A Randomized Algorithm For Cake Cutting

Jaisingh Solanki

A thesis submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements
for the degree of

Master of Science

Graduate Program in Computer Science
York University
Toronto, Canada

August 2007

A RANDOMIZED ALGORITHM FOR CAKE CUTTING

by **Jaisingh Solanki**

a thesis submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

A RANDOMIZED ALGORITHM FOR CAKE CUTTING

by **Jaisingh Solanki**

Examination Committee Members:

1. Jeff Edmonds
2. Patrick Dymond
3. Suprakash Datta
4. Neal Madras

Abstract

Cake cutting is a resource allocation problem in which a resource has to be divided *fairly* among a number of players. In this problem a player may not value all the parts of the resource in the same manner and different players may value the same part differently. A protocol(algorithm) to solve the cake cutting problem can learn about these valuations by asking queries to players. The goal of the protocol is to divide the resource fairly and make all the players *happy* by asking as few queries as possible. In 1984, Even and Paz gave an $O(n \log n)$ protocol and recently Edmonds and Pruhs proved that every deterministic protocol will ask $\Omega(n \log n)$ queries. In a separate paper, they also gave an $O(n)$ randomized approximate fair protocol with $O(1)$ success probability. We improved their algorithm by providing $O(n)$ randomized protocol with $O(1 - \frac{1}{poly(n)})$ success probability.

To my mother

Acknowledgements

I am extremely thankful to my supervisor Professor Jeff Edmonds for his guidance, support, patience and constant encouragement. He was friendly and approachable and gave me time for discussions whenever needed. I am also thankful to him for providing financial support in the form of research assistantship. He was not only a great supervisor to me but is also a great friend. It has been my privilege and great luck to work with him.

I am grateful to Professor Suprakash Datta, Professor Patrick Dymond and Professor Neal Madras for agreeing to serve on my supervisory committee and providing very useful feedback on my thesis. I am also thankful to Professor Franck van Breugel for giving me an opportunity to present my work in his weekly ”Concurrency group” meetings and for the encouragement provided.

I wish to thank graduate director Professor Richard Wildes for his help in arranging my thesis defence at a short notice. I also thank graduate assistant Mel Poteck for her help with administrative matters. I am thankful to the Computer Science and Engineering Department and NSERC for providing financial support.

I thank my fellow graduate students and friends especially Nassim Nasser, James Hyukjoon Kwon, Hai Feng Huang, Anton Belov, Nelson Moniz, Sergey Kulikov, Dusty Philips, Romil Jain, Ahmed Sabbir Arif, Amir Ali Ghadiri, Tomasz Robert Nykiel, Niloufar Shafiei and Shakil Khan for the wonderful company and cheerful environment in the department.

Finally, I would like to thank my mother, my sisters and all other family members in India for their constant love. I would like to thank my uncle Shyamji Parmar for his encouragement. I thank my wife Babita for proof-reading my thesis, and most importantly for her love, encouragement and understanding.

Table of Contents

Abstract **i**

Abstract **iv**

Acknowledgements **vi**

List of Tables **x**

List of Figures **xi**

1 Introduction **1**

1.1 Related Work 5

1.2 Organization of Thesis 7

2 Preliminaries **8**

2.1 Cake Cutting Problem 8

2.2 Balls and Bins Problem 11

2.2.1 Relation with Cake Cutting 12

2.3 Deterministic Algorithms for Cake Cutting 15

3 Randomized Cake Cutting Protocols **19**
3.1 Edmonds and Pruhs Approach 19
3.1.1 Implication Graph and Pair Paths 19
3.1.2 Two Types of Pair Paths 22
3.1.3 Balanced Allocation Algorithm 24
3.2 Our Approach 25
3.2.1 Probability of Player Being Bad 27

4 Initial Unsuccessful Approach **33**
4.1 Probability of Pair Cycle 36

5 Summary and Future Directions **41**
5.1 Open Problems 42

Bibliography **44**

A Proofs by Edmonds and Pruhs **47**
A.1 Final Piece Selection Algorithm 47
A.2 The Number of Vees 48
A.3 The Existence of Pair Paths 50
A.4 Coloring Same-Player-Vee Graphs 53
A.5 Computing the Probability of Failure 56

List of Tables

1.1 Summary of known results. An asterisk (*) means that the result holds for both choices. 5

List of Figures

2.1 Balls and bins version of cake. 14

2.2 Divisions of cake by players into equal number of pieces as per their own valuations 14

3.1 Players' two selected pieces and corresponding implication graph. 21

3.2 Pair Paths. 21

3.3 The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph. 22

3.4 Flowchart of Our Protocol. 28

4.1 The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph 34

4.2 The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph. 40

4.3 Cake distribution of 4 players with their selected semifinal pieces. 40

A.1 The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph. 51

1 Introduction

The cake cutting problem originated in the 1940's in the Polish mathematics community. It is a resource allocation problem where a cake (any resource which can be divided infinitely) has to be divided *fairly* among a number of players. An important point to note here is that a player may not value all the parts of the cake in the same manner and different players may value the same part differently. For example, consider the case where one player may want the icing and another may want the strawberries on a cake. These likings are not known apriori to the protocol solving the cake cutting problem. However, a protocol can learn about these likings by asking queries to players. The aim of the protocol is to make all the players *happy* by asking as few queries as possible.

