On the Complexity of Cake Cutting

Gerhard J. Woeginger *   Jiří Sgall†

June 2, 2004

Abstract

In the cake cutting problem, $n \geq 2$ players want to cut a cake into n pieces so that every player gets a 'fair' share of the cake by his own measure. Two results are given.

A positive result: For every $\varepsilon > 0$, there exists a cake division scheme for n players that uses at most $c_\varepsilon n$ cuts (with free evaluation queries), and in which each player can enforce to get a share of at least $(1-\varepsilon)/n$ of the cake according to his own private measure.

A negative result: In a certain cake cutting model, every fair cake division protocol for n players must use $\Omega(n \log n)$ cuts and evaluation queries in the worst case. Up to a small constant factor, this lower bound matches a corresponding upper bound in the same model by Even & Paz from 1984.

1 Introduction

The poem *"The voice of the lobster"* by Lewis Carroll [2] gives a classical example for the unfair division of a common resource:

"I passed by his garden, and marked, with one eye,
How the owl and the panther were sharing a pie.
The panther took pie-crust, and gravy, and meat,
While the owl had the dish as its share of the treat."

Note that pie-crust, gravy, and meat might be of completely different value to the owl and to the panther. Is there any protocol which enables owl and panther to divide the food into two pieces such that both will get at least half of it by their own measure? The answer to this question is yes, and there is a fairly simple and fairly old solution described by Hugo Steinhaus [10] from 1948: The owl cuts the food into two pieces, and the panther chooses its piece out of the two. The owl is sure to get at least half the food if it cuts two equal pieces by its measure. The panther is sure to get at least half the food by its measure by choosing the better half.

*Email: gwoegi@win.tue.nl. Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

†Email: sgall@math.cas.cz. Mathematical Institute of the Academy of Sciences of the Czech Republic, Žitná 25, CZ-11567 Praha 1, The Czech Republic. Partially supported by Institute for Theoretical Computer Science, Prague (project LN00A056 of MŠMT ČR) and grant IAA1019401 of GA AV ČR.

Problem description. In a more general and a more mathematical formulation, there is a certain resource $\mathcal{C}$ (hereinafter referred to as: *the cake*), and there are n players $1, \ldots, n$. Without much loss of generality and in agreement with the cake cutting literature, we will assume throughout the paper that $\mathcal{C} = [0, 1]$ is the unit interval and the cuts divide the cake into its subintervals. Every player p ($1 \leq p \leq n$) has his own private measure μ_p on sufficiently many subsets of $\mathcal{C}$. These measures μ_p are assumed to be well-behaved; this means that they are:

- Defined on all finite unions of intervals.
- Non-negative: For all $X \subseteq \mathcal{C}$, $\mu_p(X) \geq 0$.
- Additive: For all disjoint subsets $X, X' \subseteq \mathcal{C}$, $\mu_p(X \cup X') = \mu_p(X) + \mu_p(X')$
- Divisible: For every measurable $X \subseteq \mathcal{C}$ and $0 \leq \lambda \leq 1$, there exists $X' \subseteq X$ with $\mu_p(X') = \lambda\, \mu_p(X)$.
- Normalized: $\mu_p(\mathcal{C}) = 1$.

All these assumptions are standard assumptions in the cake cutting literature, sometimes subsumed in a concise statement that each μ_p is a probability measure defined on Lebesgue measurable sets and absolutely continuous with respect to Lebesgue measure. We stress that the divisibility of μ_p forbids concentration of the measure in one or more isolated points. As one consequence of this, corresponding open and closed intervals have the same measure, and thus we do not need to be overly formal about the endpoints of intervals.

The cake $\mathcal{C}$ is to be divided among the n players according to some fixed *cake division protocol*, which is an interactive procedure for the players that guides and controls the division process of the cake $\mathcal{C}$. Typically it consists of cut requests like *"Cut cake piece Z into two equal pieces, according to your measure!"* and evaluation queries like *"Is your measure of cake piece Z_1 less, greater, or equal to your measure of cake piece Z_2?"* or *"What is your measure of cake piece Z_1?"*. Eventually, the protocol assigns to the players pairwise disjoint subsets of the cake. A cake division protocol is not a priori aware of the measures μ_p, but it will learn something about them during its execution, and the answers may influence which cut requests and evaluation queries are issued in the future (i.e., the protocol is adaptive). A *strategy* of a player is an adaptive sequence of moves consistent with a given protocol.

A cake division protocol is *(perfectly) fair*, if every player p has a strategy that guarantees him a piece of size at least $\mu_p(\mathcal{C})/n$ according to his own measure μ_p. So, even in case $n-1$ players would all plot up against a single player and would coordinate their moves, then this single player will still be able to get his share of $\mu_p(\mathcal{C})/n$. This is called *simple fair division* in the literature.

We also consider approximately fair protocols. A cake division protocol is *β-fair*, if every player p has a strategy that guarantees him a piece of size at least $\beta\mu_p(\mathcal{C})$ according to his own measure μ_p. Note that a protocol for n players is perfectly fair if and only if it is $1/n$-fair.

The complexity of a cake cutting protocol is generally measured by the number of cuts performed in the worst case. We discuss the exact definition of this measure and the notion of a cake cutting protocol later.

The central open problem in this area is whether there exist perfectly fair n-player protocols that only use $O(n)$ cuts. This problem was explicitly formulated by Even & Paz [3], and essentially goes back to Steinhaus [10]. The general belief is that no such protocol exists.

