Balanced Allocations of Cake

Jeff Edmonds∗ Kirk Pruhs†

Abstract

We give a randomized algorithm for the well known caking cutting problem that achieves approximate fairness, and has complexity $O(n)$. The heart of this result involves extending the standard offline multiple-choice balls and bins analysis to the case where the underlying resources/bins/machines have different utilities to different players/balls/jobs.

∗York University, Canada. jeff@cs.yorku.ca. Supported in part by NSERC Canada.
†Computer Science Department. University of Pittsburgh. kirk@cs.pitt.edu. Supported in part by NSF grants CNS-0325353, CCF-0448196, CCF-0514058 and IIS-0534531.
1 Introduction

The protocol’s goal in the well known cake cutting problem is to fairly apportion some resources among \(n\) players. Here we consider a continuous resource, modeled, without great loss of generality, by the unit interval. We assume that each player \(p\) has an initially unknown value function \(V_p\) that specifies how player \(p\) values each subinterval of the unit interval. A portion is a union of disjoint subintervals, and the value function is additive, so that the value of a portion is the sum of the values of the underlying subintervals. A player believes that a portion is \(c\)-fair if that portion has value at least \(\frac{1}{cn}\) of the total value of cake according to his value function. In the standard model, the protocol is allowed to make two types of queries to the players. In an evaluation query, the protocol asks a player how much he values a particular subinterval of the cake. In a cut query, the protocol asks the player to identify the shortest subinterval with a fixed value and a fixed left endpoint. We are interested in the query complexity of a protocol, which is the worst-case number of queries required to achieve a fair allocation for each player that follows the protocol.

The cake cutting problem originated in 1940’s Polish mathematics community. Since then the problem has blossomed and been widely popularized. The motivation for using cake as a resource is the well known phenomenon that some people prefer frosting, while others do not. Cake cutting, and related fair allocation problems, are of wide interest in both social sciences and mathematical sciences. Sgall and Woeginger [11] provide a nice brief overview. There are several books written on fair allocation problems, such as cake cutting, that give more extensive overviews, see for example [3, 10]. Some quick Googling reveals that cake cutting algorithms, and their analysis, are commonly covered by computer scientists in their algorithms and discrete mathematics courses.

A deterministic 1-fair protocol with complexity \(\Theta(n^2)\) was described in 1948 by Steinhaus in [12]. In 1984, Evan and Paz [5] gave a deterministic divide and conquer 1-fair protocol that has complexity \(\Theta(n \log n)\). Recently, there has been several lower bound results for cake cutting. In particular, we showed that the Even-Paz algorithm is optimal for deterministic 1-fair protocols [4]. That is, every deterministic 1-fair protocol for cake cutting has complexity \(\Omega(n \log n)\). This lower bound also applies to deterministic protocols that need only only guarantee \(O(1)\)-fairness.

A natural open question is then whether there exists a randomized protocol with linear query complexity. Some lower bound results for randomized algorithms are known. Sgall and Woeginger [11] showed that every randomized 1-fair protocol has complexity \(\Omega(n \log n)\) if every portion is restricted to be a contiguous subinterval of the cake. We showed that every randomized \(O(1)\)-fair protocol has complexity \(\Omega(n \log n)\) if there is a small relative error in the response to the queries [4].

In this paper we give a randomized protocol with \(O(n)\) query complexity. Our protocol requires exact answers to the queries, guarantees only \(O(1)\)-fairness, and does not in general assign a contiguous subinterval to each player. That is, we show that linear complexity is obtainable in the variant that is most in the protocol’s favor. Note that by the results in [4], there is no deterministic protocol that guarantees \(O(1)\)-fairness. So this result separates deterministic and randomized query complexity for approximate fairness.

Our protocol also requires that all of the players are honest. Honesty is not a real issue in deterministic protocols, but is a significant issue in most conceivable randomized protocols. For example, a randomized protocol might ask a player to generate a subinterval/piece according to a particular probability distribution. To handle a dishonest player, a protocol would seem to need to be able to determine if the player actually generated a piece according to this distribution. This seems like a daunting task for the protocol.

Additionally, we show that \(O(n)\)-complexity is still achievable even if there is a small relative error in the response to the queries, as long as the error that results from a cut query is independent of value in the query. We call this a weak adversary.

The heart of our cake cutting algorithm is the following Balanced Allocation Lemma in the cake model that generalizes the standard multiple-choice balls and bins model [8].
Table 1: Summary of known results. An asterisk means that the result holds for both choices.

**Lemma 1 (Balanced Allocation).** Let \( \alpha \) be some sufficiently large constant. Each of \( n \) players has a partition of the unit interval \([0, 1]\), or cake, into \( \alpha n \) disjoint candidate subintervals/pieces. Each player independent picks \( d' = 2d = 4 \) of his pieces uniformly at random, with replacement. Then there is an efficient method that, with probability \( \Omega(1) \), picks one of the \( d' \) pieces for each player, so that every point on the unit interval is covered by \( O(1) \) pieces.

In the analogous multiple-choice balls and bins model, each player independently selects \( d' \) of \( \alpha n \) discrete bins uniformly at random. This balls and bins model is equivalent to the special case of the cake model in which each player has the same collection of \( \alpha n \) candidate pieces. It is a folklore result that in the balls and bins model, the maximum load is \( \Theta(\log \log n) \) if \( d' = 1 \); And if \( d' > 1 \), then with one can with probability \( \Omega(1) \) pick one of the \( d' \) pieces for each player in such a way that each bin only has 1 ball, and one can with high probability pick one of the \( d' \) pieces for each player in such a way that each bin only at most 2 balls. One can even get maximum load \( O(\log \log n) \) if the assignment has to be made online player by player [2].

We now briefly discuss how our Balanced Allocation Lemma can be used to solve the cake cutting problem (See Appendix Section A for more details). The \( i \)th candidate piece is the \( i \)th subinterval of value \( 1/\alpha n \), which can be found by two cut queries. After the application of the Balanced Allocation Lemma, any standard fair allocation algorithm can be used to divide any portion of the cake desired by more than one player.