A deterministic fair protocol with complexity $\Theta(n^2)$ was given in 1948 by Steinhaus in [14]. In 1984, Even and Paz [6] gave a deterministic divide and conquer fair protocol that has complexity $\Theta(n \log n)$. Recently, there has been several lower bound results for cake cutting. Sgall and Woeginger [13] showed that every exact fair protocol (deterministic or randomized) has complexity $\Omega(n \log n)$ if every portion given to players is restricted to be a contiguous piece of the cake. Edmonds and Pruhs [4] extended the lower bound to apply even when the protocol need to guarantee only *approximate fairness* and not necessarily assign contiguous portions. They proved that every deterministic approximate fair protocol for cake cutting has complexity $\Omega(n \log n)$. In the same paper, they also proved that every randomized approximate fair protocol has complexity $\Omega(n \log n)$ if answers to queries asked by protocol are approximations to actual answers.

Edmonds and Pruhs also gave in a separate paper [5], a randomized approximate fair protocol of complexity $O(n)$ with $O(1)$ success probability. Their protocol need not assign contiguous portions to players. We improve this by giving a randomized approximate fair protocol with complexity $O(n)$ with high probability[1] of success. We make this improvement by modifying the protocol of Edmonds and Pruhs [5]. An outline of their algorithm is as follows:

- Each player chooses independently and uniformly $2d$ pieces of value $\frac{1}{\alpha n}$ fraction of the value of whole cake, where α and d are appropriate constants.
- Select two semifinal pieces out of these $2d$ pieces for each player.
- From these semifinal pieces, build an *implication graph* and *same-player-vee graph*. Implication graph and same-player-vee graph are defined in Chapter 3.
- They defined the notion of a *bad* player in these graphs. Then under the assumption that no player is *bad*, narrow down to one final piece for every player such that each point of the cake has at most 2 players wanting it.

They proved that,

$$\text{Prob}[\textit{their protocol does not work}] \leq \text{Prob}[\textit{there exists a bad player}] \leq O(1).$$

[1] with probability at least $0(1 - \frac{1}{poly(n)})$

We tried to improve Edmonds and Pruhs protocol by relaxing the notion of *bad* but we were unsuccessful with that approach. We initially got some success with the approach but later we got more and more complex counterexamples to our approach. Every time we got rid of a counterexample (by modifying the protocol), we got a new counterexample. Our initial success kept our hopes alive and we persisted with this approach until we came up with a completely different protocol. This new protocol is based on one key observation of the Edmonds and Pruhs protocol, i.e.,

$$\mathrm{Prob}[player\ p\ is\ bad] \leq O(\frac{1}{n}).$$

From this observation, we came up with the following protocol:

- Independently run twice, the protocol by Edmonds and Pruhs upto the formation of *implication graph* and *same-player-vee graph*.
- Delete players that are *bad* for the run from the corresponding graphs.
- Each run is then completed by narrowing down to one final piece for every player in the run.
- Merge results from the two runs. This results into a situation where each point of the cake has at most $4 = 2 \times 2$ players (at most 2 players from each run) wanting it.

We delete players in our method. By doing so we might not assign any piece to deleted players. However, we also prove that it is unlikely that some player will be deleted from both the runs. In particular, we show that,

$$\begin{aligned}
& \mathrm{Prob}[our\ protocol\ does\ not\ work] \\
\leq\ & \mathrm{Prob}[there\ exists\ a\ player\ p\ that\ is\ bad\ in\ both\ runs] \\
\leq\ & n \times \mathrm{Prob}[player\ p\ is\ bad\ in\ both\ runs] \\
\leq\ & n \times (\mathrm{Prob}[player\ p\ is\ bad\ in\ one\ run])^2 \\
\leq\ & n \times O(\tfrac{1}{n})^2 \\
\leq\ & O(\tfrac{1}{n}).
\end{aligned}$$

We can execute the above two-run protocol k times and select the first successful execution. This improves the probability of success,

$$\begin{aligned}
& \mathrm{Prob}[none\ of\ the\ k\ executions\ work] \\
\leq\ & (\mathrm{Prob}[one\ execution\ does\ not\ work])^k \\
\leq\ & O(\tfrac{1}{n^k}).
\end{aligned}$$

Note that for each of the k executions of the two-run protocol, no point in the cake is wanted by more than 4 players. Since we select only one successful execution from k executions, number of players sharing any point in the cake is at most 4. We get the same probability of success if we run the Edmonds and Pruhs algorithm $k+1$ times instead of two times but then the number of players wanting any point of the cake could be as much as $2k+2$. As with the Edmonds and Pruhs protocol, our protocol may not assign contiguous pieces to players.

Cake cutting, and related fair allocation problems, are of wide interest in both social sciences and mathematical sciences. (See, for example, Sgall and Woeginger [13] for a nice overview). There are several books written on fair allocation problems such as cake cutting, that give more extensive overviews. (See, for example, [3, 12]). Some quick *Googling* reveals that cake cutting algorithms, and their analysis, are commonly covered by computer scientists in their algorithms and discrete mathematics courses.

All of the results that are known so far in the area of cake cutting are summarized in Table 1.

Deterministic vs. Randomized Protocol	Exact vs. Approx. Queries	Exact vs. Approx. Fairness	Contiguous vs. Non-contiguous Portions	Probability of failure	Complexity	Reference
Deterministic	Exact	*	*	0	$O(n \log n)$	[6]
*	*	Exact	Contiguous	0	$\Omega(n \log n)$	[13]
Deterministic	*	*	*	0	$\Omega(n \log n)$	[4]
*	Approx.	*	*	0	$\Omega(n \log n)$	[4]
Randomized	Exact	Approx.	Non-contiguous	$\Omega(1)$	$O(n)$	[5]
Randomized	Exact	Approx.	Non-contiguous	O$(\frac{1}{n^{O(1)}})$	$O(n)$	This thesis

Table 1.1: Summary of known results. An asterisk (*) means that the result holds for both choices.

