

Non-exploitable protocols for repeated cake cutting

Omer Tamuz* and Shai Vardi and Juba Ziani

California Institute of Technology

Abstract

We introduce the notion of *exploitability* in cut-and-choose protocols for repeated cake cutting. If a cut-and-choose protocol is repeated, the cutter can possibly gain information about the chooser from her previous actions, and exploit this information for her own gain, at the expense of the chooser. We define a generalization of cut-and-choose protocols—*forced-cut protocols*—in which some cuts are made exogenously while others are made by the cutter, and show that there exist non-exploitable forced-cut protocols that use a small number of cuts per day: When the cake has at least as many dimensions as days, we show a protocol that uses a single cut per day. When the cake is 1-dimensional, we show an adaptive non-exploitable protocol that uses 3 cuts per day, and a non-adaptive protocol that uses n cuts per day (where n is the number of days). In contrast, we show that no non-adaptive non-exploitable forced-cut protocol can use a constant number of cuts per day. Finally, we show that if the cake is at least 2-dimensional, there is a non-adaptive non-exploitable protocol that uses 3 cuts per day.

1 Introduction

The problem of *cake cutting* or *fair division* concerns the splitting of a divisible good among a number of agents, without transfers. This problem has important applications in many fields, including multiple agent systems (Chen et al. 2010; Balkanski et al. 2014), and has attracted the attention of mathematicians (Knaster 1944; Neyman 1946; Dubins and Spanier 1961), computer scientists (Even and Paz 1984; Edmonds and Pruhs 2006), political scientists (Brams and Taylor 1996) and economists (Steinhaus 1948; Varian 1974) for many decades.

Consider a heterogeneous divisible good (i.e., a cake), represented by the interval $(0, 1)$. There are two strategic agents, each with a private (additive, non-atomic) valuation function v_i . The classical goal in the cake cutting literature is to design a protocol that divides the cake *fairly* between the two agents. There are several common definitions of fairness, including *proportionality* (each agent’s utility from her piece is at least half of her utility for the entire cake) and

envy-freeness (neither agent prefers the other agent’s piece to her own). When there are two players, envy-freeness implies proportionality (see, e.g., Procaccia 2016). The following well known cut-and-choose protocol is envy-free.

Protocol 1.1. *One agent cuts the cake into two pieces, X and \bar{X} . The other agent chooses one of $\{X, \bar{X}\}$. The agent that cut the cake receives the remaining piece.*

We make the standard assumption of the cake cutting literature that the agents use maximin solutions: they maximize their minimum utility over all possible actions of the others (Wald 1945; 1949; Brams, Jones, and Klamler 2006; Sinn 2012). It follows that in Protocol 1.1 the agent that cuts the cake (henceforth *the cutter*) will cut the cake into two pieces of equal value to her, and the agent that chooses (*the chooser*) will choose a piece that is worth at least as much as its complement. This is also sometimes referred to as *risk-aversion*.

We consider now the following scenario. On each of n days the chooser is approached by a different cutter that has a divisible good (or cake). The chooser has the same valuation for all cakes, but each cutter may have a different valuation. Each cutter cuts the cake, and the chooser chooses the piece she prefers. At first glance, it seems like executing Protocol 1.1 every day is a good idea: it is envy-free on every day. Indeed, if the cutters cannot observe the previous choices made by the chooser, then all of the desirable properties of Protocol 1.1 carry over to the dynamic case. But what if the cutters *can* observe the previous choices of the chooser?

Say that on the first day the first cutter cuts the cake down the middle into $a = (0, 1/2)$ and $b = (1/2, 1)$, and that the chooser chooses a . If the second cutter values b at strictly more than $1/2$, she can cut the cake in the same place, knowing the chooser will choose a again. Thus the cutter no longer cuts the cake into two pieces that are equally valuable to her, and receives a utility strictly greater than $1/2$. In other words, the cutter is able to *exploit* the information she learns about the chooser. If the first $k - 1$ cutters make different cuts, the k^{th} cutter may have quite a bit of information that she can exploit. Thus, in this dynamic setting, it is no longer true that the cutter always cuts the cake into two equally valuable pieces, and always gets utility $1/2$.

We note that we do not constrain the chooser to always choose the larger of the two pieces offered to her. Indeed, it

*Omer Tamuz was supported by a grant from the Simons Foundation (#419427).

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

may be reasonable for the chooser to try to trick the cutter into thinking she has a different utility function (see Example A.1). However, as our goal will be to design protocols in which the cutter cannot exploit the chooser, the chooser will have no incentive to choose a smaller piece.

We develop cut-and-choose protocols in which the cutter is restricted, in her choice of cuts, while still having enough options to always be able to cut the cake into two pieces of equal value. A protocol is *non-exploitable* if—informally—the cutter, for any choice of allowed division of the cake, can never be sure which piece the chooser will choose, given her observations of the chooser’s past actions. Our goal in this paper is to develop and analyze simple non-exploitable, envy-free dynamic cut-and-choose protocols. In addition to non-exploitable being a desirable characteristic in and of itself, it implies that the agents behavior is simple, as in Protocol 1.1: in equilibrium, the cutter cuts the cake into two pieces that are each worth $1/2$ to her, and the chooser chooses the “larger” piece. This is advantageous, as being able to predict and analyze agents’ behavior is important in strategic environments, e.g., (Wald 1949; Fishburn 1970; Gilboa and Schmeidler 1989).

We concentrate on a family of cut-and-choose protocols that we call *forced-cut protocols*. In these, there are zero or more exogenous *forced cuts* that partition the cake into segments on each day, and the cutter is required to divide each segment into two slices. The chooser then chooses between two sets of slices; each set of slices is called a *piece* of cake. We show that this family of protocols is sufficient in order to obtain non-exploitable protocols that are envy-free and use few cuts.

