of players is three.*

For three players, the closed set of all possible cuts is a triangle. To obtain an approximate discrete envy-free solution, the triangle is partitioned by Kuhn's triangulation. Let V be the set of vertices of all base cells. Recall the labeling and coloring rules defined in the previous sections. We partition V into three control subsets V_0, V_1, V_2, each corresponding to one player. According to the labeling rule given in the proof of Claim 1, a vertex $\mathbf{X}$ is assigned to V_t if $\sum_{i=0}^{d} iX_i \equiv t (\bmod\, d+1)$. By this labeling rule, the three vertices of each base simplex belong to different subsets V_t's.

Next, we should color each vertex in V. For a vertex $\mathbf{X}$ labeled by t, we let player t choose. If its optimal segment is the one of length X_s/N, $0 \leqslant s \leqslant 2$, we assign color s to the vertex $\mathbf{X}$.

We proved a $\theta(1/\epsilon)$ time solution for three players with general utility functions in the previous section.

The improvement has been made possible by an efficient way to compute the index of a triangulated polygon, i.e., counting the sum of the signs of base intervals on the boundary. The main idea is an observation that at some appropriate cuts, the colors along the cut are monotone. Therefore, we can find the boundary of the three different colors on this cut to calculate the sum quickly.

More specifically, suppose the cake is cut into three pieces: a fixed interval $[0, c/N]$, an interval varying in its right end $[c/N, (c + x)/N]$, and an interval varying in its left end $[(c + x)/N, 1]$, where c is a fixed integer. For any player, as x increases, the utility of the first piece will not change, the second increases, and the third decreases because of the monotonicity. Therefore, given the player, its preferred pieces along the cuts in the form $(c, x, N - x - c)$ with x varying will be monotone with only two possible changes in its colorings of those nodes. Note that such cuts correspond to nodes in a line segment in the barycentric coordinates, parallel to one of the boundaries of the triangle. If we can find the two possible changing points of the colors, we know the colors of all vertices along the line segments that belong to that player. We should do it one by one for each of the three players. We should, in the end, know all the colors along this line segment. Therefore, we can calculate the number of base segments with colors 0 and 1 on their different end vertices. That allows us to calculate the number of such base segments along the boundaries of the two polygons obtained by the line segment crossing the previous polygon. This allows us to design the following algorithm.

At any time in the algorithm, we maintain a subset of V, with a nonzero index value, $V(i_1, i_2, k_1, k_2) = \{(i, j, k)\colon i, j, k \geqslant 0, i+j+k = N, i_1 \leqslant i \leqslant i_2, k_1 \leqslant k \leqslant k_2\}$, delimited by $i = i_1$, $i = i_2$ and $k = k_1$, $k = k_2$.

Algorithm 1. 1. If $i_2 - i_1 = 1$ and $k_2 - k_1 = 1$, find a fully colored base triangle of $V(i_1, i_1+1, k_1, k_1+1)$, terminate.

2. Choose $\max\{i_2 - i_1, k_2 - k_1\}$ (and assume it is $i_2 - i_1$ w.l.o.g.).

3. Let $i_3 = \lfloor (i_1 + i_2)/2 \rfloor$.

4. Calculate *index*$(V(i_1, i_3, k_1, k_2))$ and *index*$(V(i_3, i_2, k_1, k_2))$.

5. Recurse on one of the two subpolygons with a nonzero index.

Note that Step 4 is based on the above analysis and will be explained in more detail later. By definition, $\text{index}(V(i_1, i_2, k_1, k_2)) = \text{index}(V(i_1, i_3, k_1, k_2)) + \text{index}(V(i_3, i_2, k_1, k_2))$. At least one of the $\text{index}(V(i_1, i_3, k_1, k_2))$ and $\text{index}(V(i_3, i_2, k_1, k_2))$ is nonzero because *index*$(V(i_1, i_2, k_1, k_2))$ is nonzero. The algorithm always keep a polygon of nonzero index because the initial polygon has a nonzero index. At the base case $i_2 = i_1 + 1$ and $k_2 = k_1 + 1$, $V(i_1, i_1+1, k_1, k_1+1)$ is either already a base triangle (with a nonzero index), or a diamond shape consisting of two base triangles, one of which must be of index nonzero. The correctness follows because the only possibility for its index being nonzero is when it is colored with all three colors.