1.1 Related Work

Cake cutting is closely related to the multiple-choice balls and bins problem. In the multiple-choice balls and bins model, d' of αn discrete bins are selected for each ball uniformly at random. Then we select one bin out of d' bins such that maximum number of balls in the bin is the smallest. This number is called as the *maximum load*. Later we will see that the balls and bins model is equivalent to the special case of the cake model in which all the players like the cake uniformly. Analysis of the balls and bins model has found wide applications 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, speed $s_{i,j}$ that a machine i can work on a job j is also specified. 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 job j at a speed of $\sum_i \frac{s_{i,j}}{cn}$.

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 Krishnamoorthy as described in [7]. They proved that any protocol must make $\Omega(n \log n)$ comparisons to compute the assignment. This result does not address query complexity i.e. the number of queries used by the protocol. Approx. fair protocols were introduced by Robertson and Webb [11]. 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 deterministic protocol [11, 10, 15] that achieves $O(1)$-fairness with $\Theta(n)$ cuts and $\Theta(n^2)$ evaluations.

There are several other objectives studied in the cake cutting setting, most notably, *max-min* fairness, and *envy-free* fairness. (See, for example, [3] for details).

The literature on balanced allocations is also rather large. A nice survey is given in [8]. We are not aware of any other results on balanced allocations for unrelated machines.

1.2 Organization of Thesis

The remainder of the thesis is organized as follows.

- In Chapter 2 we present the formal definition of the cake cutting problem. We also present key observations (properties) of the problem and briefly discuss two known deterministic algorithms.

- In Chapter 3, we describe Edmonds and Pruhs randomized algorithm in detail. We then, present our approach and our improved randomized algorithm for cake cutting and prove the correctness of our algorithm.

- In Chapter 4, we outline our initial approaches that in the end did not work.

- Finally, in Chapter 5, we summarize our results and provide directions for future work.

- In Appendix A, we provide the proofs given by Edmonds and Pruhs in [5]. These proofs are given as they were presented in their paper.

2 Preliminaries

In this chapter we first give a formal definition of the cake cutting problem in Section 2.1. We then provide some observations or key properties of the problem. In Section 2.2, we give a brief overview of the balls and bins problem and its connection with the cake cutting problem. In Section 2.3 we briefly describe two deterministic algorithms for cake cutting.

2.1 Cake Cutting Problem

The cake cutting protocol involves division of a resource among players who measure the resource in their own way. These satisfaction measures are unknown to the protocol, but the protocol can learn about these by asking questions to the players. The goal of the protocol is to make everyone *happy* by asking as few questions as possible. In the next section we formally state the cake cutting problem and also define the notion of *happiness*.

We denote the resource by $\mathcal{C}$, and denote n players by $1, \ldots, n$.

- We assume that $\mathcal{C}$ can be divided indefinitely. We model it by the interval $[0, 1]$ and call it a *cake*.

- Every player p has her own private value function V_p on $\mathcal{C}$. It is a function from the power set of $\mathcal{C}$ to $[0, 1]$, i.e., it takes any subset $X \subseteq \mathcal{C}$ as input and returns the value of that subset. It satisfies the following properties.

  - Additive: for all disjoint subsets $X, X' \subseteq \mathcal{C}$, $V_p(X \cup X') = V_p(X) + V_p(X')$.
  - Divisible: for every $X \subseteq \mathcal{C}$ and $0 \leq \lambda \leq 1$, there exist $X' \subseteq X$ with $V_p(X') = \lambda \cdot V_p(X)$.
  - Normalized: $V_p(\mathcal{C}) = 1$.

- All the V_p's are unknown to the protocol, but the protocol can learn about them by asking two kinds of queries to players:
  - Value query, $Val_p(x_1, x_2)$: this valuation query to player p returns a value of the interval $[x_1, x_2]$ of the cake for player p. That is, $Val_p(x_1, x_2) = V_p([x_1, x_2])$.
  - Cut query, $Cut_p(x_1, \alpha)$: this cut query to player p returns x_2 such that the interval $[x_1, x_2]$ has value α to player p. That is, $V_p([x_1, x_2]) = \alpha$. If there is no such point then it returns -1.

- The aim of the protocol is to divide the cake into $C_1, C_2, \ldots, C_n$ pieces such that
  - Piece C_p is given to player p.
  - Pieces must be disjoint, that is, C_p and C_q must be disjoint for all players $p \neq q$.
  - Every player is *happy*.

- Happiness of the player: There are many ways to define happiness, but two important ones are:

  - c-fair: if every player p receives C_p such that $V_p(C_p) \geq \frac{1}{cn}$.
  - Envy-free: if every player p receives C_p such that $V_p(C_p) \geq V_p(C_q)$ for every piece C_q which is given to player q. Every player thinks that she got the best piece of the cake.

- Optimization: The protocol has to ask as few queries as possible. The protocol will be charged for queries, any other computation is free.

Some important observations of the cake cutting problem are:

- Divisibility of V_p guarantees that value of the singleton point is zero (i.e. value of the $[x, x]$ is zero for any $x \in [0, 1]$). This makes this problem completely different from the allocation of indivisible resource among n players. One consequence of this is that the corresponding open and closed subintervals of $[0, 1]$ have the same value to any player p.

- If $c = 1$ in c-fair then we say that it is *exact* fair. Otherwise we say that it is *approximate* fair.