1.1 Results

Our first result shows that if the cake is not one dimensional, but instead has at least as many dimensions as there are days, there is a non-exploitable protocol that uses the minimum possible number of cuts—a single cut per day.

Theorem 1.2. *There is a non-adaptive, non-exploitable, envy-free n -day forced-cut protocol for k -dimensional convex sets (where $k \geq n$) that uses n cuts.*

For a one dimensional cake, we give an adaptive protocol that uses a linear number of forced cuts.

Theorem 1.3. *There is an adaptive, non-exploitable, envy-free n -day forced-cut protocol for the interval $(0, 1)$ that uses $3n$ cuts.*

Our most technical result shows that there is no forced-cut single-dimensional non-exploitable protocol that uses a linear number of cuts.

Theorem 1.4. *Let $c > 0$ be a constant, and let \mathcal{P} be a non-adaptive, envy-free forced-cut protocol that uses at most cn cuts. Then there exists a constant n_0 such that for every $n > n_0$, \mathcal{P} is exploitable.*

Nevertheless, we give a non-adaptive non-exploitable protocol that uses a quadratic number of cuts.

Theorem 1.5. *There is a non-adaptive, non-exploitable, envy-free n -day forced-cut protocol for the interval $(0, 1)$ that uses $\Theta(n^2)$ cuts.*

Finally, show that adding a single dimension is enough to overcome the lower bound of Theorem 1.4.

Theorem 1.6. *There is a non-adaptive non-exploitable n -day forced-cut protocol for the set $(0, 1) \times (0, 1)$ that uses $3n$ cuts.*

1.2 Related work

There is a vast literature on cake-cutting dating back to the first half of the previous century e.g., (Knaster 1944; Neyman 1946; Steinhaus 1948; Dubins and Spanier 1961; Varian 1974; Stromquist 1980; Even and Paz 1984; Brams and Taylor 1996; Edmonds and Pruhs 2006; Chen et al. 2010; Balkanski et al. 2014). For an introduction to cake-cutting, we refer the reader to (Robertson and Webb 1998; Procaccia 2016). We focus here on work that is related to the major themes of this paper.

Repeated cake-cutting. The work most related to ours is the paper of (Delgosha and Gohari 2012), who consider the problem of understanding how agents behave when engaging in classical protocols, when partial information about the other agent’s utility can be revealed. They consider several ways information can be revealed: the cutter can spy on the chooser; the chooser can voluntarily give information about her utility to the cutter if it is beneficial to her; the cutter and chooser repeatedly cut and choose from identical cakes with identical utilities, and only gain information about each other’s utilities by looking at the history of actions taken by the players. The last variant is very similar to our setting; the main difference being they only analyze the existing cut-and-choose protocol if it is played repeatedly; they do not design new protocols. Additional differences are that in their setting, the utility functions are sampled from some known probability distribution (we perform a worst-case analysis), and the cutter has exactly two options (compared to an infinite amount, as in our setting). They show that a particular set of actions, which include the chooser acting myopically, leads to a Trembling Hand Perfect Equilibrium.

Cut complexity. There has been much work on the cut complexity of protocols - how many cuts are necessary to achieve envy-free, proportional and/or equitable allocations? (Steinhaus 1948) introduced the *last diminisher* procedure, which uses $O(n^2)$ cuts to obtain a proportional division, where n is the number of agents. (Even and Paz 1984) gave a deterministic protocol that uses $O(n \log n)$ cuts and (Edmonds and Pruhs 2006) showed that this is tight. (Robertson and Webb 1998) propose a more formal model for the query complexity of cake cutting algorithms (the algorithm of (Even and Paz 1984) falls within this model, as does the bound of (Edmonds and Pruhs 2006)), in which there are two types of queries: (I) Given a value $\alpha \in (0, 1)$, an agent cuts the cake so that the resulting piece is worth α to her; (II) An agent says how much a piece previously cut by the protocol is worth to her. This model has become the standard model for investigating the complexity of cake-cutting, e.g., (Edmonds and Pruhs 2006; Woeginger and Sgall 2007;

Procaccia 2009; Aziz and Mackenzie 2016a; 2016b; Procaccia and Wang 2017); however it is not relevant to our setting as it assumes that the agents are not strategic and reply to queries truthfully.

Exploitability, truthfulness and privacy. The reader that is familiar with the literature on cake-cutting will likely make a connection between the notion of exploitability and truthfulness. Truthfulness (sometimes called strategyproofness or incentive-compatibility), e.g., (Brams, Jones, and Klamler 2006; Chen et al. 2010; Mossel and Tamuz 2010) is when the agents are incentivized to reveal their true preferences. The notion of truthfulness makes sense when there is a centralized mechanism to which the agents reveal their preferences. One of the advantages afforded by cut-and-choose protocols is that the agents do not need to reveal their entire utility functions. Revealing the function in its entirety is not always computationally or indeed even information-theoretically possible, and does not preserve privacy.

The notion of exploitability is also related to *privacy*. Clearly, if the chooser’s utility function remains completely private, no protocol is exploitable. However, it is inevitable that some information leaks in a repeated game (barring the trivial case in which the chooser acts independently of her valuation). An advantage of cut-and-choose protocols over many others, e.g., (Even and Paz 1984; Brams, Jones, and Klamler 2006; Chen et al. 2010; Mossel and Tamuz 2010; Balkanski et al. 2014)) is that very little information is revealed about the players’ utilities. (Brams, Jones, and Klamler 2006) remark upon this extensively, although their approach to this notion is slightly less rigorous than ours.