For complexity analysis, we first characterize the boundary conditions:

Property 1. *There are up to three types of boundaries for* $V(i_1, i_2, k_1, k_2)$

1. $B_{i=c} = \{(i, j, k) \in V\colon i \textit{ constant}, k_1 \leqslant k \leqslant k_2, i+j+k = N, i, j, k \geqslant 0\}$

2. $B_{k=c} = \{(i, j, k) \in V\colon k \textit{ constant}, i_1 \leqslant i \leqslant i_2, i+j+k = N, i, j, k \geqslant 0\}$

3. $B_{j=0} = \{(i, 0, k) \in V\colon i+k = N, i, k \geqslant 0\}$.

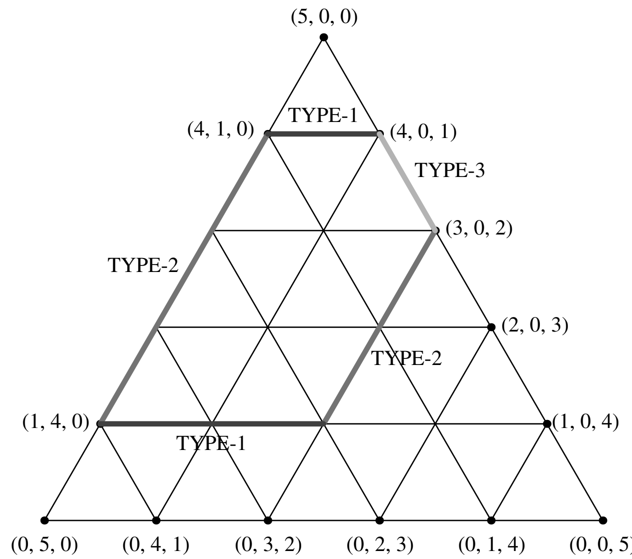
See Figure 10 as an example.

Second, we establish the monotonicity property. Note that for the third type of boundaries listed above, the monotonicity property does not hold for $B_{j=c}$ in general, but only for $B_{j=0}$.

Property 2. *The colors of $V_0 \cap B_{i=c}$ are monotone in k, and so are $V_1 \cap B_{i=c}$ and $V_2 \cap B_{i=c}$. The same hold for $V_t \cap B_{k=c}$, as well as $V_t \cap B_{j=0}$, $t = 0, 1, 2$.*

Proof. Observe that at $V_0 \cap B_{i=c}$, the color is determined by Player 0 by finding the maximum of $u_0([0, c/N])$, $u_0([c/N, (c+j)/N])$, and $u_0([(c+j)/N, 1])$, where $0 \leqslant j \leqslant N - c$. The first item is fixed. The second item is increasing in j because $[c/N, (c+j)/N] \subseteq [c/N, (c+j')/N]$ for $j \leqslant j'$ and u_i is assumed to be monotone. For the same reason, the last item is decreasing in j. The color will be 0 if $u_0([0, c/N])$ is the maximum, 1 if $u_0([c/N, (c+j)/N])$ is the maximum, or 2 otherwise. In general, as j increases, the color in $V_0 \cap B_{i=c}$ will start with 2, then 0, and finally 1, assuming nondegeneracy. However, any of those colors may be missing.

Figure 10. An example of three types of boundaries.



The same analysis holds for other players and for the case when k is fixed. However, it does not hold when j is fixed, except when $j = 0$.

Property 2 allows us to find the colors of all the vertices on the boundaries $V_t \cap B_{i=i_1}$, $V_t \cap B_{i=i_2}$, $V_t \cap B_{j=0}$, $V_t \cap B_{k=k_1}$ and $V_t \cap B_{k=k_2}$, $t = 0, 1, 2$, in time proportional to the logarithm of the length of the sides, by finding all vertices at which the color changes along it. Once the changing points of the colors are found, using monotonicity, the total number of positively and negatively signed base intervals along the boundary can be calculated in constant time.

The recursive algorithm reduces the size of $V(i_1, i_2, k_1, k_2)$ geometrically. Let $L = \max\{|i_1 - i_2|, |k_1 - k_2|\}$. L halves in two rounds of the algorithm. It takes $2\log N$ steps to reduce L to 1.