1.1 Related Results

The first step towards obtaining an \( \Omega(n \log n) \) lower bound on the complexity of cake cutting was taken by Magdon-Ismail, Busch, and Krishnamoothy [7], who were able to show that any protocol must make \( \Omega(n \log n) \) comparisons to compute the assignment. So this result does not address query complexity. Approximately fair protocols were introduced by Robertson and Webb [9]. Traditionally, much of the research has focused on minimizing the number of cuts, presumably out of concern that too many cuts would lead to crumbling of a literal cake. There is a deterministic protocol that achieves \( O(1) \)-fairness with \( \Theta(n) \) cuts and \( \Theta(n^2) \) evaluations [9, 6, 13]. There are several other objectives studied in the cake cutting setting, most notably, max-min fairness, and envy-free fairness.

The literature on balanced allocations is also rather large. A nice survey is given in [8].
2 Intuition

In this section we try to give some intuition and a road map for the proof of our Balanced Allocation Lemma. We start with an example instance, see Figure 1 that demonstrates several interesting features of the cake model and our analysis. Each of the rows consists of the $\alpha n$ subintervals of the $n$ players. The $n/2$ $A$ players have $\alpha n$ candidate pieces of identical length. Then for $i \in [1, \sqrt{\frac{n}{2}}]$, there is a group of $\sqrt{\frac{n}{2}}$ $B_i$ players. Half of a $B_i$’s candidate pieces overlap with the $2i^{th}$ piece of the $A$ players, and half with the $2i+1$ st piece of the $A$ players.

![Figure 1: An example in which player’s intervals overlap in more complex ways.](image)

One immediate observation is that maximum load equal to 1 result from the standard multiple-choice balls and bins model will not carry over to the cake model. To see this, note that with high probability, one of the $A$ players chooses all of his $d'$ pieces from his first $2\sqrt{\frac{n}{2}}$ candidate pieces. Call this player $A'$. Also with high probability, for each $d'$ pieces of $A'$, there is a $B_i$ player that has all of $d'$ pieces overlapping with it. This explains the need to relax the maximum load bound from 1 to $O(1)$.

The Implication Graph: To gain intuition, let us assume for the moment that $d' = 2$. Let $c_{(p,i)}$ denote the $i^{th}$ in $[1, \alpha n]$ candidate piece for player $p$. Let $a_{(p,0)}$ and $a_{(p,1)}$ be the two semifinal pieces selected for player $p$. We now define what we call the implication graph. The vertices of the implication graph are the $2n$ pieces $a_{(p,r)}$, $1 \leq p \leq n$ and $0 \leq r \leq 1$. If piece $a_{(p,r)}$ intersects piece $a_{(q,s)}$, then there is an directed edge from piece $a_{(p,r)}$ to piece $a_{(q,1-s)}$ and similarly from $a_{(q,s)}$ to $a_{(p,1-r)}$. The intuition is that if player $p$ gets $a_{(p,r)}$ as his final piece, then player $q$ must get piece $a_{(q,1-s)}$ if $p$ and $q$’s pieces are not going to overlap. Similarly if $q$ gets $a_{(q,s)}$, then $p$ must get $a_{(p,1-r)}$. As an example, Figure 2 gives a subset of the semifinal pieces selected from the candidate pieces in Figure 1. The directed edges arising from this example are given.

![Figure 2: Two excerpts from an implication graph.](image)

Pair Path: We define a pair path in the implication graph to be a directed path between the two pieces for one player, i.e. from some $a_{(p,r)}$ to $a_{(p,1-r)}$. In Figure 2.a, there are two such paths of length four from the $A$ player’s left semifinal piece to his right and in Figure 2.b two paths of length two. We will show that if the implication graph $G$ does not contain any such pair paths, then the following algorithm selects a final piece for each player in such a way that these final pieces are disjoint. (See Section 3.1.)

Final Piece Selection Algorithm Description: We repeatedly pick an arbitrary player $p$ that has not selected a final piece. We pick the piece $a_{(p,0)}$ as the final piece for $p$. Further, we pick as final pieces all those pieces in $G$ that are reachable from $a_{(p,0)}$ in $G$.

Independent Edges: To gain intuition, we now sketch a proof that the implication graph does not contain a pair path for the balls and bins model (each player’s collection of $\alpha n$ candidate pieces are identical). Note that in the balls and bins model, every pair path has to be of length at least 3. Consider a possible pair path $a_{(p_0,r_0)}, a_{(p_1,r_1)}, \cdots, a_{(p_k-1,r_{k-1})}, a_{(p_0,1-r_0)}$ with $k$ edges in the implication graph. The probability that a particular pair of nodes $\langle a_{(p_0,r_0)}, a_{(p_1,r_1)} \rangle$ has an edge between them, i.e. the probability that the candidate
piece chosen to be $a_{(p_0, r_0)}$ intersects with that chosen to be $a_{(p_1, 1-r_1)}$, is $\frac{1}{\alpha n}$. The presence or absence of these $k$ edges in the implication graph are statistically independent. Thus the probability that this particular pair path appears in the implication graph is at most $\left(\frac{1}{\alpha n}\right)^k$. Since there are at most $\binom{2n}{k} k!$ possible pair paths with $k$ edges, the probability that there is pair path is at most $\sum_{k=3}^{n} \binom{2n}{k} k! \frac{1}{(\alpha n)^k}$. If $\alpha$ is sufficiently large, then this probability is say at most $1/2$.

We now return to the general cake model. One difficulty is that the edges in the implication graph are no longer independent. To see this, recall Figure 1. The probability that any two semifinal pieces overlap is still $O\left(\frac{1}{\alpha n}\right)$. However, if one of an $A$ player’s semifinal pieces overlaps with one $B_i$ player’s semifinal piece, then we know that this $A$ player must have selected either his $2^{i}\text{th}$ or $2i+1^{\text{st}}$ candidate piece and hence it very likely to also overlap with another $B_i$ player’s semifinal piece.

**Pair Paths of Length $\geq$ Three and Vees:** Such dependencies can occur when there is what we call a vee among the candidate pieces. We define a vee to consist of a triple of pieces, one center piece and two base pieces, with the property that the center piece intersects both of the base two pieces. For example, see the three left most pieces in Figure 2.a.