- We can always divide the cake into m pieces for some player p such that the value of each piece is $\frac{1}{m}$. This can be achieved by asking cut queries $Cut_p(0, \frac{i}{m}) = x_i$ for $1 \leq i \leq m - 1$ to p. The m pieces will then be $[0, x_1), [x_1, x_2), \ldots [x_{m-1}, 1]$. If we have to find any random piece $[a, b]$ out of these m pieces then it can be found in two cut queries. For this, we first generate any random $i \in \{1, 2, \ldots, m\}$, and ask cut queries $Cut_p(0, \frac{i-1}{m}) = a$ and $Cut_p(0, \frac{i}{m}) = b$.

In this thesis, we will focus on c-fair happiness criteria. In the next section we describe the connection between cake cutting problem and the balls and bins problem.

2.2 Balls and Bins Problem

In this section we give a brief overview of the balls and bins problem and its connection with the cake cutting problem. In the balls and bins problem, we throw n balls into n distinct bins. Thus each ball is placed in a bin chosen independently and uniformly at random. We are then interested in finding the maximum number of balls in any bin. There are various results known for the problem. (See, for example [8], for a nice overview). Some of the known results are given below.

Theorem 1 ([9]). *Assume n balls are thrown into n bins, with each ball choosing a bin independently and uniformly at random. Then the maximum number of balls in any bin is $\theta(\frac{\log n}{\log \log n})$ with probability $\Omega(1 - \frac{1}{poly(n)})$.*

Theorem 2 ([2]). *Assume n balls are thrown sequentially into n bins, each ball is placed in the least full bin at the time of the placement, among d bins, $d \geq 2$, chosen independently and uniformly at random. Then after all the balls are placed, the maximum number of balls in any bin is $\theta(\frac{\log \log n}{\log d})$ with probability $\Omega(1 - \frac{1}{poly(n)})$.*

In the light of Theorem 1, we can see that $d \geq 2$ is essential in Theorem 2.

Theorem 3 ([2]). *Assume each of the n balls have been assigned to d bins chosen independently and uniformly at random from n bins. Then there is an efficient method that, after knowing all the d balls assigned to each ball, picks one of the d bins for each player, so that maximum number of balls in any bin is $O(1)$ with probability $\Omega(1 - \frac{1}{poly(n)})$.*

Note that, in Theorem 3, we are making assignments offline, i.e., the algorithm knows d pieces for every player before assigning any piece. In contrast, in Theorem 2, we are assigning online, i.e., the algorithm has to assign a bin to the k^{th} ball without looking at the choices for any future balls.

2.2.1 Relation with Cake Cutting

Cake cutting is generalization of balls and bins. If every player likes cake uniformly than it is the same as balls and bins. But if the players have different preferences then the situation is much more complex and there is a lot more interdependence between events. In this generalized setting we proved Theorem 3. The online version of Theorem 2 for cake cutting is open.

The connection between balls and bins and uniform cake cutting is as follows. Imagine a case where we have n players who like the cake uniformly and we divide the cake into n equal pieces. Now if we assign players to pieces, independently and uniformly at random, then maximum number of players assigned to any piece is same as maximum number of balls placed in any bin if we map players to balls and cake pieces to bins. Figure 2.1 shows the balls and bins version of the cake where each player has divided the cake into 9 equal parts. In general, players may like cake differently (see Figure 2.2).

The generalization of Theorem 3 is proved by Edmonds and Pruhs is Lemma 4 as follows.

Lemma 4 (Balanced Allocation Lemma [5]). *Let $\alpha \geq 17$ be some sufficiently large constant. Each of n players has a partition of the unit interval $[0, 1]$, or cake, into αn disjoint candidate subintervals/pieces. Each player independently picks $d' = 2d = 4$ of her 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 cake is covered by at most 2 players.*

We improve Lemma 4 to Lemma 5 to have arbitrarily small failure probability. To achieve this we only need to weaken it by having constant times as many cuts and twice as much overlap.

Lemma 5 (Improved Balanced Allocation Lemma). *Let $\alpha \geq 17$ be some sufficiently large constant. Each of n players has a partition of the unit interval $[0, 1]$, or cake, into αn disjoint candidate subintervals/pieces. Each player independently picks $d' = k \times 2 \times 2d = 8k$ of her pieces uniformly at random, with replacement. Then there is an efficient method that, with probability $\Omega(1 - \frac{1}{n^k})$, picks one of the d' subintervals for each player, so that every point on the cake is covered by at most 4 players.*

Removing Conflict: 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. Each player has one final contiguous piece worth $\frac{1}{\alpha n}$ and every point of the cake is covered by at most 4 of player's final pieces. 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 $|S_j| \leq 4$, because every point of the cake is covered by at most 4 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 involves only $\Theta(1)$ players. In the process, the value of a player's final portion might get reduced but we can still guarantee that value of her portion is at least $\frac{1}{4\alpha n}$. Consider a player p. For each j for which $p \in S_j$, let $v_{\langle p,j \rangle}$ denote the amount she values piece f_j. Note $\sum_j v_{\langle p,j \rangle} = V_p(\cup_j f_j) = V_p(\text{his final piece}) = \frac{1}{\alpha n}$. When fairly dividing f_j, she receives a piece of f_j with value at least $\frac{v_{\langle p,j \rangle}}{k}$. The total cake that she receives has total value $\sum_j \frac{v_{\langle p,j \rangle}}{k} \geq \frac{1}{4\alpha n}$. Note that unlike all previous cake cutting algorithms, this one does not guarantee contiguous portions since a player's final interval may be involved in many different such subintervals f_j.

Once we remove all the conflicts, each player will receive a portion worth $\frac{1}{4\alpha n}$.