At each round of the algorithm, we need to find out, for each of three players, for each side of $V(i_1, i_2, k_1, k_2)$ (up to five in all), the boundary of the player's color changes (two boundaries). That takes time $\log L$, bounded by $\log N$, to derive a total of $3 \times 5 \times 2\log N$.

Therefore, the time complexity is $O(\log N)^2$.

5. Discussion and Conclusion

It remains open whether the approximate envy-free cake-cutting problem for four or more players would allow for a fully polynomial-time approximation scheme as it does in the three-player case, when we are dealing with monotone functions.

Utility functions may often not be monotone, and superadditive utility functions are abundant in real-life situations. However, the significant improvement in the three-player case for monotone functions would by itself pose the question whether such progress can be carried over to higher dimensions.

Because the envy-free cake-cutting problems have extensive applications in political and social sciences, explicit procedures solving such problems have been developed to implement this fairness solution concept. Our current complexity study for cake cutting in the general setting could become more practical in those applications. We leave those as open problems for interested readers to further explore the complexity of algorithms developed in those areas, such as consensus halving (Simmons and Su 2003) and cutting multiple cakes (Cloutier et al. 2010).

Acknowledgments

The authors thank the referees for their comments and suggestions that significantly improved the previous versions of this paper in the presentation, organization, and possible extensions, as well as references. The third author thanks Arash Asadpour, as well as the organizers and participants of Dagstuhl's fair division workshop for valuable discussions. The work of the first author is supported by a grant from the University of Liverpool, by the Research Grants Council of Hong Kong [Project RGC CityU 112909], and by the National Science Foundation of China [Grant 61173011].

References

Asadpour A, Saberi A (2007) An approximation algorithm for max-min fair allocation of indivisible goods. *Proc. 39th Annual ACM Sympos. Theory Comput.* (ACM, New York), 114–121.

Bansal N, Sviridenko M (2006) The Santa Claus problem. *Proc. 38th Annual ACM Sympos. Theory Comput.* (ACM, New York), 31–40.

Blum A, Jackson J, Sandholm T, Zinkevich M (2004) Preference elicitation and query learning. *J. Machine Learn. Res.* 5:649–667.

Blumrosen L, Nisan N (2007) Combinatorial auctions. Nisan N, Roughgarden T, Tardos E, Vazirani V, eds. *Algorithmic Game Theory* (Cambridge University Press, London), 267–298.

Brams SJ, Taylor AD (1995) An envy-free cake division protocol. *Amer. Math. Monthly* 102(1):9–18.

Brouwer LEJ (1910) Ueber eineindeutige, stetige Transformationen von Flächen in sich. *Math. Ann.* 69:176–180.

Chen X, Deng X (2008) Matching algorithmic bounds for finding a Brouwer fixed point. *J. ACM* 55(3).

Chen X, Deng X (2009a) On the complexity of 2D discrete fixed point problem. *Theoret. Comput. Sci.* 410(44):4448–4456.

Chen X, Deng X (2009b) A simplicial approach for discrete fixed point theorems. *Algorithmica* 53(2)250–262.

Cloutier J, Nyman KL, Su FE (2010) Two-player envy-free multi-cake division. *Math. Soc. Sci.* 59(1):26–37

Crescenzi P, Silvestri R (1998) Sperner's lemma and robust machines. *Comput. Complexity* 7(2):163–173.

Daskalakis C, Goldberg PW, Papadimitriou CH (2006) The complexity of computing a Nash equilibrium. *Proc. 38th Annual ACM Sympos. Theory Comput.* (ACM, New York), 71–78.

Deng X, Qi Q, Saberi A, Zhang J (2011) Discrete fixed points: Models, complexities and applications. *Math. Oper. Res.* 36(4):636–652.

Dobzinski S, Nisan N, Schapira M (2010) Approximation algorithms for combinatorial auctions with complement-free bidders. *Math. Oper. Res.* 35(1):1–13.

Edmonds J, Pruhs K (2006a) Cake cutting really is not a piece of cake. *Proc. 17th Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM, New York), 271–278.

Edmonds J, Pruhs K (2006b) Balanced allocations of cake. *Proc. 47th Annual IEEE Sympos. Foundations Comput. Sci.* (IEEE, New York), 623–634.

Elimam AA, Girgis M, Kotob S (1996) The use of linear programming in disentangling the bankruptcies of Al-Manakh stock market crash. *Oper. Res.* 44:665–676.