Note that in the balls and bins model, the expected number of vee’s among the semifinal pieces is $O\left(\left(\frac{2^n}{3}\right) \frac{1}{(\alpha n)^2}\right) = O(n)$. And in the cake model, we will show that if the expected number of vee’s among the semifinal pieces is $O(n)$, then with probability $\Omega(1)$ there will be no pair path with three of more edges in the implication graph of the semifinal pieces. (See Section 3.3). Unfortunately, in the example in Figure 1, it is the case that, with high probability, the number of vee’s among the semifinal pieces is $\Omega\left(\sqrt{n} \cdot \sqrt{n/2}\right) = \Omega(n^{3/2})$. The consequence of this is that, with high probability, there will be pair paths like those in Figure 2.a. One can also construct instances where the number of vee’s is $\Omega(n^2)$ with probability $\Omega(1)$.

Getting the expected number of vee’s in the semifinal pieces down to $O(n)$ necessitates that $d' \geq 4$. Let us now explain how we accomplish this. The selection of final pieces will occur in three instead of two phases. First, each player independently at randomly chooses $d' = 2d$ quarterfinal pieces. These are partitioned into two brackets $A_{(p,0)}$ and $A_{(p,1)}$ containing $d$ pieces each. From each such bracket, we choose one interval, denoted $a_{(p,r)}$ to be a semifinal piece. The semifinal piece is chosen to be the one that intersects the smallest number of other candidate pieces, $c_{(q,j)}$. Note that this processes is independent for the different players $p$ and for each bracket. We will show then that the expected number of vee’s in the resulting $2n$ semifinal pieces is $O(n)$ (see Section 3.2). We show that as a consequence of this, with probability $\Omega(1)$, the implication graph of the semifinal pieces does not contain a pair path of length 3 or longer.

**Pair Paths of Length Two and Same-Player-Vees:** Another difficulty is that the implication graph of the semifinal pieces may, with high probability, have pair paths of length two. See Figure 2.b. A pair path of length two occurs if and only if there is such a player-vee involving two players in the same partition. Therefore, with probability $\Omega(1)$, we can partition the players into 2 partitions in such a way there is no same-player-vee involving two players in the same partition.

**Summary of Balanced Allocation Algorithm:** We summarize our Balanced Allocation Algorithm.

- Independently, for each player $p \in [1, n]$ and each $r \in [0, 1]$, randomly choose $d'$ of the candidate pieces $c_{(p,i)}$ to be in the quarterfinal bracket $A_{(p,r)}$. 

• In each quarterfinal bracket $A_{(p,r)}$, pick as the semifinal piece $a_{(p,r)}$, the piece that intersects the fewest other candidate pieces $c_{(q,j)}$. If we are unlucky and the Implication Graph contains a pair path of length greater than 3, then start over. See Sections 3.2 and 3.3.

• Construct and vertex color the same-player-vee graph using the greedy coloring algorithm using at most $w = 2$ colors. See Section 3.4. Let $S_k$ be the subgraph of the implication graph containing only those players colored $k$. This ensures that Implication Graph restricted to $S_k$ contains no pair paths of length 2.

• For each $S_k$, pick the final piece for each player involved in $S_k$ by applying the Final Piece Selection Algorithm to $S_k$. See Section 3.1. Because the Implication Graph on $S_k$ contains no pair paths of any length, this algorithm ensures that these final pieces for each player are disjoint, i.e. for any point in the cake, the final piece of at most one player from $S_k$ covers this point. Conclude that for any point in the cake, the final piece of at most $w = 2$ players cover this point. The total probability of success is computed in Section 3.5.

In section 3.6 we extend this Balanced Allocation Algorithm to the case of approximate queries against a weak adversary.

3 The Proofs

In this section we prove the various claims that we made in the previous section. Each subsection can essentially be read independently of the others. Due to space limitations, some proofs are moved to the appendix, and some of the easier proofs are omitted.

3.1 Final Piece Selection Algorithm

We show some structural properties of the implication graph imply the correctness of the Final Piece Selection Algorithm.

Lemma 2. If there is a path in $G$ from $a_{(p,r)}$ to $a_{(q,s)}$ then there must be a path from $a_{(q,1-s)}$ to $a_{(p,1-r)}$ in $G$.

Lemma 3. If both the pieces $a_{(q,0)}$ and $a_{(q,1)}$ are reachable from a piece $a_{(p,r)}$ in the implication graph $G$, then $G$ has a pair path.

Lemma 4. If an implication graph $G$ of the semifinal pieces does not contain a pair path, then the Final Piece Selection Algorithm selects a final piece for each player and these final pieces are disjoint.

Proof. Consider an iteration that starts by assigning $a_{(p,0)}$ to player $p$. This iteration will force the assignment of at most one piece to any one player because by Lemma 3 there can not be a player $q$ such that both $a_{(q,0)}$ and $a_{(q,1)}$ are reachable from $a_{(p,0)}$. Similarly, if this same iteration forces player $q$ to be assigned say to $a_{(q,0)}$, then we need to prove that he has not already been assigned $a_{(q,1)}$ during an earlier iteration. If assigning $a_{(p,0)}$ forces $a_{(q,0)}$, then there is a path from the one to the other. Hence, by Lemma 2, there is a path from $a_{(q,1)}$ to $a_{(p,1)}$. Hence, if $a_{(q,1)}$ had been previously assigned, then player $p$ would have been forced to $a_{(p,1)}$ and in this case $p$ would not be involved in this current iteration. The disjointness of the final pieces follows from the definition of the implication graph.

3.2 The Number of Vees

In this subsection we show that the number of vees is $O(n)$ with probability $\Omega(1)$. Recall that a vee consists of a triple of semifinal pieces, one center piece $a_{(p,r)}$ and two base pieces $a_{(q,s)}$ and $a_{(q',s')}$, with the property that the center piece intersects both of the base two pieces.
Lemma 6. The probability that semifinal piece \( a_{(p, r)} \) overlaps with semifinal piece \( a_{(q, s)} \) is at most \( \frac{2d^2}{\alpha n} \).