Adaptivity vs. non-adaptivity. Recently there has been a surge of work in the computer science community on the *adaptivity gap* in stochastic settings algorithms, e.g., (Gupta, Nagarajan, and Singla 2017; Blum et al. 2015) – how much can be gained by allowing the algorithm to adapt to the behavior or strategies of the players. Clearly, non-adaptive protocols are simpler to describe and implement, as they can simply be described upfront, while adaptive protocols can be strictly stronger (Gupta, Nagarajan, and Singla 2017). When designing a protocol or algorithm, it is useful to know the tradeoff in order to decide whether to design an adaptive or non-adaptive protocol.

Multiple dimensions. Most of the work on cake-cutting focuses on single dimensional cakes. (Segal-Halevi et al. 2017) explain this concisely “This ... is usually justified by the reasoning that higher-dimensional settings can always be projected onto one dimension, and hence fairness in one dimension implies fairness in higher dimensions.” Our results show that the complexity of protocols can depend on the number of dimensions of the cake.

There has been some recent work on multiple-dimensional cakes, mostly on 2-dimensional cakes, e.g., (Iyer and Huhns 2009; Segal-Halevi et al. 2017). (Segal-Halevi et al. 2017) consider the problem of dividing 2-dimensional cakes when the shape of the allocated pieces

is restricted; they give upper and lower bounds on achievable levels of proportionality that can be guaranteed for different shapes of pieces. We note that there is some literature on pie-cutting, e.g., (Barbanel, Brams, and Stromquist 2009), but this is still a one-dimensional object, albeit a circle instead of a line.

Dynamic fair division. In recent years, there has been a growing interest in dynamic (or online) resource allocation. (Walsh 2011) considers the cake-cutting problem in which players arrive and depart online, and must be allocated a slice before they depart. (Friedman, Psomas, and Vardi 2015; 2017) consider the scenario in which a single resource needs to be divided between arriving players such that there are few reallocations when an agent arrives. (Segal-Halevi 2016) considers a dynamic reallocation setting with some agents that are present from the start some newly arrived agents.

2 Model and preliminaries

A k -dimensional *cake* \mathcal{C} is a compact convex subset of \mathbb{R}^k . A valuation (or utility) function u is a Borel probability measure on \mathcal{C} . We assume that valuation functions are nonatomic, i.e., they assign zero measure to singletons, and furthermore assign measure zero to sets with empty interior (e.g., lower dimensional sets). Equivalently, valuations are absolutely continuous with respect to the Lebesgue measure.

A *cut* of a cake is a $(k-1)$ -dimensional hyperplane bisecting the cake. A finite set of cuts of a cake defines a *partition* of the cake into finitely many *slices* in the obvious way.¹ A *piece* of a cake is a union of slices. A *division* $D = (X, \bar{X})$ of the cake is a partition of the cake into a piece X and its complement.

We consider a setting with n identical copies $(\mathcal{C}^1, \dots, \mathcal{C}^n)$ of a cake \mathcal{C} . One player—the *chooser*—remains constant across days, and there is a (possibly) different *cutter* on each day.

In a *repeated cut and choose* protocol, on each day t the cutter selects a division $D^t = (X^t, \bar{X}^t)$ of \mathcal{C}^t ; the chooser selects one of the two pieces, receives that piece, and the cutter receives the other. Each player’s utility for that day is her valuation for the piece she receives. The cutter is constrained to choose a division out of a set P^t that depends on the day t , and possibly also on the previous choices made by the players.

The complexity of a protocol $\mathcal{P} = (P^1, \dots, P^n)$ is the maximal total number of cuts needed to generate any sequence of legal divisions $(D^1, \dots, D^n) \in \mathcal{P}$.

A strategy of the cutter is an assignment, on each day, of a division $D^t \in P^t$, depending possibly on the previous choices made by the players. A strategy of the chooser is a function that, depending on the division offered on each day, and possibly also on previous history, determines her choice of piece. We say that the chooser is *myopic* if her strategy is, on each day t , to choose whichever of (X^t, \bar{X}^t)

¹We can arbitrarily assign elements of the (lower-dimensional) cuts to the slices. This will not affect the slices’ valuations, since the cuts themselves always have zero value.

has higher valuation for her. We assume that in case of indifference the chooser chooses consistently, so that if she is offered the same division on a subsequent day, she makes the same choice.

We say that a protocol is *non-exploitable* (by the cutter) if for every strategy of the cutter there exists some valuation u of the chooser, such that, assuming that the chooser is myopic, the cutter receives on each day a piece that she values by at most $1/2$. Equivalently, and perhaps more intuitively, the cutter, regardless of the strategy she chooses, can never be sure which piece the (myopic) chooser will choose, given her observations of the chooser's past actions. That is, on each day t , for any history occurring on the previous days, and any choice of division D^t , there are utilities u_1, u_2 of the chooser that are both consistent with the chooser's previous actions, but lead to different choices on day t .

For example, consider executing Protocol 1.1 on consecutive days. The sets P^t are the sets of all divisions; the cutter is unconstrained in her choices. Even for $n = 2$ this protocol is exploitable: the cutter can offer the same division twice, in which case she knows that for any utility u of the chooser, the choice on the second day will be identical to that of the first.

The set P^t can be arbitrarily complex; as we are interested in *simple* protocols, we introduce the notion of *forced cuts* and *forced-cut protocols*, which are simple to describe and implement with only a few cuts while being powerful enough to guarantee non-exploitability. A *forced cut* is a cut that is made exogenously; analogously one can think of it as a cut that the cutter *must* make. A *forced-cut protocol* is one in which (possibly zero) forced cuts partition the cake \mathcal{C}^t into *segments* X_i^t , and the cutter is required to make a single cut in each segment, partitioning it into two slices $X_{i,0}^t$ and $X_{i,1}^t$. The chooser is then required to choose between the pieces $\cup_i X_{i,0}^t$ and $\cup_i X_{i,1}^t$.