Independent Events: One difference between the balls and bins problem and the cake cutting problem is the independence of certain events. Consider 3 players and their valuations of cake as shown in Figure 2.1 (which corresponds to balls and bins problem) and Figure 2.2.

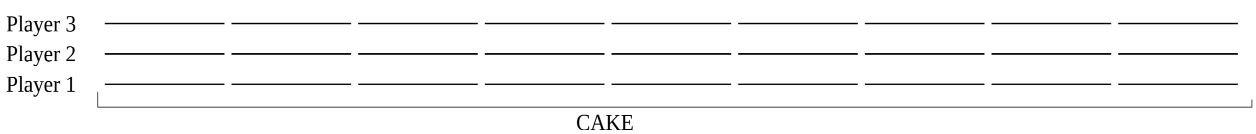

Figure 2.1: Balls and bins version of cake.

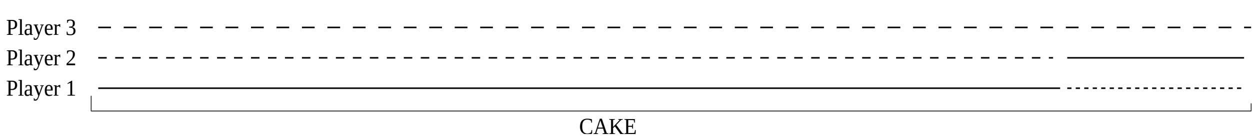

Figure 2.2: Divisions of cake by players into equal number of pieces as per their own valuations

Each player receives some piece chosen uniformly and randomly from their own partition. Let E_1 be the event that player 1 overlaps with player 2 and E_2 be the event that player 1 overlaps with player 3. These events are independent in Figure 2.1. In Figure 2.2, if event E_1 occurs then it is more likely that player has received her first piece and hence it is more likely that E_2 will also occur.

2.3 Deterministic Algorithms for Cake Cutting

In this section we briefly describe two deterministic algorithms. The first algorithm is known as the *Trimming Algorithm* and it was given by Banack and Knaster [14]. The second algorithm is known as the *Divide and Conquer Algorithm* and was given by Even and Paz [6].

First, consider a simple scenario to understand the cake cutting problem. Alice and Bob have been given a strawberry-chocolate cake to share. Bob suggests that he will cut the cake into two equal pieces and then give one piece to Alice. Alice is not happy with this. This is because Alice prefers more strawberries and she fears that even if Bob cuts the cake into two equal parts, he may give her a piece that has less strawberries. Bob likes both strawberries and chocolate. For similar reasons Bob doesn't want Alice to cut the cake and choose a piece for him. They can call Charlie to cut the cake and then give one piece each to both of them. However, they are unhappy with this approach as well. Let us consider that the value of the entire cake is 1 for both Alice and Bob. When Charlie divides the cake into two pieces A and B, it may happen that Bob feels that piece A is worth $\frac{1}{3}$ and piece B is worth $\frac{2}{3}$. Alice might feel the opposite. Now if Charlie gives them the pieces which they like $\frac{1}{3}$ then they both will be unhappy. A simple solution is that one of them cuts the cake into two equal pieces according to him/her and the other person chooses the piece which he/she likes more. This will ensure that both will get a piece worth at least $\frac{1}{2}$ according to their liking. If Bob divides the cake into two equal pieces, he is guaranteed to get exactly $\frac{1}{2}$ of the cake (no matter which piece Alice chooses). Since Alice likes the whole cake by value 1, she likes one of the pieces (two pieces of cake, cut by Bob) at least $\frac{1}{2}$. This method is called as *cut and choose* method and results in fair allocation.

Next we present two deterministic algorithms for the general case. Algorithm 1 is a iterative algorithm and requires $O(n^2)$ queries. It first finds out points c_i (by asking cut queries) for each player i such that value of the piece $[0, c_i]$ is equal to $\frac{1}{n}$ for player i. Note that the value of the piece $[c_i, 1]$ is equal to $\frac{n-1}{n}$ for player i. If we give piece $[0, c_{i_{min}}]$ to player i_{min}, where $c_{i_{min}}$ is the minimum of all the c_i's, then player i_{min} will be *happy*. Also, every other player values the remaining cake at least $\frac{n-1}{n}$ so the algorithm can iterate and make the other players happy as well. An outline of the algorithm follows.

Algorithm 2 is known as *Divide and Conquer Algorithm* and was given by Even and Paz [6]. First, it finds the middle point of the cake for every player by asking cut queries. Then, it divides the cake into two parts, C_1 and C_2. Half of the players are then happily assigned to part C_1 and the remaining players are happily assigned to part C_2. By happily, we mean that players are guaranteed that they will receive a fair piece from the assigned part. An outline of the algorithm follows. The query complexity of Algorithm 2 is $O(n \log n)$.

Algorithm 1 Cake Division Protocol - Trimming

Pre-cond: Set $P = \{1, 2, \ldots, n\}$ of n players with their own private value function V_i on the cake.

Post-cond: Division of cake into disjoint pieces $[a_1, b_1], [a_2, b_2] \ldots [a_n, b_n]$ such that $V_i([a_i, b_i]) \geq \frac{1}{n}$ for $1 \leq i \leq n$.

a=0 and $Q = \varnothing$
loop
  exit when $|Q| = n$
  for every $i \in P - Q$ **do**
    $c_i = Cut_i(a, \frac{1}{n})$.
  end for
  i_{min}= the $i \in P - Q$ that minimizes c_i
  $[a_{i_{min}}, b_{i_{min}}] = [a, c_{i_{min}}]$
  $a = c_{i_{min}}$
  $Q = Q + i_{min}$
  all parts $[a_i, b_i]$ for each $i \in P$
end loop

Algorithm 2 Cake Division Protocol - Divide and Conquer

Pre-cond: Cake= $[a, b]$ and set $P = \{1, 2, \ldots, n\}$ of n players with their own private value function V_i on the cake.