Proof. Consider a particular player \( p \). Again let \( \ell_{(p, i)} \) denote the total number of candidate pieces overlapping the \( i^{th} \) candidate piece \( c_{(p, i)} \) of the player \( p \). Without loss of generality, let us renumber \( p \)'s candidate pieces in non-increasing order by \( \ell_{(p, i)} \), that is, \( \ell_{(p, i)} \geq \ell_{(p, i+1)} \).

For \( p \in [n], i \in [\alpha n], \) and \( r \in [0, 1], \) let \( R_{(p, i, r)} \) be the event that the candidate \( c_{(p, i)} \) is selected to be the semifinal piece \( a_{(p, r)} \). To understand this, let us review how this is chosen. First, player \( p \) randomly chooses \( d \) candidate pieces to be in his quarterfinal brackets \( A_{(p, r)} \). Then the semifinal piece \( a_{(p, r)} \) is chosen to be the one with the smallest \( \ell_{(p, i)} \) value or, by our ordering, the one with the largest index. Hence, the probability of \( R_{(p, i, r)} \) is the probability that \( d \) indexes are randomly selected from \( \alpha n \) indexes and the largest selected index is \( i \). This gives \( \text{Prob}[R_{(p, i, r)}] = d \cdot \left( \frac{1}{\alpha n} \right)^d \cdot \left( \frac{i}{\alpha n} \right) \).

Let \( x_{(p, r)} \) be the number of vee’s with \( a_{(p, r)} \) as the center. There are \( \binom{\ell_{(p, i)}}{2} \) pairs of candidate pieces that might be the two base pieces \( a_{(q, s)} \) and \( a_{(q', s')} \) with the center piece \( a_{(p, r)} = c_{(p, i)} \). The probability that both of this pair are semifinal pieces is at most \( \left( \frac{2d}{\alpha n} \right)^2 \). Hence, \( E[x_{(p, r)} \mid R_{(p, i, r)}] \) is at most \( \left( \frac{\ell_{(p, i)}}{2} \right) \left( \frac{2d}{\alpha n} \right)^2 \leq 2d^2 \left( \frac{d}{\alpha n} \right)^2 .

\[
E[x_{(p, r)}] = \sum_{i=1}^{\alpha n} \text{Prob}[R_{(p, i, r)}] \cdot E[x_{(p, r)} \mid R_{(p, i, r)}] \leq \sum_{i=1}^{\alpha n} \left( \frac{d}{\alpha n} \right)^d \left( \frac{i}{\alpha n} \right)^{d-1} \cdot 2d^2 \left( \frac{d}{\alpha n} \right)^2 \leq 2\alpha n^2.
\]

Lemma 5 bounds that \( \sum_{i=1}^{\alpha n} \ell_{(p, i)} \leq 2\alpha n^2 = M \). The next lemma then bounds \( \sum_{i=1}^{m} i^{d-1} \ell_{(p, i)}^2 \leq m^{d-2}M^2 \).

\[
E[x_{(p, r)}] \leq \left( \frac{2d^3}{(\alpha n)^{d+2}} \right) \cdot (\alpha n)^{d-2} \cdot (2\alpha n^2)^2 \leq \frac{8d^3}{\alpha^2} .
\]

By linearity of expectation, the expected number of vee’s over all is \( \sum_{p=1}^{n} \sum_{r=0}^{1} E[x_{(p, r)}] \leq 2n \cdot \frac{8d^3}{\alpha^2} . \]

Lemma 8. If \( d \geq 2, \forall i \in [1, m-1], \ell_i \geq \ell_{i+1} \geq 0, \) and \( \sum_{i=1}^{m} \ell_i = M, \) then \( \sum_{i=1}^{m} i^{d-1} \ell_i^2 \leq m^{d-2}M^2 \).

Proof. Let \( \ell_{m+1} = 0, \) and \( s_i = \ell_i - \ell_{i+1} \) for \( 1 \leq i \leq m \). Note that our constraint gives that \( s_i \geq 0 \). Further more, \( \ell_i = \sum_{j=i}^{m} s_j \) and \( M = \sum_{i=1}^{m} \ell_i = \sum_{i=1}^{m} is_i \). Then let \( t_i = is_i \) so that \( M = \sum_{i=1}^{m} t_i \). Now using basic algebra we conclude that

\[
\sum_{i=1}^{m} i^{d-1} \ell_i^2 = \sum_{i=1}^{m} i^{d-1} \left( \sum_{j=i}^{m} s_j \right)^2 = \sum_{i=1}^{m} i^{d-1} \sum_{j=i}^{m} \sum_{k=i}^{m} s_j s_k = \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{i=1}^{\min(j, k)} i^{d-1} \leq \sum_{j=1}^{m} \sum_{k=1}^{m} t_j t_k \min(j, k)^d \leq m^{d-2} \sum_{j=1}^{m} \sum_{k=1}^{m} t_j t_k = m^{d-2} \left( \sum_{j=1}^{m} t_j \right)^2 = m^{d-2}M^2 .
\]
3.3 The Existence of Pair Paths

In this subsection, we show that with probability $\Omega(1)$, the implication graph doesn’t contain a pair path of length three or more. Recall that if the semifinal pieces $a_{(p,r)}$ and $a_{(q,s)}$ intersect, then there is an directed edge in the implication graph $G$ from $a_{(p,r)}$ to $a_{(q,1-s)}$ and from $a_{(q,s)}$ to $a_{(p,1-r)}$ and that a pair path is a directed path between the two semifinal pieces for the same player, i.e. from some $a_{(p,r)}$ to $a_{(p,1-r)}$. The next lemma is best understood by studying Figure 3.