Example 2.1. We give two examples of forced-cut protocols.

1. Let the cake be the interval $(0, 1)$, and consider the following one-day protocol: there is a forced cut at 0.5 . The cutter makes a cut in each of the segments $(0, 0.5)$ and $(0.5, 1)$, to create the slices $X_{1,0} = (0, a)$, $X_{1,1} = (a, 0.5)$, $X_{2,0} = (0.5, b)$ and $X_{2,1} = (b, 1)$ for some $a \in (0, 0.5)$, $b \in (0.5, 1)$. The chooser must choose between the two pieces $X = X_{1,0} \cup X_{2,0}$ and $\bar{X} = X_{1,1} \cup X_{2,1}$.
2. Protocol 1.1 is a forced-cut protocol, with 0 forced cuts.

We remark that it is possible to extend the family of forced cut protocols to allow the chooser to make more than one cut in each segment. This is unnecessary for the upper bounds, and it is easy to extend the proof of the lower bound to allow such protocols. We remark upon this further in Section 4.2.

As a more illustrative example, we give a simple non-exploitable n -day forced-cut protocol. We note that this protocol applies to any cake.

Protocol 2.2. For $t = \{1, 2, \dots, n\}$, let H_{t-1} be the set of (forced and unforced) cuts made on day $t - 1$, and let $H_0 = \emptyset$. On day t , the forced cuts are H_{t-1} . Let the segments resulting from these cuts be X_1^t, \dots, X_m^t . The cutter

cuts each $X_k^t, k \in \{1, \dots, m\}$ into two slices, $X_{k,0}^t, X_{k,1}^t$. The chooser chooses between $\cup_k X_{k,0}^t$ and $\cup_k X_{k,1}^t$.

Note that on the first day, this protocol is identical to Protocol 1.1. Before we show that it is non-exploitable, we give a characterization of non-exploitable protocols will be useful for this and subsequent proofs.

Lemma 2.3. An n -day cut-and-choose protocol is non-exploitable if and only if for any possible choice of divisions by the cutter and choices of the chooser w.r.t. those divisions, there is utility function for the chooser that is consistent with those choices.

Proof. For the first direction, assume that the protocol is non-exploitable, but there exists a set of divisions $D = (D^1, \dots, D^n)$ and choices that is not consistent with any utility function of the chooser. Let t be the first day for which there is a division and choice that is not consistent with any utility function (note that $t \geq 2$, as for any D^1 , both choices are always possible). There must be at least one chooser utility function that is consistent with D^1, \dots, D^{t-1} , by the assumption that t is the first day for which there is not utility function consistent with a choice. Therefore, for D^t , the chooser will necessarily have a choice consistent with all previous choices. But by the assumption that this is a division for which some choice does not have a utility function, the chooser will have exactly one available choice. Therefore the protocol is exploitable.

The other direction is immediate from the definition of exploitability. \square

Theorem 2.4. Protocol 2.2 is non-exploitable and uses $\Theta(2^n)$ cuts.

Proof. To show that Protocol 2.2 is non-exploitable, we describe, for any set of divisions D and choices, a utility function of the chooser that is consistent with these choices; by Lemma 2.3 this implies that the protocol is non-exploitable. We will do this by showing that for any set of divisions D and any choices, there is at least one slice that is common to all the choices, and we let the chooser's utility be concentrated in that slice. We show this using induction on the number of days. The base of the induction, $t = 1$, is trivial. For the inductive step, assume that the inductive hypothesis holds for $t - 1$ days; there is therefore a slice $X_{k',j'}^{t-1}$ that is common to all of the choices until day $t - 1$. By the protocol definition, this slice is also a segment X_k^t on day t . The cutter cuts this segment into two slices: $X_{k,0}^t$ or $X_{k,1}^t$, and one of these must be in the piece chosen by the chooser, hence it is in all pieces chosen made until day t .

It is easy to verify that $2^t - 1$ cuts in total are made on each day, hence the total number of cuts for all n days is $\sum_{i=1}^n (2^i - 1) = 2^{n+1} - n - 2$. \square

The problem with Protocol 2.2 is that it uses $\Theta(2^n)$ cuts. We are interested in protocols that use a linear number of cuts (amounting to a constant number every day), or at the worst polynomial in the number of days. We first consider a simple case, when the cake is a k -dimensional convex set.

3 k -dimensional convex cakes

In this section we consider a compact, finite k -dimensional convex set \mathcal{C} with non-zero volume in \mathbb{R}^n . We define the following protocol:

Protocol 3.1. *On day t , the cutter makes a single cut that divides the cake into (X^t, \bar{X}^t) . The cut must be orthogonal to all previous cuts, and intersect all previous cuts within the interior of \mathcal{C} .*

Note that this protocol can only be implemented whenever the total number of days n is at most the dimensionality of the cake k : if $n > k$, on any day $t > k$ there cannot exist a cut that is orthogonal to all previous cuts. Without loss of generality, we fix $n = k$. We show the following.

Theorem 1.2. *Protocol 3.1 is non-exploitable and uses n cuts.*

Proof. Let h^1, \dots, h^n be the cuts made by the cutter on days $1, \dots, n$ respectively. In accordance with Protocol 3.1, these are orthogonal. Let $\mathbf{b}_1, \dots, \mathbf{b}_n$ be orthonormal vectors to the hyperplanes h^1, \dots, h^n respectively. $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ is a basis of \mathbb{R}^n ; henceforth, we write all points in the Euclidean space using the origin $\mathbf{0}$ and basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$. For any point x in \mathbb{R}^n , we denote x_t the t^{th} coordinate of that point (w.r.t. B). The cut h^t can be represented by a pair (t, c_t) , where $t \in \{1, \dots, n\}, c_t \in \mathbb{R}$ (h^t is the set of all points $x \in \mathcal{C}$ for which $x_t = c_t$). We denote $X^t = \mathcal{C} \cap \{x : x_t \leq c_t\}$ and $\bar{X}^t = \mathcal{C} \cap \{x : x_t > c_t\}$. By the description of the protocol, the point $c = (c_1, \dots, c_n)$ must be in the interior of the polytope.