Even S, Paz A (1984) A note on cake-cutting. *Discrete Appl. Math.* 7:285–286.

Friedl K, Ivanyos G, Santha M, Verhoeven F (2005) On the black-box complexity of Sperner's lemma. *Proc. 15th Internat. Sympos. Fundamentals Comput. Theory* (Springer, Lübeck, Germany), 245–257.

Halland A, Zeckhauser R (1979) The efficient allocation of individuals to positions. *J. Political Econom.* 87(2):293–314.

Hirsch PM (1986) From ambushes to golden parachutes: Corporate takeover as an instance of cultural framing and institutional integration. *Amer. J. Sociol.* 9(4):800–837.

Kelly FP, Maulloo AK, Tan DHK (1998) Rate control in communication networks. Shadow prices, proportional fairness, and stability. *J. Oper. Res. Soc.* 49:237–252.

Kleinberg J, Rabani Y, Tardos E (1999) Fairness in routing and load balancing. *Proc. 40th IEEE Sympos. Foundations Comput. Sci.* (IEEE, New York).

Kuhn HW (1960) Some combinatorial lemmas in topology. *IBM J. Res. Development* 4:518–524.

Kumar A, Kleinberg JM (2006) Fairness measures for resource allocation. *SIAM J. Comput.* 36(3):657–680.

Lipton RJ, Markakis E, Mossel E, Saberi A (2004) On approximately fair allocations of indivisible goods. *ACM Conf. Electronic Commerce* (ACM, New York), 125–131.

Megiddo N, Papadimitriou CH (1991) A note on total functions, existence theorems and computational complexity. *Theoret. Comput. Sci.* 81:317–324.

Meunier F (2008) Discrete splittings of the necklace. *Math. Oper. Res.* 33(August):678–688.

Moulin H (2007) Minimizing the worst slowdown: Offline, online. *Oper. Res.* 55(5):876–889.

Papadimitriou CH (1994) On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Systems Sci.* 48(3):498–532.

Pratt JW, Zeckhauser RJ (1990) The fair and efficient division of the Winsor family silver. *Management Sci.* 36(11):1293–1301.

Scarf H (1977) The computation of equilibrium prices: An exposition. H. Cowles Foundation for Research in Economics, Yale University, New Haven, CT.

Sherstyuk K (1998) How to gerrymander: A formal analysis. *Public Choice* 95:27–49.

Simmons FW (1980) Private communication to Michael Starbird.

Simmons FW, Su FE (2003) Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Math. Soc. Sci.* 45(1):15–25.

Sperner E (1928) Neuer Beweis für die invarianz der dimensionszahl und des gebietes. *Abh. Math. Sem. Univ. Hamburg* 6:265–272.

Steinhaus H (1948) The problem of fair division. *Econometrica* 16:101–104.

Steinhaus H (1949) Sur la division pragmatique. *Econometrica* (supplement) 17:315–319.

Stromquist W (1980) How to cut a cake fairly. *Amer. Math. Monthly* 87(8):640–644. Addendum 88(8, 1981):613–614.

Su FE (1999) Rental harmony: Sperners lemma in fair division. *Amer. Math. Monthly* 106:930–942.

Thomson W (2007) Children crying at birthday parties. Why? *Econom. Theory* 31:501–521.

Todd MJ (1976) *The Computation of Fixed Points and Applications*. Lecture Notes in Economics and Mathematical Systems (Springer-Verlag, New York).

Woeginger GJ, Sgall J (2007) On the complexity of cake cutting. *Discrete Optim.* 4(2):213–220.

Xiaotie Deng is a chair professor in economics and computation in the Department of Computer Science, University of Liverpool. His research focus is on algorithmic game theory, which deals with computational issues on fundamental economic problems such as Nash equilibrium, resource pricing and allocation protocols such as auction and market equilibrium, as well as theory and practice in Internet market design.

Qi Qi is an assistant professor in the Department of Industrial Engineering and Logistics Management at Hong Kong University of Science and Technology. Her research interests include game theory and mechanism design, Internet and computational economics, online auctions, resource allocation, and optimization.

Amin Saberi is an associate professor and 3COM Faculty Scholar in the Department of Management Science and Engineering at Stanford University. His research interests include algorithms, approximation algorithms, and algorithmic aspects of games, markets, and networks.