Previous results. In the 1940s, the Polish mathematicians Banach and Knaster designed a simple fair cake division protocol that uses $O(n^2)$ cuts in the worst case; this protocol was explained and discussed in 1948 by Steinhaus [10]. In 1984, Even & Paz [3] used a divide-and-conquer approach to construct a better deterministic protocol that only uses $O(n \log n)$ cuts in the worst case. Remarkably, Even & Paz [3] also design a randomized protocol that uses an expected number of $O(n)$ cuts.

Tighter results are known for small values of n: For $n = 2$ players, the Steinhaus protocol yields a perfectly fair protocol with a single cut. For $n = 3$ and $n = 4$ players, Even & Paz [3] present perfectly fair protocols that make at most 3 and 4 cuts, respectively. Webb [12] presents a perfectly fair protocol for $n = 5$ players with 6 cuts.

Approximately fair protocols were studied by Robertson & Webb [7]. For any $n \geq 2$, they design $1/(2n-2)$-fair protocols that make only $n-1$ cuts, and they show that this result is best possible for $n-1$ cuts. This result was rediscovered independently by Krumke et al [4].

The problem of establishing lower bounds for cake cutting goes back at least to Banach (see [10]). Even & Paz [3] explicitly conjecture that there does not exist a fair deterministic protocol with $O(n)$ cuts. Robertson & Webb [8] support and strengthen this conjecture by saying they "would place their money against finding a substantial improvement on the $n \log_2 n$ [upper] bound".

One basic difficulty in proving lower bounds for cake cutting is that most papers derive upper bound results and to do that, they simply describe a certain procedure that performs certain steps, and then establish certain nice properties for it, but they do not provide a formal definition or a framework. Even & Paz [3] give a proof that for $n \geq 3$, no protocol with $n-1$ cuts exists; since $n-1$ cuts are the smallest possible number, such protocols would need to be rather special (in particular they would have to assign a single subinterval to each player) and not much formalism is needed. Webb [12] shows that for $n = 5$, no perfectly fair protocol exists that uses only 5 cuts, and thus his protocol with 6 cuts is optimal.

Only recently, Robertson & Webb [7, 8] give a more precise definition of a protocol that covers all the protocols given in the literature. This definition avoids some pathological protocols, but it is still quite general and no super-linear lower bounds are known.

A recent paper [5] by Magdon-Ismail, Busch & Krishnamoorthy proves an $\Omega(n \log n)$ lower bound for a certain non-standard cake cutting model: The lower bound does not hold for the number of performed cuts or evaluation queries, but for the number of comparisons needed to administer these cuts.

For more information on this fair cake cutting problem and on many of its variants, we refer the reader to the books by Brams & Taylor [1] and by Robertson & Webb [8].

Our contribution and organization of the paper. In Section 2 we formally define first the Robertson-Webb cake cutting model and then we introduce a certain restriction of this model. The restriction has two requirements: (i) each player receives a single subinterval of the cake and (ii) the evaluation queries are counted towards the complexity of the protocol together with cuts. Our model is also general enough to cover the $O(n \log n)$ cut deterministic protocol of Even & Paz [3], and we believe that it is fairly natural. We discuss some of the issues related to the formal definition of both models, and we put it into context with other results from the cake cutting literature.

In Section 3 we design protocols with $O(n)$ cuts that come arbitrarily close to being $1/n$-fair. Namely, for every $\varepsilon > 0$, there exists a $(1-\varepsilon)/n$-fair cake division protocol for n players such that at most $O(n)$ cuts are made. This protocol is in the Robertson-Webb model, thus we do not count the evaluation queries towards its complexity (their number is $O(n^2)$). On the other hand, the protocol still obeys the first requirement in the definition of restricted protocols, i.e., each player receives a single subinterval of the cake.

In Section 4 we then show that in our restricted model, every deterministic fair cake division protocol for n players must use $\Omega(n \log n)$ cuts and evaluation queries in the worst case. This result yields the first super-linear lower bound on the number of cuts for simple fair division (in our restricted model), and it also provides a matching lower bound for the result in [3].

Section 5 gives the discussion and open problems.

This paper covers the results of extended abstracts [13, 9].

2 The restricted cake cutting model

A general assumption in the cake cutting literature is that at the beginning of an execution a protocol has absolutely no knowledge about the measures μ_p, except that they are defined on intervals, non-negative, additive, divisible, and normalized. The protocol issues queries to the players, the players react, the protocols observes their reactions, issues more queries, observes more reactions, and so on, and so on, and in the end the protocol assigns the cake pieces to the players.

Definition of the Robertson-Webb model and our restricted model. We recall that the cake $\mathcal{C}$ is represented by the unit interval. For a real number α with $0 \leq \alpha \leq 1$, the α-point of a player p is the infimum of all numbers x for which $\mu_p([0,x]) = \alpha$ and $\mu_p([x,1]) = 1 - \alpha$ holds.

In the *Robertson-Webb model*, the following two types of queries are allowed.

> Cut$(p; \alpha)$: Player p cuts the cake at his α-point (where $0 \leq \alpha \leq 1$). The value x of the α-point is returned to the protocol.
>
> Eval$(p; x)$: Player p evaluates the value of the cut x, where x is one of the cuts previously performed by the protocol. The value $\mu_p(x)$ is returned to the protocol.