**Lemma 9.** Consider a simple pair path $P = \langle a_{(p_0,r_0)}, a_{(p_1,r_1)}, \ldots a_{(p_k-1,r_{k-1})}, a_{(p_0,1-r_0)} \rangle$ of length $k \geq 3$. Let $V$ be the vee with center $a_{(p_0,r_0)}$ and bases $a_{(p_1,1-r_1)}$ and $a_{(p_k-1,r_{k-1})}$. For $i \in [1, k-2]$, let $I_i \in G$ be the event that semifinal pieces $a_{(p_i,r_i)}$ and $a_{(p_{i+1},1-r_{i+1})}$ intersect. Then

$$\text{Prob}[P \in G] \leq \text{Prob}[V \in G] \cdot \Pi_{i=1}^{k-2} \text{Prob}[I_i \in G]$$

**Proof.** The edges from $a_{(p_0,r_0)}$ to $a_{(p_1,r_1)}$ and from $a_{(p_k-1,r_{k-1})}$ to $a_{(p_0,1-r_0)}$ mean that $a_{(p_0,r_0)}$ intersect with both $a_{(p_1,1-r_1)}$ and $a_{(p_k-1,r_{k-1})}$. Hence, the vee $V$ occurs. The edge from $a_{(p_i,r_i)}$ to $a_{(p_{i+1},r_{i+1})}$ means that $a_{(p_i,r_i)}$ and $a_{(p_{i+1},1-r_{i+1})}$ intersect, i.e. $I_i$. It follows that $\text{Prob}[P \in G] \leq \text{Prob}[V \& \text{each } I_i \in G]$. What remains is to prove that the events $V$ and each $I_i$ are independent. Whether a semifinal piece of players $p$ and $q$ intersect is independent of whether a semifinal piece of different players $p'$ and $q'$ intersect because these event have nothing to do with each other. This remains true when the players $p$ and $p'$ are the same, but the we are talking about different semifinal pieces of this player, namely event $I_i$ and $I_{i+1}$ are independent. This is because the selection of the quarterfinal pieces for the bracket $A_{(p,0)}$ and the selection of $p'$s semifinal piece $a_{(p,0)}$ within this bracket is independent of this process for his other semifinal piece $a_{(p,1)}$. □

**Lemma 10.** The probability that the implication graph $G$ contains a pair path of length at least three is at most $\frac{32d^5}{\alpha^2 (\alpha - 4d^2)}$.

**Proof.** Let $V$ be the set of all 3-tuples representing all possible vee’s in $G$ and for $V \in V$ let $\mathcal{P}_k(V)$ be the set of all possible pair paths of length $k$ that include the vee $V$. The probability that $G$ contains a pair path of length at least three is at most

$$\sum_{k=3}^{n} \sum_{V \in V} \sum_{P \in \mathcal{P}_k(V)} \text{Prob}[P \in G]$$

$$\leq \sum_{k=3}^{n} \sum_{V \in V} \sum_{P \in \mathcal{P}_k(V)} \text{Prob}[V \in G] \cdot \Pi_{i=1}^{k-2} \text{Prob}[I_i \in G]$$

$$\leq \sum_{k=3}^{n} \sum_{V \in V} \text{Prob}[V \in G] \sum_{P \in \mathcal{P}_k(V)} \left( \frac{2d^2}{\alpha n} \right)^{k-2}$$

(1)

(2)

(3)

(4)
\[ \leq \sum_{k=3}^{n} \sum_{V \in \mathcal{V}} \operatorname{Prob}[V \in G] \left( \frac{2n}{k-3} \right) (k-3)! \left( \frac{2d^2}{\alpha n} \right)^{k-2} \]  
\[ \leq \sum_{k=3}^{n} (2n)^{k-3} \left( \frac{2d^2}{\alpha n} \right)^{k-2} \sum_{V \in \mathcal{V}} \operatorname{Prob}[V \in G] \]  
\[ \leq \sum_{k=3}^{n} (2n)^{k-3} \left( \frac{2d^2}{\alpha n} \right)^{k-2} \left( \frac{16d^3}{\alpha^2 n} \right) \]  
\[ \leq \frac{8d^3}{\alpha^2} \sum_{k=3}^{n} \left( \frac{4d^2}{\alpha} \right)^{k-2} \leq \frac{8d^3}{\alpha^2} \left( \frac{4d^2}{\alpha} \right) \left( \frac{1}{1 - 4d^2/\alpha} \right) = \frac{32d^5}{\alpha^2(\alpha - 4d^2)} \]  

The inequality in line 2 follows from Lemma 9 and line 3 from Lemma 6. The inequality in line 5 holds since there are \( k - 3 \) pieces in \( P \) that are not part of the vee \( V \). The inequality in line 7 follows from Lemma 7.

### 3.4 Coloring Same-Player-Vee Graphs

In this subsection we show that with probability \( \Omega(1) \), we can color the same-player-vee graph with 2 colors since this graph will have no paths of length \( w = 2 \).

**Lemma 11.** The probability that the same-player-vee graph is not \( w = 2 \) colorable is at most \( \frac{16d^3}{\alpha^2} + \frac{8d^2}{\alpha} \).

Recall that we put the directed edge \( \langle p, q \rangle \) in the same-player-vee graph if one of player \( p \)'s two semifinal pieces, namely \( a_{(p,0)} \) or \( a_{(p,1)} \), overlap with both of player \( q \)'s two semifinal pieces, namely \( a_{(q,0)} \) and \( a_{(q,1)} \). Hence, a path of length 3 consists of semi-final pieces \( a_{(p_1,r_1)}, a_{(p_2,r_2)}, a_{(p_2,1-r_2)}, a_{(p_3,0)}, \) and \( a_{(p_3,1)} \) for three players \( p_1, p_2, \) and \( p_3 \), where both \( a_{(p_2,r_2)} \) and \( a_{(p_2,1-r_2)} \) overlap with \( a_{(p_1,r_1)} \), and both \( a_{(p_3,0)} \) and \( a_{(p_3,1)} \) overlap with \( a_{(p_2,r_2)} \). We will consider the probability of such paths starting backwards.

**Lemma 12.** Suppose we are considering a set of \( \hat{\ell} \) candidate pieces for the semi-final pieces \( a_{(p_3,0)} \) and \( a_{(p_3,1)} \). The probability that some player gets both of his semi final pieces from this set is at most \( \min\left( \frac{\hat{\ell} d}{\alpha n}, 1 \right) \).