Post-cond: Division of cake into disjoint pieces $[a_1, b_1], [a_2, b_2] \ldots [a_n, b_n]$ such that $V_i([a_i, b_i]) \geq \frac{V_i([a,b])}{n}$ for $1 \leq i \leq n$.

if n=1 **then**
    $[a_1, b_1] = [a, b]$
    exit
end if
Let k=$\lfloor n/2 \rfloor$.
for i= 1 to n **do**
    $v_i = V_i([a, b])$
    $c_i = Cut_{p_i}(a, \frac{k \cdot v_i}{n})$
end for
middle = k^{th} highest element in $\{c_1, c_2, \ldots, c_n\}$
Rename first k players whose $c_i \leq c_{middle}$ as $\{1, 2, \ldots, k\}$ and recurse them on $[a, b] = [0, c_{middle})$
Rename remaining players as $1, 2, \ldots, n-k$ and recurse them on $[a, b] = [c_{middle}, 1]$

3 Randomized Cake Cutting Protocols

In this chapter, we first present Edmonds and Pruhs randomized algorithm for cake cutting, as given in [5], in Section 3.1. We then, present our approach and our improved randomized algorithm for cake cutting in Section 3.2. We also prove the correctness of our algorithm in Section 3.2. The main result proved by us is the *Improved Balanced Allocation Lemma*. This lemma is used in the design of our cake cutting protocol.

3.1 Edmonds and Pruhs Approach

Edmonds and Pruhs presented the *Balanced Allocation Lemma*(Lemma 4) in [5]. To prove this lemma, they defined many new interesting concepts such as *implication graph*, *same-player-vee graph*, and *pair path*. Since our protocol makes use of these concepts, we give their definitions in the next section for easy reference.

3.1.1 Implication Graph and Pair Paths

Let $c_{\langle p,i \rangle}$ denote the $i^{th}, (i \in [1, \alpha n])$ candidate piece for player p. Suppose that every player has chosen two semifinal pieces $a_{\langle p,0 \rangle}$ and $a_{\langle p,1 \rangle}$ (in fact we choose these semifinal pieces for every player from the d' pieces chosen independently for every player). We can then construct the implication graph as defined below.

Definition 6. Implication Graph: *The vertices of the implication graph IG are the $2n$ pieces $a_{\langle p,r \rangle}$, $1 \leq p \leq n$ and $0 \leq r \leq 1$. If piece $a_{\langle p,r \rangle}$ intersects piece $a_{\langle q,s \rangle}$, then there is a directed edge from piece $a_{\langle p,r \rangle}$ to piece $a_{\langle q,1-s \rangle}$ and similarly from $a_{\langle q,s \rangle}$ to $a_{\langle p,1-r \rangle}$.*

The intuition behind the above definition is that if a player p gets $a_{\langle p,r \rangle}$ as her final piece, then player q must get piece $a_{\langle q,1-s \rangle}$ if p's and q's pieces are not to overlap. Similarly if q gets $a_{\langle q,s \rangle}$, then p must get $a_{\langle p,1-r \rangle}$. As an example, Figure 3.1 gives a subset of the semifinal pieces selected from the candidate pieces. The corresponding implication graph is also given in Figure 3.1.

Definition 7. Pair Path: *A pair path in an implication graph is a directed path between two pieces for one player.*

In Figure 3.1, there are two pair paths of length three from the first player's left semifinal piece to her right and two pair paths of length two from the fourth player's left semifinal piece to her right. Two such pair paths from the implication graph in Figure 3.1 are shown in Figure 3.2.

Note that such paths are problematic because they effectively imply that if the first player gets her left semifinal piece as his final piece then she must get her right piece too. Edmonds and Pruhs prove that if the implication graph IG does not contain pair paths then the following algorithm selects a final piece for each player in such a way that these final pieces are disjoint.

Final Piece Selection Algorithm: We repeatedly pick an arbitrary player p that has not selected