The protocol can also assign an interval to a player; by doing this several times, a player may end up with a finite union of intervals.

> Assign$(p; x_i, x_j)$: Player p is assigned the interval $[x_i, x_j]$, where $x_i \leq x_j$ are two cuts previously performed by the protocol or 0 or 1. All the intervals assigned to the players are required to be pairwise disjoint.

The complexity of a protocol is given by the number of cuts performed in the worst case, i.e., evaluation queries may be issued for free.

In *our restricted model*, the additional two restrictions are:

> Assign$(p; x_i, x_j)$ is used only once for each p. Hence, in the restricted model every player ends up with a *single* (contiguous) subinterval of the cake.

The complexity of a protocol is given by the number of cuts plus evaluation queries, i.e., each evaluation query contributes to the complexity the same as a cut. Note that this also covers counting only the number of cuts in protocols that do not use evaluation queries at all.

Discussion of the restricted model. The currently best deterministic protocol for exact fair division of Even & Paz [3] does not need evaluation queries and assigns single intervals; we provide a matching bound within these restrictions.

Nevertheless, both restrictions of our model are essential. Our protocol from Section 3 as well as protocols in [4, 6, 7], especially those that achieve not exactly but only approximately fair division, do use evaluation queries, sometimes even a quadratic number of them. The randomized protocol of Even & Paz [3] also uses evaluation queries in addition to expected $O(n)$ cuts; the expected number of evaluation queries is $\Theta(n \log n)$.

We feel that the other restriction, that every player must receive a single, contiguous subinterval of the cake, is perhaps even stronger. By imposing this restriction, it seems that we severely cut down the set of possible protocols; in particular, for some instances, the solution is essentially unique (see our lower bound). Note, however, that *all* known discrete cake cutting protocols from the literature produce solutions where every player ends up with a contiguous subinterval. For instance, all the protocols in [3, 4, 6, 7, 10, 12] as well as our protocol from Section 3 have this property. In particular, the divide-and-conquer protocols of Even & Paz [3], both deterministic and randomized, assign single contiguous subinterval to each player, as noted above.

Discussion of the Robertson-Webb model. The Robertson-Webb model restricts the format of queries to cuts at α points and evaluation queries. This restriction is severe, but it is crucial and essentially unavoidable. Such a restriction must be imposed in one form or the other, just to prevent certain uninteresting types of 'cheating' protocols from showing up with a linear number of cuts. Consider the following 'cheating' protocol:

> **Phase 1.** Every player makes a cut that encodes his i/n-points with $1 \leq i \leq n-1$ (just fix any bijective encoding of $n-1$ real numbers from $[0,1]$ into a single number from $[0,1]$).
>
> **Phase 2.** The protocol executes the Banach-Knaster protocol in the background (Banach-Knaster [10] is a fair protocol that only needs to know the positions of the i/n-points). That is, the protocol determines the relevant cuts without performing them.
>
> **Phase 3.** The protocol tells the players to perform the relevant $n-1$ cuts for the Banach-Knaster solution. If a player does not perform the cut that he announced during the first phase, he is punished and receives an empty piece (and his piece is added to the piece of some other player).

Clearly, every honest player will receive a piece of size at least $1/n$. Clearly, the protocol also works in the friendly environment where every player truthfully executes the orders of the protocol. And clearly, the protocol uses only $2n-1$ cuts—a linear number of cuts. Moreover, there are (straightforward) implementations of this protocol where every player ends up with a single subinterval of the cake. In cake cutting models that allow announcements of arbitrary real numbers, the cuts in (Phase 1) can be replaced by direct announcements of the i/n-point positions; this yields fair protocols with only $n-1$ cuts.

These 'cheating' protocols are artificial, unnatural and uninteresting, and it is hard to accept them as valid protocols. In the Robertson-Webb model they cannot occur, since they violate the form of queries. (One could try to argue that the players might disobey the queries and announce any real number. However, this fails, since the definition of a protocol enforces that a player that honestly answers allowed queries should get a fair share.)

Second important issue is that in the Robertson-Webb model it is sufficient to assume that all players are honest, i.e., execute the commands "Cut at an α-point" and evaluation queries truthfully. Under this assumption all of them get a fair share. Often in the literature, a protocol has no means of enforcing a truthful implementation of these cuts by the players, since the players may cheat, and lie, and try to manipulate the protocol; the requirement is than that any honest player gets a fair share, regardless of the actions of the other players. In the Robertson-Webb model, any protocol that works for honest players can be easily modified to the general case as follows. As long as the answers of a player are consistent with some measure, the protocol works with no change, as it assigns a fair share according to this measure (and if the player has a different measure, he lied and has no right to complain). If an inconsistency is revealed (e.g., a violation of non-negativity), the protocol has to be modified to ignore the answers from this player (or rather replace them by some trivial consistent choices).

Of course, in general, the honesty of players is not a restriction on the protocol, but a restriction on the environment. Thus it is of no concern for our lower bound argument which uses only honest players. On the other hand, it simplifies the presentation of protocols; compare, for example, the presentation of the same approximately fair protocol in Section 3 and in the conference paper [13].

In some details our description of the model is different than that of Robertson & Webb. Their formulation in place of evaluation queries is that after performing the cut, its value in all the players' measures becomes known. This covers all the possible evaluation queries, so it is clearly equivalent if we do not count the number of these queries. However, the number of evaluations is an interesting parameter, which is why we chose this formulation.