Consider some candidate piece \( c_{(p_3,j)} \) that potentially might be \( a_{(p_3,j_1)} \). Let \( \ell_{(p_3,j)} \) denote the number of other candidate pieces of overlapping it. Consider some player \( p_2 \). Let \( c_{(p_2,j_1)}, c_{(p_2,j_1+1)}, \ldots, c_{(p_2,j_t)} \) be the candidate pieces of player \( p_2 \) that overlap with piece \( c_{(p_3,j)} \). Let \( \ell_{(p_2,j)} \) denote the number of other candidate pieces of overlapping \( c_{(p_2,j)} \). Consider some player \( p_3 \). Define \( \ell_{(p_2,j_3,p_3)} \) to be the number of player \( p_3 \)'s candidate pieces that overlap \( c_{(p_2,j)} \). Note that if \( \ell_{(p_2,j_3,p_3)} = 1 \), then it is impossible to have both of player \( p_3 \)'s semi-final pieces overlap with \( c_{(p_2,j)} \). Hence, we can ignore player \( p_3 \) when considering \( c_{(p_2,j)} \) as being \( a_{(p_2,r_2)} \). Hence, define \( \hat{\ell}_{(p_2,j_3,p_3)} \) to be \( \ell_{(p_2,j_3,p_3)} \) if \( \ell_{(p_2,j_3,p_3)} \geq 2 \) and zero otherwise. Define \( \hat{\ell}_{(p_2,j)} = \sum_q \hat{\ell}_{(p_2,j_3,p_3)} \). Note this is the number of pieces that overlap \( c_{(p_2,j)} \) excluding those pieces whose player only has one piece overlapping \( c_{(p_2,j)} \).

**Lemma 13.** Then \( \sum_{i=j_1+1}^{j_1+1} \hat{\ell}_{(p_2,j)} \leq 2\ell_{(p_3,j)} \).

**Lemma 14.** Consider a candidate piece \( c_{(p_1,i)} \) such that there are \( \ell_{(p_1,i)} \) other candidate pieces overlapping it and some other player \( p_2 \). The probability that there are semi-final pieces \( a_{(p_2,r_2)}, a_{(p_3,0)}, \) and \( a_{(p_3,1)} \) for some player \( p_3 \), where \( a_{(p_2,r_2)} \) overlaps with \( c_{(p_1,i)} \), and both \( a_{(p_3,0)} \) and \( a_{(p_3,1)} \) overlap with \( a_{(p_2,r_2)} \) is at most \( \frac{4d}{\alpha n} \left( \frac{d\ell_{(p_1,i)}}{\alpha n} + 1 \right) \).
Proof. Consider a candidate piece \(c_{(p_2,j)}\) that overlaps with \(c_{(p_1,i)}\). The probability that candidate piece \(c_{(p_2,j)}\) is a semi-final piece for player \(p_2\) is at most \(\frac{2d}{\alpha n}\). By Lemma 12, the probability that there are semi-final pieces \(a_{(p_3,0)}\) and \(a_{(p_3,1)}\) for some player \(p_3\) which both overlap with \(c_{(p_2,j)}\) is at most \(\min(\frac{d\ell_{(p_2,j)}}{\alpha n}, 1)\). It follows that the required probability is at most

\[
\frac{ \sum_{i=j_1}^{j_r} \frac{2d}{\alpha n} \cdot \min \left( \left( \frac{d\ell_{(p_2,j)}}{\alpha n} \right)^2, 1 \right) }{ \sum_{i=j_1}^{j_r-1} \min \left( \left( \frac{d\ell_{(p_2,j)}}{\alpha n} \right)^2, 1 \right) + 1 }.
\]

By Lemma 13, \(\sum_{i=j_{1}+1}^{j_r} \ell_{(p_2,j)} \leq 2\ell_{(p_1,i)}\). Hence, because of the quadratics in the sum, our sum is maximized by having a few \(\ell_{(p_2,j)}\) as big as possible. But because of the min, there is no reason to make a \(\ell_{(p_2,j)}\) bigger than \(\frac{\alpha n}{d}\). Hence, the sum is maximized by setting \(\frac{2d\ell_{(p_1,i)}}{\alpha n}\) of the values \(\ell_{(p_2,j)}\) to \(\frac{\alpha n}{d}\) and the rest to zero. This gives the result

\[
\frac{2d}{\alpha n} \cdot \left[ 1 + \frac{2d\ell_{(p_1,i)}}{\alpha n} \cdot \min(1, 1) \right] + 1.
\]

We will now add the requirement that player \(p_2\)’s other candidate piece \(a_{(p_2,1-r_2)}\) also overlaps with \(c_{(p_1,i)}\) and sum the resulting probability over all possible players \(p_2\).

Lemma 15. Consider a candidate piece \(c_{(p_1,i)}\) such that there are \(\ell_{(p_1,i)}\) other candidate pieces overlapping it. The probability that there are semi-final pieces \(a_{(p_2,r_2)}\), \(a_{(p_2,1-r_2)}\), \(a_{(p_3,0)}\), and \(a_{(p_3,1)}\) for two players \(p_2\) and \(p_3\), where both \(a_{(p_2,r_2)}\) and \(a_{(p_2,1-r_2)}\) overlap with \(c_{(p_1,i)}\), and both \(a_{(p_3,0)}\) and \(a_{(p_3,1)}\) overlap with \(a_{(p_2,r_2)}\) is at most \(\frac{4d^4\ell_{(p_1,i)}^3}{(\alpha n)^3} \cdot \left[ 1 + \frac{\alpha n}{d\ell_{(p_1,i)}} \right]\).

Proof. The probability that a particular candidate piece \(c_{(p_2,j)}\) is player \(p_2\)’s semi-final piece \(a_{(p_2,1-r_2)}\) is at most \(\frac{d}{\alpha n}\). Denote the number of player \(p_2\)’s candidate pieces \(c_{(p_2,j_1)}, c_{(p_2,j_1+1)}, \ldots, c_{(p_2,j_r)}\) that overlap with piece \(c_{(p_1,i)}\) to be \(q_{p_2} = j_r - j_1 + 1\). Because these all overlap with \(c_{(p_1,i)}\), we have that \(\sum_{p_2} q_{p_2} = \ell_{(p_1,i)}\). Using Lemma 15, we get that the required probability is at most

\[
\sum_{p_2} \frac{d}{\alpha n} \cdot q_{p_2} \cdot \left[ \frac{4d}{\alpha n} \cdot \left[ \frac{d\ell_{(p_1,i)}}{\alpha n} + 1 \right] \right] = \frac{d}{\alpha n} \cdot \ell_{(p_1,i)} \cdot \left[ \frac{4d}{\alpha n} \cdot \left[ \frac{d\ell_{(p_1,i)}}{\alpha n} + 1 \right] \right] = \frac{4d^2\ell_{(p_1,i)}^2}{(\alpha n)^3} \cdot \left[ 1 + \frac{\alpha n}{d\ell_{(p_1,i)}} \right].
\]

We will now add the requirement that \(c_{(p_1,i)}\) is one of player \(p_1\)’s semi-final pieces and sum up over all \(p_3\) candidate pieces and over all players \(p_3\).