We apply Lemma 2.3. Fix the cuts as above. Let $s = s_1, \dots, s_t$ denote the choices made by the chooser, where $s_i = 0$ iff the chooser chose X^t . Let Z be the intersection of all chosen pieces:

$$Z = \left(\bigcap_{i:s_i=0} X^i \right) \cap \left(\bigcap_{i:s_i=1} \bar{X}^i \right).$$

Any utility function of the chooser that sets $u(Z) > 1/2$ is consistent with these cuts and choices. There must be at least one such function, because Z contains an open set, as c is in the interior of the polytope. \square

4 The 1-dimensional case

In this section, we consider the cake that is the interval $(0, 1)$. This is the most common representation of cakes in the cake-cutting literature. It turns out that with respect to exploitability, it is the most interesting: for higher dimensions, we can design non-adaptive forced-cut protocols that use a linear number of cuts, but, as we show below, this is not the case when the cake is $(0, 1)$. In contrast, there exists an adaptive non-exploitable protocol that uses a linear number of cuts.

4.1 An adaptive non-exploitable 1-dimensional protocol

Protocol 4.1. *On day t , there is a single forced cut at h^t , whose location is adaptively determined in the manner described below. Given h^t , the cutter bisects the left part $(0, h^t)$ and right part $(h^t, 1)$ by cutting at ℓ^t and r^t respectively. The pieces are $X^t = (0, \ell^t) \cup (h^t, r^t)$, $\bar{X}^t = (\ell^t, h^t) \cup (r^t, 1)$.*

h^t is set adaptively as follows: $h^1 = 0.5$. For $t > 1$, there is some (non-atomic) interval (y^t, z^t) that is common to all of the chosen pieces on days $1, \dots, t-1$. The forced cut h^t is at $(y^t + z^t)/2$.

Before proving the protocol is non-exploitable, we first show that

Claim 4.2. *On day t , an interval (y^t, z^t) such as the one specified in the protocol always exists.*

Proof. The proof is by induction. The base case, $t = 2$ holds as the chooser chose some piece, and hence some slice that is a non-atomic interval.

For the inductive step, it holds by the inductive hypothesis that there is some interval (y^t, z^t) that is in all of the chosen piece on days $1, \dots, t-1$. On day t , the chooser either chose the piece containing the slice (ℓ^t, h^t) , in which case set $y^{t+1} = \max\{y^t, \ell^t\}, z^{t+1} = h^t$ or the piece containing the slice (h^t, r^t) , in which case set $y^{t+1} = h^t, z^{t+1} = \min\{z^t, r^t\}$. \square

Theorem 1.3. *Protocol 4.1 is non-exploitable and uses $3n$ cuts.*

Proof. Once again, we apply Lemma 2.3. From Claim 4.2, for any divisions and any choices, there is a slice Z that is in common to all chosen pieces. Let the chooser's utility function u be such that $u(Z) > 1/2$. \square

4.2 Lower bound on the number of cuts for non-adaptive non-exploitable forced-cut protocols

In this section we show that there is no forced-cut n -day non-exploitable protocol that uses $O(n)$ cuts. Recall that by the definition of forced-cut protocols, the cutter makes a single cut in each segment defined by the forced cuts. It is not difficult to adapt the proof to hold for protocols that either allow or require the cutter to use more than one cut per segment, by an extra application of the pigeonhole principle in the proof of Theorem 1.4.

Given a set of slices $\{X_{1,0}, X_{1,1}, \dots, X_{m,0}, X_{m,1}\}$, a *division string* is a binary string encoding to which piece each slice belongs. For example, assume that there is a single forced cut. The three cuts made are ℓ, h, r , from left to right, where h is the forced cut. Denote $a = (0, \ell)$, $b = (\ell, h)$, $c = (h, r)$, $d = (r, 1)$. $X = \{a\} \cup \{c\}$ (which implies $\bar{X} = \{b\} \cup \{d\}$) is denoted by the binary string 00, and $X = \{b\} \cup \{c\}$ by the binary string 10. Note that 00 and 11 amount to the same choice for the chooser, as they both lead to a choice between $\{a\} \cup \{c\}$ and $\{b\} \cup \{d\}$. There are 2^m

possible division strings, hence 2^{m-1} division strings that lead to different choices.

Before presenting the proof, we need a technical lemma and a simple claim.

Lemma 4.3. *Let $S_k = \{x = (x_1, \dots, x_k)\}$ be a set where each element x is an (unique, ordered) set of k monotone increasing real numbers in $(0, 1)$. If $|S_k| \geq 2^{\sum_{i=1}^k i^2}$, then there exist $x, y \in S_k$ such that $\forall i \neq j, x_i \neq y_j$.*

Proof. The proof is by induction. The base case $S_1 : |S_1| = 2$ is trivially true, as each set consists of a single number. For the inductive step, assume that the claim holds for all $\ell \leq k-1$, and assume towards contradiction that $|S_k| \geq 2^{\sum_{i=1}^k i^2}$ and for all $x, y \in S_k$, there are $i \neq j$ such that $x_i = y_j$. Fix some $x \in S_k$. Partition $S_k \setminus \{x\}$ into sets S_r as follows: $y \in S_r$ if r is the set of pairs of indices $\{(i, j)\}$ for which $x_i = y_j$. There are less than 2^{k^2} possible values of r . Therefore by the pigeonhole principle, there must be at least one such S_r whose cardinality is at least $2^{\sum_{i=1}^{k-1} i^2}$. Consider the elements of S_r . They all share at least one coordinate (i.e., $\exists \alpha : \forall y, y' \in S_r, y_\alpha = y'_\alpha$). Removing this coordinate leaves, for each member of S_r , a set of $k-1$ monotone increasing real numbers. By the inductive hypothesis, there are $y, y' \in S_r$ for which $\forall i \neq j, y_i \neq y'_j$. \square