Robertson & Webb also allow cut requests of the form "cut this piece into two pieces with a given ratio of their measures". This is very useful for an easy formulation of recursive divide-and-conquer protocols. Again, once free evaluation queries are allowed, this is no more general, as we know all the measures of all the existing pieces. Even if we count evaluation queries, we can first evaluate the cuts that created the piece, so such a non-standard cut is replaced by two evaluations and a standard cut at some α-point.

Finally, instead cutting at the α-point, Robertson & Webb allow an honest player to return any x with $\mu_p([0,x]) = \alpha$, i.e., we require the answer which is the minimum of the honest answers according to Robertson & Webb. This is a restriction if the instance contains non-trivial intervals of measure zero for some players, otherwise the answer is unique. However, any such instance can be replaced by a sequence of instances with measures that are very close to the original ones and have non-zero density everywhere. If done carefully, all the α-points in the sequence of modified instances converge to the α-points in the original instance. Thus the restriction to a particularly chosen honest answer is not essential as well; on the other hand, it keeps the description of our lower bound much simpler.

3 The approximately fair protocol

In this section we present the cake division protocol that uses only a linear number of cuts and achieves an approximately fair solution.

Theorem 1 *For every $\varepsilon > 0$, there exists a constant $c_\varepsilon > 0$ and a $(1-\varepsilon)/n$-fair cake division protocol for n players such that at most $c_\varepsilon n$ cuts are made. The number of evaluation queries is $O(n^2)$.*

We define a recursive protocol $P(t)$ that is based on an integer parameter $t \geq 1$; later we set t to be about $1/\varepsilon$. The steps (S0)–(S5) of protocol $P(t)$ are described in Figure 1.

(S0) If there are $n \leq 2t-1$ players, the cake is divided using the deterministic protocol of Even & Paz. STOP.

(S1) Each of the first $2t$ players p ($p = 1, \ldots, 2t$) makes a cut $c_p = \mathrm{Cut}(p; 1/2)$.

(S2) Let c^* be the median of the numbers $c_1, \ldots, c_{2t}$, i.e., the t-th smallest number among them. The cut c^* divides the cake $\mathcal{C} = [0,1]$ into a left piece $\mathcal{C}_L = [0, c^*]$ and a right piece $\mathcal{C}_R = [c^*, 1]$.

(S3) Every player $p = 1, \ldots, n$ answers an evaluation query $x_p = \mathrm{Eval}(p, c^*)$.

(S4) Sort the numbers x_p; let π be a permutation such that $x_{\pi(1)} \geq x_{\pi(2)} \geq \cdots \geq x_{\pi(n)}$. Let i be the smallest index such that $x_{\pi(i)} \leq \frac{i-1}{n-1}$. (This is defined, as $x_{\pi(n)} \leq 1$.) Let $L = \{\pi(1), \pi(2), \ldots, \pi(i-1)\}$ and $R = \{\pi(i), \ldots, \pi(n)\}$.

(S5) The players in L recursively share the left piece $\mathcal{C}_L$. The players in R recursively share the right piece $\mathcal{C}_R$.

Figure 1: The protocol $P(t)$ for n players.

As presented, it is assumed that the cake and the measures of the players are renormalized after each recursive call. Since the cake is divided at a point c^* such that we know all the values $\mathrm{Eval}(p; c^*)$, it is easy to implement this.

Note that the sets L and R in step (S5) give a partition of all the players. The next lemma guarantees that both sets in the partition are sufficiently large, and that each player considers the piece of the cake to which it is assigned in step (S6) sufficiently large, given how many people will split it.

Lemma 2 *The sets L and R chosen in step (S4) satisfy:*

(i) $|L| \geq t$ *and* $|R| \geq t$.

(ii) For every player $p \in L$, $x_p \geq \dfrac{|L|-1}{n-1}$.

(iii) For every player $p \in R$, $1 - x_p \geq \dfrac{|R|-1}{n-1}$.

Proof. By the choice of c^* in step (S2), there are t players p such that $x_p \geq 1/2$ (even among the first $2t$ players). Consequently, for $j \leq t$, we have $x_{\pi(j)} \geq 1/2 > (j-1)/(n-1)$, as $n \geq 2t \geq 2j$; we conclude that i in step (S4) is at least $t+1$ and thus $|L| \geq t$. Similarly, there are t players p such that $x_p \leq 1/2$. Thus $x_{\pi(n-t+1)} \leq 1/2 < (n-t)/(n-1)$, as $n \geq 2t$; we conclude that i in step (S4) is at most $n-t+1$ and thus $|R| \geq t$.

For $p \in L$, using the definition of i in step (S4), we have $x_p \geq x_{\pi(|L|)} > (|L|-1)/(n-1)$

For $p \in R$, using the definition of i in step (S4), we have $x_p \leq x_{\pi(|L|+1)} \leq |L|/(n-1)$. Thus $1 - x_p \geq 1 - |L|/(n-1) = (|R|-1)/(n-1)$ ∎

Next we analyze the fairness of the protocol.

Lemma 3 *Let $n \geq t$. Then every player p is assigned at least a fraction $(t-1)/(t(n-1))$ of the cake $\mathcal{C}$. Consequently, the protocol $P(t)$ is $(1-1/t)/n$-fair.*

Proof. The proof of the first claim is by induction on n. For $n \leq 2t-1$, step (S0) applies and p gets a fraction of at least $1/n \geq (t-1)/(t(n-1))$, using $t \geq n$ to obtain the inequality.