Lemma 16. The probability that there are semi-final pieces \(a_{(p_1,r_1)}\), \(a_{(p_2,r_2)}\), \(a_{(p_2,1-r_2)}\), \(a_{(p_3,0)}\), and \(a_{(p_3,1)}\) for three players \(p_1\), \(p_2\), and \(p_3\), where both \(a_{(p_2,r_2)}\) and \(a_{(p_2,1-r_2)}\) overlap with \(a_{(p_1,r_1)}\), and both \(a_{(p_3,0)}\) and \(a_{(p_3,1)}\) overlap with \(a_{(p_2,r_2)}\) is at most \(\frac{4d^4}{\alpha n} + \frac{8d^4}{\alpha n^2}\).

Proof. As in the proof of Lemma 7, let \(R_{(p_1,i,r)}\) be the event that the candidate \(c_{(p_1,i)}\) is selected to be the semifinal piece \(a_{(p,r)}\). Recall that \(\text{Prob}[R_{(p_1,i,r)}] = d \cdot (\frac{1}{\alpha n}) \cdot (\frac{1}{\alpha n})^{d-1}\). There are \(n\) choices for player \(p_1\).
Thus by Lemma 15, our desired probability is at most
\[
n \left( \sum_{i=1}^{\alpha n} \frac{d}{\alpha n} \left( \frac{i-1}{\alpha n} \right)^{d-1} \frac{3d^2\ell_{(p_1,i)}}{(\alpha n)^3} \cdot \left[ 1 + \frac{\alpha n}{d\ell_{(p_1,i)}} \right] \right) \leq n \left( \frac{4d^3}{(\alpha n)^{d+3}} \sum_{i=1}^{\alpha n} \ell_{(p_1,i)}^2 (i-1)^{d-1} + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{(p_1,i)} (i-1)^{d-1} \right) \leq n \left( \frac{4d^3}{(\alpha n)^{d+3}} (\alpha n)^{d-2} (2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{(p_1,i)} (i-1)^{d-1} \right) \leq n \left( \frac{4d^3}{(\alpha n)^{d+3}} (\alpha n)^{d-2} (2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}} (\alpha n)^d \left( \frac{2\alpha n^2}{\alpha n} \right) \right) = \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}.
\]
The second inequality follows by Lemma 8. The third inequality follows from noting that, given that the \( \ell_{(p_1,i)}'s \) are nonincreasing, the sum is obviously maximized if each \( \ell_{(p_1,i)} \) is equal. That is, each \( \ell_{(p_1,i)} = \frac{2\alpha n^2}{\alpha n} \).

3.5 Computing the Probability of Failure

The probability that the total same-player-pee graph is not 2-colorable is at most \( \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} \). The probability that the implication graph contains a pair path of length three or more is at most \( \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} + \frac{32d^3}{\alpha^2(\alpha-d^2)} \). Thus we get that the probability that the maximum overlap of the final pieces is more than 2 is at most \( \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} + \frac{32d^3}{\alpha^2(\alpha-d^2)} \). By setting \( d = 2 \), and then setting \( \alpha \) to be sufficiently large, one can make this probability arbitrarily small. Hence, the probability that our caking cutting algorithm is not at least 2\( \alpha \)-fair is at most \( \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} + \frac{32d^3}{\alpha^2(\alpha-d^2)} \).

3.6 Approximate Cuts with a Weak Adversary

In this section, we show that even if the cut operations are only approximate, then approximate fairness is still achievable in \( O(n) \) complexity against a weak adversary, which must specify the relative error without knowing the value of the cake specified in the cut. For the proof, see appendix section B.

**Theorem 17.** If a protocol can only make \( 1 + \epsilon \) approximate queries against a weak adversary, then there is a randomized protocol for cake cutting that achieves \( O(1) \)-fairness in \( O(n) \) time.

4 Conclusion

The results in this paper suggest several interesting open questions. As in the balls and bins case, can we get a high probability result, perhaps at the cost of increasing by a constant factor the maximum load bound? Is linear query complexity achievable by randomized algorithms for exact fairness? But perhaps most interesting is to see how other balanced allocation results in the literature extend to the unrelated machines case. Analysis of balls and bins models have found wide application in areas such as load balancing [8]. In these situations, a ball represents a job that can be assigned to various bins/machines. Roughly speaking, load balancing of identical machines is to balls and bins, as load balancing on unrelated machines is to cake cutting. Unrelated machines is one of the standard models in the load balancing literature [1]. In the unrelated machines model there is a speed \( s_{i,j} \) that a machine \( i \) can work on a job \( j \). Assume that jobs can use more than one machine, and that machines can be shared. Then the total value of the machines to job \( j \) is \( \sum_i s_{i,j} \), and a \( c \)-fair allocation for job \( j \) would be a collection of machines, or portions of machines, that
can together process \( j \) at a speed of \( \sum_i \frac{a_{i,j}}{c_{i,j}} \). So it seems to us reasonable to presume the cake model, and balanced allocation lemmas, should have interesting applications in settings involving load balancing on unrelated machines.