Claim 4.4. *Let \mathcal{P} be a forced-cut protocol. If there are two days $x \neq y \in \{1, \dots, n\}$ whose forced cuts are $0 < x_1 < x_2 < \dots < x_k < 1$ and $0 < y_1 < y_2 < \dots < y_k < 1$ respectively such that $\forall i \neq j, x_i \neq y_j$ and their division string is identical, then \mathcal{P} is exploitable.*

Proof. Assume that there are σ values of i for which $x_i = y_i$ (possibly $\sigma = 0$). Let $\gamma_1, \dots, \gamma_\sigma$ be those values.

It suffices to show that on days x and y , the cutter can make divisions $\{X, \bar{X}\}$ and $\{Y, \bar{Y}\}$ respectively such that $X \subseteq Y$, as if the chooser chose X , she will surely choose Y . We prove this separately for every interval (γ_q, γ_{q+1}) , $q \in \{1, \dots, \sigma-1\}$ and take the union of these cuts. Henceforth, we focus a single interval (γ_q, γ_{q+1}) . As we are now only concerned with a single such interval, we assume w.l.o.g. that $\sigma = 0$ and simply consider the interval $(0, 1)$. Let $\epsilon > 0$ be the smallest distance between x_i and y_j , for all $i, j \in \{1, \dots, k\}$. Let the division string be s_1, \dots, s_{k+1} . For $i \in \{1, \dots, k+1\}$, if $s_i = 0$, cut at $x_{i-1} + \epsilon/2$ (hence $X_{i,0}^x = (x_{i-1}, x_{i-1} + \epsilon/2)$). If $s_i = 1$, cut at $x_i - \epsilon/2$. There are no y_j s in any of the slices that consist X , hence it is possible to divide $(0, 1)$ into two pieces $\{Y, \bar{Y}\}$ such that $X \subseteq Y$. \square

We are now ready to prove our lower bound.

Theorem 1.4. *Let $c \geq 1$ be a constant, and let \mathcal{P} be a non-adaptive forced-cut protocol that uses at most cn cuts. Then there exists a constant n_0 such that for every $n \geq n_0$, \mathcal{P} is exploitable.*

Proof. The proof is by contradiction. Assume that there exists a constant $c \geq 1$ such that there is a protocol that uses at most cn cuts for any n . Set $n_0 = 2^{8c^3+4c}$; choose $n \geq n_0$.

By Markov's inequality, on at least half of the days, the protocol uses at most $2c$ cuts. We focus on these days, and

henceforth assume that the cutter makes at most $2c$ cuts on each day, and the number of days is $n/2 > (2c+1) \cdot 2^{8c^3+2c}$.

We now apply the pigeonhole principle twice to show:

1. There is at least one k for which the cutter makes exactly k cuts, $0 \leq k \leq 2c$ on at least $2^{8c^3+2c} \geq 2^k 2^{k^3}$ days.
2. There are at least $2^{k^3} > 2^{\sum_{i=1}^k i^2}$ days that have the same number of cuts and the same division string, as there are 2^k possible division strings that lead to different choices.

By Lemma 4.3, there are at least 2 days for which the conditions of Claim 4.4 hold; Claim 4.4 completes the proof. \square

4.3 A non-adaptive non-exploitable 1-dimensional protocol

We consider the following non-adaptive n -day protocol:

Protocol 4.5. *On each day the forced cuts are $\{h_i = \frac{i}{n+1}\}$ for $i \in \{1, \dots, n\}$. We denote $h_0 = 0$ and $h_{n+1} = 1$. On day t , for $i \in \{1, \dots, n+1\}$, the cutter cuts segment the (h_{i-1}, h_i) at h_i^t , creating two slices $X_{i,0}^t = (h_{i-1}, h_i^t)$, $X_{i,1}^t = (h_i^t, h_i)$. The two pieces of the division on day t are $X^t = (\bigcup_{i \neq t} X_{i,0}^t) \cup X_{t,1}^t$ and $\bar{X}^t = (\bigcup_{i \neq t} X_{i,1}^t) \cup X_{t,0}^t$.*

$X_{1,0}^2$	$X_{1,1}^2$	$X_{2,0}^2$	$X_{2,1}^2$	$X_{3,0}^2$	$X_{3,1}^2$	$X_{4,0}^2$	$X_{4,1}^2$
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

Figure 1: The second day of executing Protocol 4.5 for $n = 3$. The red lines are the forced cuts (identical for all days). The blue lines are the cuts the cutter made on day 2; the chooser has to choose between $X^2 = X_{1,0}^2 \cup X_{2,1}^2 \cup X_{3,0}^2 \cup X_{4,0}^2$ (shaded) and its complement.

Theorem 1.5. *Protocol 4.5 is non-exploitable and uses $\Theta(n^2)$ cuts.*

Proof. Once again we make use of Lemma 2.3. Denote $h_i^{\min} = \min\{h_i^1, \dots, h_i^n\}$ and $h_i^{\max} = \max\{h_i^1, \dots, h_i^n\}$. For $t = 1, \dots, n$, set $s_t = 1$ if the chooser chose X^t and $s_t = 0$ if she chose \bar{X}^t . Let $S = |\{t : s_t = 1\}|$ be the number of days on which the chooser chose X^t .