For $n \geq 2t$, by Lemma 2(i), $|L| \geq t$ and $|R| \geq t$ in step (S4), thus we can use the induction assumption for the recursive invocations of the protocol. Note also that by definition $x_p = \mu_p(\mathcal{C}_L)$ and $1 - x_p = \mu_p(\mathcal{C}_R)$. We distinguish two cases.

If in step (S4) the protocol assigns player p to L, then, using the inductive assumption and Lemma 2(ii), player p receives at least

$$\frac{t-1}{t(|L|-1)} \cdot x_p \geq \frac{t-1}{t(|L|-1)} \cdot \frac{|L|-1}{n-1} = \frac{t-1}{t(n-1)}.$$

If in step (S4) the protocol assigns player p to R, then, using the inductive assumption and Lemma 2(iii), player p receives at least

$$\frac{t-1}{t(|R|-1)} \cdot (1-x_p) \geq \frac{t-1}{t(|R|-1)} \cdot \frac{|R|-1}{n-1} = \frac{t-1}{t(n-1)}.$$

This completes the inductive proof of the first claim.

To conclude that the protocol is $(1-1/t)/n$-fair, note that $(t-1)/(t(n-1)) \geq (1-1/t)/n$ for $n \geq t$. In $n < t$, step (S0) applies and thus the protocol is $1/n$-fair. ∎

It remains to examine the complexity of the protocol.

Lemma 4 *If the cake is divided among n players according to protocol $P(t)$, then the players altogether make at most $2t(n-1)$ cuts. The number of evaluation queries is at most n^2/t.*

Proof. By induction on the number n of players. If $n \leq 2t-1$, protocol $P(t)$ becomes the Even & Paz protocol. Hence, there are at most $n \log_2 n \leq 2t(n-1)$ cuts and no evaluation queries. For $n \geq 2t$, there are $2t$ cuts made in step (S1). Moreover, by the inductive assumption there are at most $2t(|L|-1)$ and at most $2t(|R|-1)$ cuts made in the recursion in step (S6). Altogether, this yields at most $2t(|L|+|R|-1) = 2t(n-1)$ cuts. There are n evaluation queries in step (S3), so by induction, the number of evaluation queries is $n + (|L|^2 + |R|^2)/t \leq n^2/t$, using $|L|, |R| \geq t$ and $|L| + |R| = n$. ∎

Finally, let us prove Theorem 1. We use the protocol $P(t)$ with $t = \lceil 1/\varepsilon \rceil$. By Lemma 3, the protocol is $(1-\varepsilon)/n$-fair. By Lemma 4, the total number of cuts is at most $2\lceil 1/\varepsilon \rceil(n-1)$ and hence grows linearly in the number n of players; in addition the number of evaluation queries is quadratic in n.

4 The proof of the lower bound

In this section, we will prove the following theorem by means of an adversary argument in a decision tree.

Theorem 5 *In the restricted cake cutting model of Section 2 (where each player is assigned a single interval), every deterministic fair cake division protocol for n players uses at least $\Omega(n \log n)$ cuts and evaluation queries in the worst case.*

The adversary continuously observes the actions of the deterministic protocol, and he reacts by fixing the measures of the players appropriately.

Let us start by describing the specific cake measures μ_p that the we uses in the input instances. Let $\varepsilon < 1/n^4$ be some small positive real number. For $i = 1, \ldots, n$ we denote by $X_i \subset [0, 1]$ the set consisting of the n points $i/(n+1) + k\varepsilon$ with $1 \leq k \leq n$. Moreover, we let $X = \bigcup_{0 \leq i \leq n} X_i$. For $p = 1, \ldots, n$, by definition the player p has his 0-point at position 0. The positions of the i/n-points with $1 \leq i \leq n$ are fixed by the adversary during the execution of the protocol: The i/n-points of all players are taken from X_i, and distinct players receive distinct i/n-points. As one consequence, all the i/n-points of all players will lie strictly to the left of all the $(i+1)/n$-points of all players.

All the cake value for player p is concentrated in tiny intervals $\mathcal{I}_{p,i}$ of length ε that are centered around his i/n-points: For $i = 0, \ldots, n$, the measure of player p has a sharp peak with value $i/(n^2+n)$ immediately to the left of his i/n-point and a sharp peak with value $(n-i)/(n^2+n)$ immediately to the right of his i/n-point. Note that the measure between the i/n-point and the $(i+1)/n$-point indeed adds up to $1/n$. Moreover, the measures of the two peaks around every i/n-point add up to $1/(n+1)$, and the intervals that support these peaks for different players are always disjoint, with the exception of the intervals $\mathcal{I}_{p,0}$ that are the same for all the players. We do not explicitly describe the shape of the peaks; it can be arbitrary, but determined in advance and the same for each player.

For every player p, the portions of the cake between interval $\mathcal{I}_{p,i}$ and interval $\mathcal{I}_{p,i+1}$ have measure 0 and hence are worthless to p. By our definition of α-points, every α-point of player p will fall into one of his intervals $\mathcal{I}_{p,i}$ with $0 \leq i \leq n$. If a player p cuts the cake at some point $x \in \mathcal{I}_{p,i}$, then we denote by $c_p(x)$ the corresponding i/n-point of player p.