References


A Our Cake Cutting Algorithm

Before turning to our Balanced Allocation Lemma, let us explain how our cake cutting protocol uses our Balanced Allocation Algorithm. Each player \( p \) has an initially unknown value function \( V_p \) that specifies how much that player values each subinterval of the unit interval. We imagine the player partitioning the cake into \( \alpha n \) pieces each of value \( \frac{1}{\alpha n} \). The \( i \)th such candidate piece of cake \( c(p,i) \) can be obtained using the following queries \((\text{Cut}_p(0, \frac{i-1}{\alpha n}), \text{Cut}_p(0, \frac{i}{\alpha n}))\). Our cake cutting protocol uses our Balanced Allocation Algorithm to obtain a final piece for each player such that every point of the cake is covered by at most \( O(1) \) of these final pieces. Because each player chooses only a constant number of candidate pieces, the query complexity is \( \Theta(n) \). Because the probability of success is \( \Theta(1) \), they expect to repeat it \( \Theta(1) \) times until they succeed.

Once each player has one final piece, we need to divide these pieces further so that the players have disjoint collections of cake intervals. This is done as follows. These \( n \) final pieces have \( 2n \) endpoints and these endpoints partition the cake into \( 2n \) pieces. Denote these by \( f_j \). For each piece \( f_j \) and each player \( p \), the player either wants all of \( f_j \) or none of it. For each \( j \), let \( S_j \) be the set of players wanting cake piece \( f_j \). Some players \( p \) may appear in more than one \( S_j \), but we have that \( |S_j| \leq k = O(1) \), because every point of the cake is covered by at most \( O(1) \) of player’s final pieces. For each piece \( f_j \), the players in \( S_j \) use any fair algorithm to partition \( f_j \) between them. Each such application has complexity \( \Theta(1) \) since it only involves \( \Theta(1) \) players. This protocol guarantees \( k\alpha \)-fairness. Consider player \( p \). For each \( j \) for which \( p \in S_j \), let \( v_{(p,j)} \) denote the amount he values piece \( f_j \). Note \( \sum_{j} v_{(p,j)} = V_p(\cup_{j} f_j) = V_p(\text{his final piece}) = \frac{1}{\alpha n} \). When fairly dividing \( f_j \), he receives a piece of \( f_j \) with value at least \( \frac{v_{(p,j)}}{k\alpha n} \). The total cake that he receives has total value \( \sum_{j} \frac{v_{(p,j)}}{k\alpha n} = \frac{1}{k\alpha n} \). Note that unlike all previous cake cutting algorithms, this one does not guarantee contiguous portions since a player’s final interval may be involved many different such subintervals \( f_j \).

B Proof of Weak Adversary Result

We start by defining an approximate cut.

\( ACut_p(\epsilon, x_1, \beta) \): This \( 1 + \epsilon \) approximate cut query returns an \( x_2 \geq x_1 \) such that the interval of cake \([x_1, x_2]\) has value approximately \( \beta \) according to player \( p \)'s value function \( V_p \). More precisely, \( x_2 \) satisfies \( \frac{1}{1+\epsilon}V_p(x_1, x_2) \leq \beta \leq (1+\epsilon)V_p(x_1, x_2) \).

Non-Adaptive Error: We say that \( ACut_p(\epsilon, x_1, \beta) \) has a nonadaptive error if each operation the algorithm first provides \( x_1 \) but not \( \beta \). The weak adversary, knowing the complete history but not \( \beta \), chooses a random variable \( E \) for the error with some distribution in the range \( [\frac{1}{1+\epsilon}, 1+\epsilon] \). When the algorithm provides \( \beta \), the operation \( ACut_p(\epsilon, x_1, \beta) \) returns the random variable \( x_2 = Cut_p(x_1, E \cdot \beta) \) such that \( V_p(x_1, x_2) = E \cdot \beta \).

**Theorem 18.** If a protocol can only make \( 1 + \epsilon \) approximate queries against a weak adversary, then there is a randomized protocol for cake cutting that achieves \( O(1) \)-fairness in \( O(n) \) time.

**Proof.** The algorithm as defined above chooses a random integer \( i \in [0, \alpha n - 1] \) and cuts out a piece starting at \( x_1 = Cut_p(0, \frac{i}{\alpha n}) \) and ending at \( x_2 = Cut_p(0, \frac{i+1}{\alpha n}) \) or equivalently at \( x_2 = Cut_p(x_1, \frac{1}{\alpha n}) \). If the second cut is replaced with the cut \( x_2 = ACut_p(\epsilon, x_1, \frac{1}{(1+\epsilon)\alpha n}) \) even with adaptive error, then the algorithm does not change significantly. The piece returned is no wider so overlaps with other player’s intervals are no more likely and the associated value, though perhaps a factor of \( (1+\epsilon)^2 \) more unfair, is still constant fair.

For the first cut \( x_1 = Cut_p(0, \frac{i}{\alpha n}) \), if the algorithm instead chooses a random real \( i \in [0, \alpha n - 1] \) instead of a random integer, the algorithm does not change significantly. This then become a cut at a uniformly chosen random value \( \beta = \frac{i}{\alpha n} \in [0, 1] \). If we replace this cut with an approximate cut with an non-adaptive adversary, it becomes a cut at value \( \beta' = E \beta \). But because error \( E \) is a random variable is independent of \( \beta, \beta' \) is basically also a uniformly chosen random value \( \beta' \in [0, 1] \). To see, this consider some fixed value
$b \in [\epsilon, 1 - \epsilon]$ not too close to the endpoints. We have

$$\Pr [\beta' \in [b, b + \delta b]] = \int_{e \in \left[\frac{1}{1+\epsilon}, 1+\epsilon\right]} \Pr \left[ \beta \in \left[\frac{b}{e}, \frac{b + \delta b}{e}\right] \right] \cdot \Pr [E = e] \, \delta e$$

$$= \int_{e \in \left[\frac{1}{1+\epsilon}, 1+\epsilon\right]} \frac{\delta b}{e} \cdot \Pr [E = e] \, \delta e = \delta b \cdot \left[ \int_{e \in \left[\frac{1}{1+\epsilon}, 1+\epsilon\right]} \frac{\Pr [E = e]}{e} \, \delta e \right].$$

This is a strange integration, but it is within $(1 + \epsilon)$ of one and it is constant with respect to $b$. Hence, $\Pr [\beta' \in [b, b + \delta b]] \approx \delta b$, meaning that $\beta'$ is uniformly chosen within $[\epsilon, 1 - \epsilon]$. \qed