Set $\epsilon = 1/n^2$. The chooser's utility function is the following. For all $i : s_i = 1$, set $u(h_i^{\max}, h_{i+1}) = \frac{1}{2S} + \epsilon$. Set $u(h_n, h_{n+1}^{\min}) = 1/2 - S\epsilon$. For all other $h_{i,j}$, set $u(h_{i,j}) = 0$. This implies $u(X_{i,1}^t) = \frac{1}{2S} + \epsilon$ and $u(X_{n+1,0}^t) = 1/2 - S\epsilon$ for all t . This utility function is consistent with all of the chooser's choices: on days t such that $s_t = 1$,

$$u(X^t) = \left(\bigcup_{i \neq t} X_{i,0}^t \right) \cup X_{t,1}^t = \frac{1}{2} + \frac{1}{2S} - (S-1)\epsilon > 1/2,$$

and on days t such that $s_t = 0$,

$$u(X^t) = \left(\bigcup_{i \neq t} X_{i,0}^t \right) \cup X_{t,1}^t = \frac{1}{2} - S\epsilon.$$

\square

5 2-dimensional cakes

Consider a square 2-dimensional cake $(0, 1) \times (0, 1)$. (It is easy to extend the results to any convex shape.)

All of our forced cuts are vertical, hence we can denote them by a single real number in $(0, 1)$.

Consider first the following n -day protocol.

Protocol 5.1. *On day t , there is forced cut $h^t = \frac{t}{n+1}$. This bisects the cake into two segments, X_0^t and X_1^t . The cutter cuts X_0^t horizontally into two slices: $X_{0,0}^t$, $X_{0,1}^t$; and X_1^t horizontally into: $X_{1,0}^t$, $X_{1,1}^t$. The two pieces are $X^t = X_{0,0}^t \cup X_{1,1}^t$ and $\bar{X}^t = X_{0,1}^t \cup X_{1,0}^t$.*

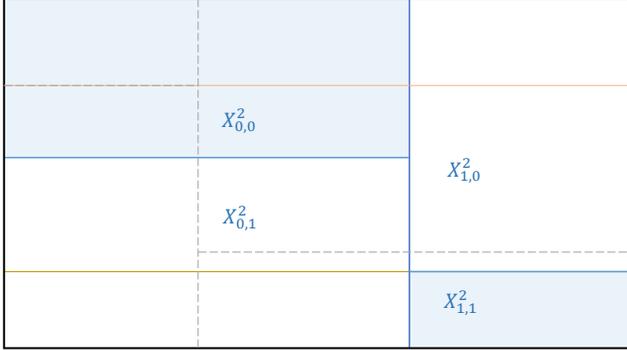


Figure 2: The second day of executing Protocol 5.1 for two days. The gray dotted lines are the cuts on day 1. The blue lines are the cuts on day 2; the chooser has to choose between $X^2 = X_{0,0}^2 \cup X_{1,1}^2$ (shaded) and $\bar{X}^2 = X_{0,1}^2 \cup X_{1,0}^2$. The orange lines represent h_{\min} and h_{\max} in the proof of Theorem 1.6.

Theorem 1.6. *Protocol 5.1 non-exploitable and uses $3n$ cuts.*

Proof. We apply Lemma 2.3. Let h_{\max} and h_{\min} denote the maximal and minimal unforced cuts that the cutter made on the n days (i.e., if the cut is denoted by a real number on the vertical axis, the largest and smallest such numbers respectively). Denote the choices by a string $s = (s_1, \dots, s_n)$, where $s_t = 0$ indicates that the chooser chose X^t on day t , and $s_t = 1$ indicates that she chose \bar{X}^t . We consider the following $3(n+1)$ rectangles: Y_{low}^t , Y_{mid}^t , and Y_{high}^t , $t \in \{1, \dots, n+1\}$:

- Y_{low}^t is the rectangle with bottom left corner at $(h^{t-1}, 0)$ and top right corner at (h^t, h_{\min}) .
- Y_{mid}^t is the rectangle with bottom left corner at (h^{t-1}, h_{\min}) and top right corner at (h^t, h_{\max}) .
- Y_{high}^t is the rectangle with bottom left corner at (h^{t-1}, h_{\max}) and top right corner at $(h^t, 1)$.

Set $u(Y_{\text{mid}}^t) = 0$ for all t . If $s_t = 0$, set $u(Y_{\text{low}}^t) = 0$, $u(Y_{\text{high}}^t) = 2^{(t-n-2)}$, otherwise ($s_t = 1$), set $u(Y_{\text{low}}^t) = 2^{(t-n-2)}$, $u(Y_{\text{high}}^t) = 0$.

Set

$$u(Y_{\text{low}}^{n+1}) = 1/2 - \left(\sum_{\tau=1}^n u(Y_{\text{low}}^\tau) \right),$$

$$u(Y_{\text{high}}^{n+1}) = 1/2 - \left(\sum_{\tau=1}^n u(Y_{\text{high}}^\tau) \right).$$

We need to verify that on each day, the chooser's utility for the chosen piece is indeed greater than for the other piece. Fix t , w.l.o.g. (from symmetry), assume that $s_t = 0$. We consider the minimum possible value that the chooser can have for X^t :

$$\begin{aligned} u(X^t) &= \sum_{\tau=1}^t u(Y_{\text{high}}^\tau) + \sum_{\tau=t+1}^{n+1} u(Y_{\text{low}}^\tau) \\ &\geq u(Y_{\text{high}}^t) + \left(1/2 - \sum_{\tau=1}^t u(Y_{\text{low}}^\tau) \right) \\ &\geq 2^{(t-n-2)} + 1/2 - \sum_{\tau=1}^{t-1} 2^{(\tau-n-2)} \\ &> 1/2 + 2^{(t-n-2)} - 2^{(t-n-2)} \\ &\geq u(\bar{X}^t), \end{aligned}$$

where the third inequality uses the fact that $u(Y_{\text{low}}^t) = 0$. This completes the proof. \square

6 Conclusion and open questions

The cut complexity of non-adaptive non-exploitable 1-dimensional forced-cut protocols remains open. In addition, it would be interesting to extend our work beyond worst-case analysis, when the agents have some partial prior knowledge about the other agents' utility, similarly to (Delgosha and Gohari 2012) (recall that they only analyze existing protocols and do not design new ones). It would also be interesting to study exploitability in other scenarios common in the cake-cutting literature: when there are more than 2 agents, with indivisible goods (as opposed to a divisible cake), and for other families of protocols, such as moving knife protocols, e.g., (Robertson and Webb 1998).