Lemma 6 *Let x be a cut that was done by player s, and let $y \geq x$ be another cut that was done by player t. Let $\mathcal{J} = [x, y]$ and $\mathcal{J}' = [c_s(x),\ c_t(y)]$. If $\mu_p(\mathcal{J}) \geq 1/n$ holds for some player p, then also $\mu_p(\mathcal{J}') \geq 1/n$.*

Proof. (Case 1) If $s = p$ and $t = p$, then let $\mathcal{I}_{p,j}$ and $\mathcal{I}_{p,k}$ be the intervals that contain the points $c_p(x)$ and $c_p(y)$, respectively. Then $\mu_p(\mathcal{J}) \geq 1/n$ implies $k \geq j+1$. The measure $\mu_p(\mathcal{J}')$ is at least the measure $(n-j)/(n^2+n)$ of the peak immediately to the right of the j/n-point plus the measure $k/(n^2+n)$ immediately to the left of the k/n-point, and these two values add up to at least $1/n$.

(Case 2) If $s = p$ and $t \neq p$, then let $\mathcal{I}_{p,j}$ be the interval that contains $c_p(x)$. Then $\mu_p(\mathcal{J}) \geq 1/n$ implies that $\mathcal{J}$ and $\mathcal{J}'$ both contain $\mathcal{I}_{p,j+1}$, and again $\mu_p(\mathcal{J}')$ is at least $1/n$. Note that the argument works also if $j = 0$.

(Case 3) The case $s \neq p$ and $t = p$ is symmetric to the second case above.

(Case 4) If $s \neq p$ and $t \neq p$, then the interval between x and $c_s(x)$ and the interval between y and $c_t(y)$ both have measure 0 for player p. By moving these two cuts, we do not change the value of $\mathcal{J}$ for p. ∎

We call a protocol *primitive*, if in all of its cut operations $\text{Cut}(p;\alpha)$ the value α is of the form i/n with $0 \le i \le n$.

Lemma 7 *For every protocol $\mathcal{P}$ in the restricted model, there exists a primitive protocol $\mathcal{P}'$ in the restricted model, such that for every cake cutting instance I of the restricted form described above,*

- *$\mathcal{P}$ and $\mathcal{P}'$ make the same number of cuts on I,*
- *if $\mathcal{P}$ applied to instance I assigns to player p a piece $\mathcal{J}$ of measure $\mu_p(\mathcal{J}) \ge 1/n$, then also $\mathcal{P}'$ applied to instance I assigns to player p a piece $\mathcal{J}'$ of measure $\mu_p(\mathcal{J}') \ge 1/n$.*

Proof. Protocol $\mathcal{P}'$ imitates protocol $\mathcal{P}$. Whenever $\mathcal{P}$ requests player p to cut at his α-point x with $0 < \alpha < 1$, then $\mathcal{P}'$ computes the unique integer k with

$$\frac{k}{n+1} \;<\; \alpha \;\le\; \frac{k+1}{n+1}$$

Then $\mathcal{P}'$ requests player p to cut the cake at his k/n-point. Note that by the choice of k, this k/n-point equals $c_p(x)$. The value of the cuts at x and $c_p(x)$ is the same for all the players other than p, thus any following answer to an evaluation query is the same in $\mathcal{P}'$ and $\mathcal{P}$. Furthermore, since the shape of the peaks is predetermined and the same for all the players, from the cut of $\mathcal{P}'$ at $c_p(x)$ we can determine the original cut of $\mathcal{P}$ at x. Consequently $\mathcal{P}'$ can simulate all the decisions of $\mathcal{P}$. When assigning pieces, each original cut x of $\mathcal{P}$ is replaced by the corresponding cut $c_p(x)$ of $\mathcal{P}'$. Clearly, both protocols make the same number of cuts, and Lemma 6 yields that if $\mathcal{P}$ is fair, then also $\mathcal{P}'$ is fair. ∎

Hence, from now on we may concentrate on some fixed primitive protocol $\mathcal{P}^*$, and on the situation where all cuts are from the set X. The strategy of the adversary is based on a permutation π of the integers $1, \ldots, n$; this permutation π is kept secret and not known to the protocol $\mathcal{P}^*$.

Now assume that at some point in time protocol $\mathcal{P}^*$ asks player p to perform a cut at his i/n-point. Then the adversary fixes the measures as follows:

- If $\pi(p) < i$, then the adversary assigns the i/n-point of player p to the smallest point in the set X_i that has not been used before.
- If $\pi(p) > i$, then the adversary assigns the i/n-point of player p to the largest point in the set X_i that has not been used before.
- If $\pi(p) = i$, then the adversary assigns the i/n-point of player p to the ith smallest point in the set X_i.

Consequently, any possible assignment of i/n-points to points in X_i has the following form: The player q with $\pi(q) = i$ sits at the ith smallest point. The $i-1$ players with $\pi(p) \le i-1$ are at the first (smallest) $i-1$ points, and the $n-i$ players with $\pi(p) \ge i+1$ are at the last (largest) $n-i$ points. The precise ordering within the first $i-1$ and within the last $n-i$ players depends on the behavior of the protocol $\mathcal{P}^*$. When protocol $\mathcal{P}^*$ terminates, then the adversary fixes the ordering of the remaining i/n-points arbitrarily (but in agreement with the above rules).