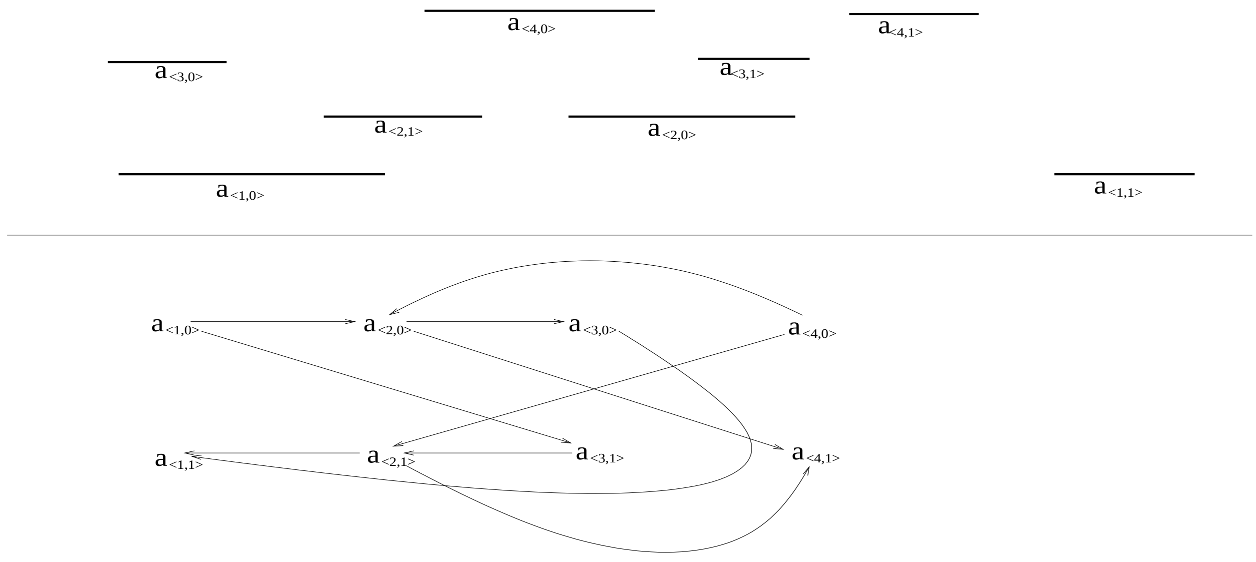

Figure 3.1: Players' two selected pieces and corresponding implication graph.

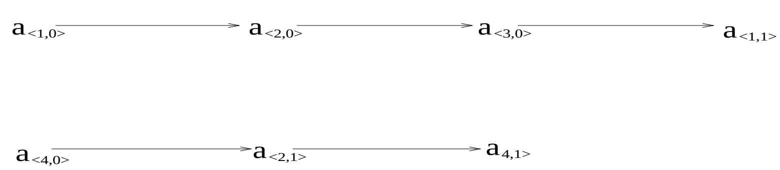

Figure 3.2: Pair Paths.

a final piece. We pick the piece $a_{\langle p,0\rangle}$ as the final piece for p. Further, we pick as final pieces all those pieces in IG that are reachable from $a_{\langle p,0\rangle}$ in IG.

Lemma 8. *If an implication graph IG 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. (See Section A.1 for a proof of this lemma.)*

3.1.2 Two Types of Pair Paths

They gave a lemma to compute the probability of having a pair path in an implication graph. First, they observed the vital difference between pair paths of length two and pair paths of length three or more. Note that a pair path occurs when there is a *vee* among the semifinal pieces. They defined 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 3.1. To understand the connection between pair paths and vees see Figure 3.3. Figure

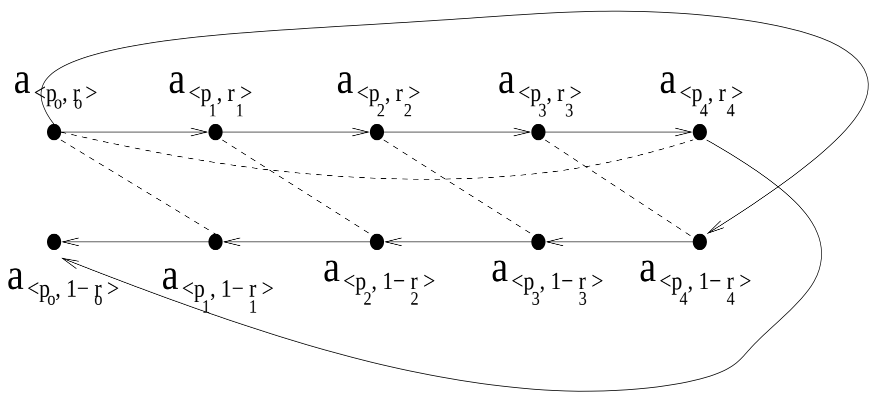

Figure 3.3: The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph.

shows a pair path of length 5 from $a_{\langle p_0,r_0\rangle}$ to $a_{\langle p_0,1-r_0\rangle}$. For this pair path to exist in IG, we need the vee formed by $a_{\langle p_0,r_0\rangle}$, $a_{\langle p_1,1-r_1\rangle}$ and $a_{\langle p_4,r_4\rangle}$ among semifinal pieces. We also need the directed edges $\left\langle a_{\langle p_1,r_1\rangle}, a_{\langle p_2,r_2\rangle}\right\rangle$, $\left\langle a_{\langle p_2,r_2\rangle}, a_{\langle p_3,r_3\rangle}\right\rangle$ and $\left\langle a_{\langle p_3,r_3\rangle}, a_{\langle p_4,r_4\rangle}\right\rangle$ in the implication graph IG.

They proved the following lemma that bounds the expected number of vees in the implication graph.

Lemma 9. *If each player only chooses 2 semifinal pieces then the expected number of vees in IG can be as high as $\Theta(n^2)$, which would be disastrous for us. However, if two brackets of $d = 2$ pieces are chosen and these are narrowed down to two semifinal piece then the expected number of vees in IG is at most $\frac{16d^3}{\alpha^2}n$. (See Section A.2 for a proof of this lemma.)*

Using Lemma 9 they proved the following lemma that bounds the probability of implication graph having pair paths of length three or more.

Lemma 10. *The probability that the implication graph IG contains a pair path of length at least three is at most $\frac{32d^5}{\alpha^2(\alpha-4d^2)}$. (See Section A.3 for a proof of this lemma.)*

A pair path of length two occurs if and only if the implication graph contains a *same-player-vee*. A *same-player-vee* is a vee where both of the base pieces belong to the same player. That is, there is a center piece $a_{\langle p,r\rangle}$ and two bases $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$. For example, see pieces $a_{\langle 4,0\rangle}$, $a_{\langle 2,0\rangle}$ and $a_{\langle 2,1\rangle}$ in Figure 3.1.