A Example

Example A.1. *Let the chooser's utility, u_{ch} , be concentrated in the interval $[0.4, 1]$, i.e., $u_{ch}([0.4, 1]) = 1$. Furthermore, $u_{ch}([0.4, 0.5]) = 1/2 - \epsilon$, $u_{ch}([0.5, 1]) = 1/2 + \epsilon$, uniform across both intervals. If the cutter cuts at 0.5 on day 1, and the chooser knows the cutter will move the cut 0.1 in the direction of her choice, it is unreasonable to assume that she would act myopically, as her total utility would be 1.1 instead of 1.5 if she lied on day 1.*

References

Aziz, H., and Mackenzie, S. 2016a. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 416–427.

- Aziz, H., and Mackenzie, S. 2016b. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, 454–464.
- Balkanski, E.; Brânzei, S.; Kurokawa, D.; and Procaccia, A. D. 2014. Simultaneous cake cutting. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 566–572.
- Barbanel, J. B.; Brams, S. J.; and Stromquist, W. 2009. Cutting a pie is not a piece of cake. *The American Mathematical Monthly* 116(6):496–514.
- Blum, A.; Dickerson, J. P.; Haghtalab, N.; Procaccia, A. D.; Sandholm, T.; and Sharma, A. 2015. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15*, 325–342.
- Brams, S. J., and Taylor, A. D. 1996. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press.
- Brams, S. J.; Jones, M. A.; and Klamler, C. 2006. Better ways to cut a cake. *Notices of the American Mathematical Society* 53(11):1314–1321.
- Chen, Y.; Lai, J. K.; Parkes, D. C.; and Procaccia, A. D. 2010. Truth, justice, and cake cutting. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.
- Delgosha, P., and Gohari, A. A. 2012. Information theoretic cutting of a cake. In *2012 IEEE Information Theory Workshop (ITW)*, 517–521.
- Dubins, L. E., and Spanier, E. H. 1961. How to cut a cake fairly. *American mathematical monthly* 1–17.
- Edmonds, J., and Pruhs, K. 2006. Cake cutting really is not a piece of cake. In *SODA*, 271–278.
- Even, S., and Paz, A. 1984. A note on cake cutting. *Discrete Applied Mathematics* 7:285–296.
- Fishburn, P. 1970. *Utility Theory for Decision Making*. Wiley, New York.
- Friedman, E. J.; Psomas, C.; and Vardi, S. 2015. Dynamic fair division with minimal disruptions. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15*, 697–713.
- Friedman, E. J.; Psomas, C.; and Vardi, S. 2017. Controlled dynamic fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC*, 461–478.
- Gilboa, I., and Schmeidler, D. 1989. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics* 18(2):141 – 153.
- Gupta, A.; Nagarajan, V.; and Singla, S. 2017. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 1688–1702.
- Iyer, K., and Huhns, M. N. 2009. A procedure for the allocation of two-dimensional resources in a multiagent system. *Int. J. Cooperative Inf. Syst.* 18(3-4):381–422.
- Knaster, B. 1944. Sur le problème du partage pragmatique de h. steinhaus. *Ann. Soc. Polonaise Math.* 19:228–231.
- Mossel, E., and Tamuz, O. 2010. Truthful fair division. In *Algorithmic Game Theory - Third International Symposium, SAGT*, 288–299.
- Neyman, J. 1946. Un théorème d’existence. *C.R. Acad. Sci. Paris* 222:843–845.
- Procaccia, A. D., and Wang, J. 2017. A lower bound for equitable cake cutting. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC*, 479–495.
- Procaccia, A. D. 2009. Thou shalt covet thy neighbor’s cake. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 239–244.
- Procaccia, A. D. 2016. Cake cutting algorithms. In *Handbook of Computational Social Choice*. Cambridge University Press. 311–330.
- Robertson, J. M., and Webb, W. A. 1998. *Cake Cutting Algorithms: be fair if you can*. AK Peters.
- Segal-Halevi, E.; Nitzan, S.; Hassidim, A.; and Aumann, Y. 2017. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics* 70:1 – 28.
- Segal-Halevi, E. 2016. How to re-divide a cake fairly. *arXiv preprint arXiv:1603.00286*.
- Sinn, H.-W. 2012. *Economic decisions under uncertainty*. Springer Science & Business Media.
- Steinhaus, H. 1948. The problem of fair division. *Econometrica* 16(1).
- Stromquist, W. 1980. How to cut a cake fairly. *American Mathematical Monthly* 640–644.
- Varian, H. R. 1974. Equity, envy, and efficiency. *Journal of Economic Theory* 9(1):63–91.
- Wald, A. 1945. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics* 46(2):265–280.
- Wald, A. 1949. Statistical decision functions. *Ann. Math. Statist.* 20(2):165–205.
- Walsh, T. 2011. Online cake cutting. In *Algorithmic Decision Theory*. Springer. 292–305.
- Woeginger, G., and Sgall, J. 2007. On the complexity of cake cutting. *Discrete Optimization* 4(2):213–220.