Lemma 8 *If $\pi(p) \leq i \leq \pi(q)$ and $p \neq q$, then in the ordering fixed by the adversary the i/n-point of player p strictly precedes the i/n-point of player q.*

Proof. Immediately follows from the adversary strategy above. ■

If the protocol $\mathcal{P}^*$ asks a player p an evaluation query on an existing cut at i/n-point of player p', the current assignment of i/n-points to points in X_i and the permutation π determine if the i/n-point of player p is smaller or larger than that of p' (for all the possible resulting assignment obeying the rules above). This is all that is necessary to determine the value of the cut, and thus the adversary can generate an honest answer to the query.

At the end, the primitive protocol $\mathcal{P}^*$ must assign intervals to players: $\mathcal{P}^*$ selects $n-1$ of the performed cuts, say the cuts at positions $0 \leq y_1 \leq y_2 \leq \cdots \leq y_{n-1} \leq 1$; moreover, we define $y_0 = 0$ and $y_n = 1$. Then for $i = 1, \ldots, n$, the interval $[y_{i-1}, y_i]$ goes to player $\phi(i)$, where ϕ is a permutation of $1, \ldots, n$.

Lemma 9 *If the primitive protocol $\mathcal{P}^*$ is fair, then*

(a) $y_i \in X_i$ holds for $1 \leq i \leq n-1$.

(b) The interval $[y_{i-1}, y_i]$ contains the $(i-1)/n$-point and the i/n-point of player $\phi(i)$, for every $1 \leq i \leq n$.

Proof. (a) If y_1 is at an $0/n$-point of some player, then $y_1 = 0$ and piece $[y_0, y_1]$ has measure 0 for player $\phi(1)$. If $y_{n-1} \in X_n$, then piece $[y_{n-1}, y_n]$ has measure at most $1/(n+1)$ for player $\phi(n)$. If $y_{i-1} \in X_j$ and $y_i \in X_j$ for some $2 \leq i \leq n-1$ and $1 \leq j \leq n-1$, then player $\phi(i)$ receives the piece $[y_{i-1}, y_i]$ of measure at most $1/(n+1)$. This leaves the claimed situation as the only possibility.

(b) Player $\phi(i)$ receives the cake interval $[y_{i-1}, y_i]$. By the statement in (a), this interval can not cover player $\phi(i)$'s measure-peaks around j/n-points with $j < i-1$ or with $j > i$. The two peaks around the $(i-1)/n$-point of player $\phi(i)$ yield only a measure of $1/(n+1)$; thus the interval cannot avoid the i/n-point. A symmetric argument shows that the interval cannot avoid the $(i-1)/n$-point of player $\phi(i)$. ■

Lemma 10 *For any permutation $\sigma \neq id$ of the numbers $1 \ldots n$, there exists some $1 \leq i \leq n$ with $\sigma(i+1) \leq i \leq \sigma(i)$.*

Proof. Take the minimum i with $\sigma(i+1) \leq i$. ■

Finally, we claim that $\phi = \pi^{-1}$. Suppose otherwise. Then $\pi \circ \phi \neq \text{id}$ and by Lemma 10 there exists an i such that

$$\pi(\phi(i+1)) \ \leq \ i \ \leq \ \pi(\phi(i)).$$

Let $p := \phi(i+1)$ and $q := \phi(i)$, let z_p denote the i/n-point of player p, and let z_q denote the i/n-point of player q. Lemma 8 yields $z_p < z_q$. According to Lemma 9(b), point z_p must be contained in $[y_i, y_{i+1}]$ and point z_q must be contained in $[y_{i-1}, y_i]$. But this implies $z_p \geq y_i \geq z_q$ and blatantly contradicts $z_p < z_q$.

This contradiction shows that the assignment permutation ρ of protocol $\mathcal{P}^*$ must be equal to the inverse permutation of π. Hence, for each permutation π the primitive protocol must reach a different leaf in the underlying decision tree. After an evaluation query $\mathrm{Eval}(p; x)$, where x is a result of $\mathrm{Cut}(p'; i/n)$, for $p \neq p'$ and $1 \leq i < n$, the protocol is returned one of only two possible answers, namely $i/(n+1)$ or $(i+1)/(n+1)$, indicating if $\mathrm{Cut}(p; i/n)$ is before or after x in X_i (if $p = p'$ or $i \in \{0, n\}$, the answer is unique and trivial). After every query $\mathrm{Cut}(p; i/n)$, the primitive protocol is returned one point of X_i: namely the first unused point if $\pi(p) < i$, the last unused point if $\pi(p) > i$, or the ith point if $\pi(p) = i$. Since the values in X_i are known in advance, the whole protocol can be represented by a tree with a binary node for each possible evaluation query and a ternary node for each possible cut. The depth of a leaf in the tree is the number of cuts and evaluation queries performed for an instance corresponding to a given permutation. Since there are $n!$ permutations, the maximal depth of a leaf corresponding to some permutation must be at least $\log_3(n!) = \Omega(n \log n)$. This completes the proof of Theorem 5.

5 Discussion

One contribution of this paper is a discussion of various models and assumptions for cake cutting (that appeared in the literature in some concise and implicit form) and a definition of a restricted model that covers the best protocols known. This discussion as well as both our results show that evaluation queries play an important role, perhaps underestimated in previous literature.

The positive result seems to be the strongest possible result one can prove for approximately fair protocols as far as the number of cuts is concerned. However, the number of evaluation queries is large, which suggests the following problem.

Problem 1 *Design a deterministic $(1-\varepsilon)/n$-fair cake division protocol that uses $O(n)$ cuts and as few evaluation queries as possible. In particular, is it possible to use $O(n \log n)$ evaluation queries?*

The lower bound of $\Omega(n \log n)$ on the number of cuts and evaluation queries needed for simple fair division with n players is proved in a restricted model. The model clearly has its weak points (see, again, the discussion in Section 2), and it would be interesting to provide similar bounds in less restricted models. In particular, we suggest the two open problems, related to the two restrictions in our model.

Problem 2 *How many cuts are needed if no evaluation queries are allowed (but any player can be assigned several intervals)?*

Our lower bound argument seems to break down even for 'slight' relaxations of the assumption about a single interval: On the instances from our lower bound, one can easily in $O(n)$ cuts assign to each player two of the intervals of size ε that support his measure and this is clearly sufficient. And we do not even know how to make the lower bound work for the case where the cake is a circle, that is, for the cake that results from identifying the points 0 and 1 in the unit interval or equivalently when a single player can receive a share of two intervals, one containing 0 and one containing 1. (Anyway, the circle is considered a non-standard cake and is not treated anywhere in the classical cake cutting literature [1, 8].) Thus the restriction to a single subinterval share for each player seems very significant in our lower bound technique. On the other hand, all the protocols known to us obey this restriction.

Problem 3 *How many cuts are needed if any player is required to receive a single subinterval (but evaluation queries are allowed and free)?*

With evaluation queries, our lower bound breaks, since the decision tree is no longer ternary. After performing a cut, we may learn that $\pi(p) < i$ or $\pi(p) > i$, in which case we gain no additional information. However, once we find i such that $\pi(p) = i$, the protocol finds out all values of p' satisfying $\pi(p') < i$ and we can recurse on the two subinstances. We can use this to give a protocol that uses only $O(n \log \log n)$ cuts (and free evaluation queries) and works on the instances from our lower bound.

It would be very desirable to prove a lower bound for a model including free evaluation queries, or perhaps find some trade-off between cuts and evaluation queries. The perfectly fair protocols typically use only limited evaluations like *"Is your measure of cake piece Z less, greater, or equal to the threshold τ?"* or *"Is your measure of cake piece Z_1 less, greater, or equal to your measure of cake piece Z_2?"*. Perhaps handling these at first would be more accessible. We hope that this problem could be attacked by a similar lower bound technique using the decision trees in connection with a combinatorially richer set of instances.

Another interesting question concerns the randomized protocols. The randomized protocol of Even & Paz [3] uses an expected number of $O(n)$ cuts and $\Theta(n \log n)$ evaluation queries. Can the number of evaluation queries be decreased? Or can our lower bound be extended to randomized protocols?

Finally, let us remark that our model seems to be incomparable with that of Magdon-Ismail, Busch & Krishnamoorthy [5]. The set of instances for which they prove a lower bound of $\Omega(n \log n)$ on the number of comparisons can be easily solved with $O(n)$ cuts with no evaluation queries even in our restricted model. On the other hand, they prove a lower bound for protocols that have no restriction similar to our requirement of assigning a single subinterval to each player. The common feature of both models seems to be exactly the lack of ability to incorporate the free evaluation queries; note that using an evaluation query generates at least one comparison.

Acknowledgment. We thank anonymous referees for providing several comments that helped us to improve the paper.

References

[1] S.J. Brams and A.D. Taylor (1996). *Fair Division – From cake cutting to dispute resolution.* Cambridge University Press, Cambridge.

[2] L. Carroll (1865). *Alice's Adventures in Wonderland.* [The quoted poem appears only in some later editions of the book.]

[3] S. Even and A. Paz (1984). A note on cake cutting. *Discrete Applied Mathematics 7*, 285–296.

[4] S.O. Krumke, M. Lipmann, W. de Paepe, D. Poensgen, J. Rambau, L. Stougie, and G.J. Woeginger (2002). How to cut a cake almost fairly. *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2002)*, 263–264.

[5] M. Magdon-Ismail, C. Busch, and M.S. Krishnamoorthy (2003). Cake cutting is not a piece of cake. *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'2003)*, LNCS 2607, Springer Verlag, 596–607.

[6] J.M. Robertson and W.A. Webb (1991). Minimal number of cuts for fair division. *Ars Combinatoria 31*, 191–197.

[7] J.M. Robertson and W.A. Webb (1995). Approximating fair division with a limited number of cuts. *Journal of Combinatorial Theory, Series A 72*, 340–344.

[8] J.M. Robertson and W.A. Webb (1998). *Cake-cutting algorithms: Be fair if you can.* A.K. Peters Ltd.

[9] J. Sgall and G. J. Woeginger (2003). A lower bound for cake cutting. LNCS 2461, Springer Verlag, 896–901. *Proc. of the 11th Ann. European Symp. on Algorithms (ESA), Lecture Notes in Comput. Sci. 2832*, Springer, 459–469.

[10] H. Steinhaus (1948). The problem of fair division. *Econometrica 16*, 101–104.

[11] W. Stromquist (1980). How to cut a cake fairly. *American Mathematical Monthly 87*, 640–644.

[12] W.A. Webb (1997). How to cut a cake fairly using a minimal number of cuts. *Discrete Applied Mathematics 74*, 183–190.

[13] G.J. Woeginger (2002). An approximation scheme for cake division with a linear number of cuts. *Proc. of the 10th Ann. European Symp. on Algorithms (ESA), Lecture Notes in Comput. Sci. 2461*, Springer, 896–901.