To get around the problem of same-player-vees, they introduced the *same-player-vee graph*.

Definition 11. Same-player-vee Graph: *The vertices of the same-player-vee graph SG are the n players p, $1 \leq p \leq n$. If player p and player q are involved in same-player-vee with player p in the center then there is a directed edge from p to q.*

Then they proved the following lemma to partition the players into two groups such that there is no same-player-vee involving two players in the same partition.

Lemma 12. *The probability that the same-player-vee graph is not $w = 2$ colourable is at most $\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}$. (See Section A.4 for a proof of this lemma.)*

They proved Lemma 12 by bounding the probability of same-player-vee graph having a path of length two. Note that if the same-player-vee graph does not have a path of length two then we can colour the graph by 2 colours such that if p and q have an edge between them then they receive different colours. If a node of the same-player-vee graph is at the head of a directed edge then colour it red, if at the tail, blue, and otherwise red. A node is only forced to be both red and blue if there is a directed path of length two in the graph.

3.1.3 Balanced Allocation Algorithm

Then they gave following algorithm which can be used as an efficient method mentioned in the Balanced Allocation Lemma:

- **Step 1:** Independently, for each player $p \in [1, n]$ and each $r \in [0, 1]$, randomly choose $d = 2$ of the candidate pieces $c_{\langle p,i \rangle}$ to be in the quarterfinal bracket $A_{\langle p,r \rangle}$.
- **Step 2:** In each quarterfinal bracket $A_{\langle p,r \rangle}$, pick as the semifinal piece $a_{\langle p,r \rangle}$, the piece that intersects the fewest other candidate pieces $c_{\langle q,j \rangle}$. If we are unlucky and the Implication Graph contains a pair path of length greater than or equal to 3, then halt(Lemma 10).
- **Step 3:** Construct and vertex colour the same-player-vee graph using the greedy colouring algorithm using at most $w = 2$ colours. This is easy if the graph does not have paths of length two or more (Lemma 12). Let S_h be the subgraph of the implication graph containing only those players coloured h. This ensures that Implication Graph restricted to S_h contains

no pair paths of length 2.

- **Step 4:** For each S_h, pick the final piece for each player involved in S_h by applying the Final Piece Selection Algorithm to S_h. Because the Implication Graph on S_h 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_h covers this point.
- **Step 5:** Conclude that for any point in the cake, the final pieces of at most $w = 2$ players cover this point.

3.2 Our Approach

In this Section we present our work and the improved randomized algorithm. From the above discussions, it can be observed that the Balanced Allocation Algorithm does not fail if there are no players that are *bad* according to the following definition.

Definition 13. Bad Player: *A player p is bad if*

- *a pair path of length three or more starting with p exists in the implication graph, or*
- *a path of length two or more starting with p exists in the same-player-vee graph.*

If we remove all the bad players with corresponding edges from the implication graph and same-player-graph before Step 3 of the Balanced Allocation Algorithm then it will always produce disjoint final pieces.

Run: A run of our algorithm is same as the Balance Allocation Algorithm, except that in our case it removes all the bad players from implication graph and same-player-vee graph before Step 3 of the Balanced Allocation Algorithm.

The only problem with the above definition of run is that bad players do not get any portion of the cake. Later we will see in Lemma 17 that probability that some player p is bad in a run is at most $O(1)$. Therefore, the probability that a run will not give pieces to every player is at least $O(1)$. However, if we execute two independent runs then probability that both runs will delete the same player p is at most $O(\frac{1}{n})$ (see Lemma 18).

Execution: We define an execution as a following sequence of steps:

1. Independently start two runs, run_1 and run_2 upto formation of implication graph and same-player-vee graph.

2. If some player p is bad in both the runs then halt the execution.

3. If some player player p is good (i.e. p is not bad) in both the runs then delete it from run_1.

4. Finish both the runs.

Note that an execution will halt only if there is a player that is bad in both the runs. The probability that some player is bad in both of the two independent runs is at most $O(\frac{1}{n})$ (see Lemma 18). In each run, number of players wanting any point of the cake is at most 2. Hence, amongst the two runs, each point of the cake will be shared by at most 4 players in the execution. We still have to make the final pieces disjoint. We do this by removing conflict among pieces (See Section 2.2.1). We present our protocol for the cake cutting problem as follows:

Protocol:

- Execute k independent executions.
- If all executions are unsuccessful then fail otherwise select the first successful execution.
- Assign the final pieces according to this execution. Every player receives a final piece worth $\frac{1}{\alpha n}$.
- Remove conflicts among final pieces. Each player receives portion worth $\frac{1}{4\alpha n}$.

Note that protocol will fail only if all k executions halt and this happens with probability at most $O(\frac{1}{n^k})$ (see Lemma 18). So our protocol succeeds with high probability.

3.2.1 Probability of Player Being Bad

We observe the following fact from Lemma 28(see Section A.2).

Lemma 14. *The expected number of vees with any particular player p in the center in the implication graph IG is at most $\frac{16d^3}{\alpha^2}$.*

By using Lemma 14 we proved the following lemma.

Lemma 15. *The probability that the implication graph IG contains a pair path of length at least three starting with player p is at most $\frac{32d^5}{\alpha^2(\alpha-4d^2)} \cdot \frac{1}{n}$.*

Proof. This proof is constructed by making small changes in the proof of Lemma 31. Consider a player p and let $\mathcal{V}_p$ be the set of all 3-tuples representing all possible vees in IG with p in the center of the vee and for $V \in \mathcal{V}_p$, let $\mathcal{P}_z(V)$ be the set of all possible pair paths of length z that include the vee V.

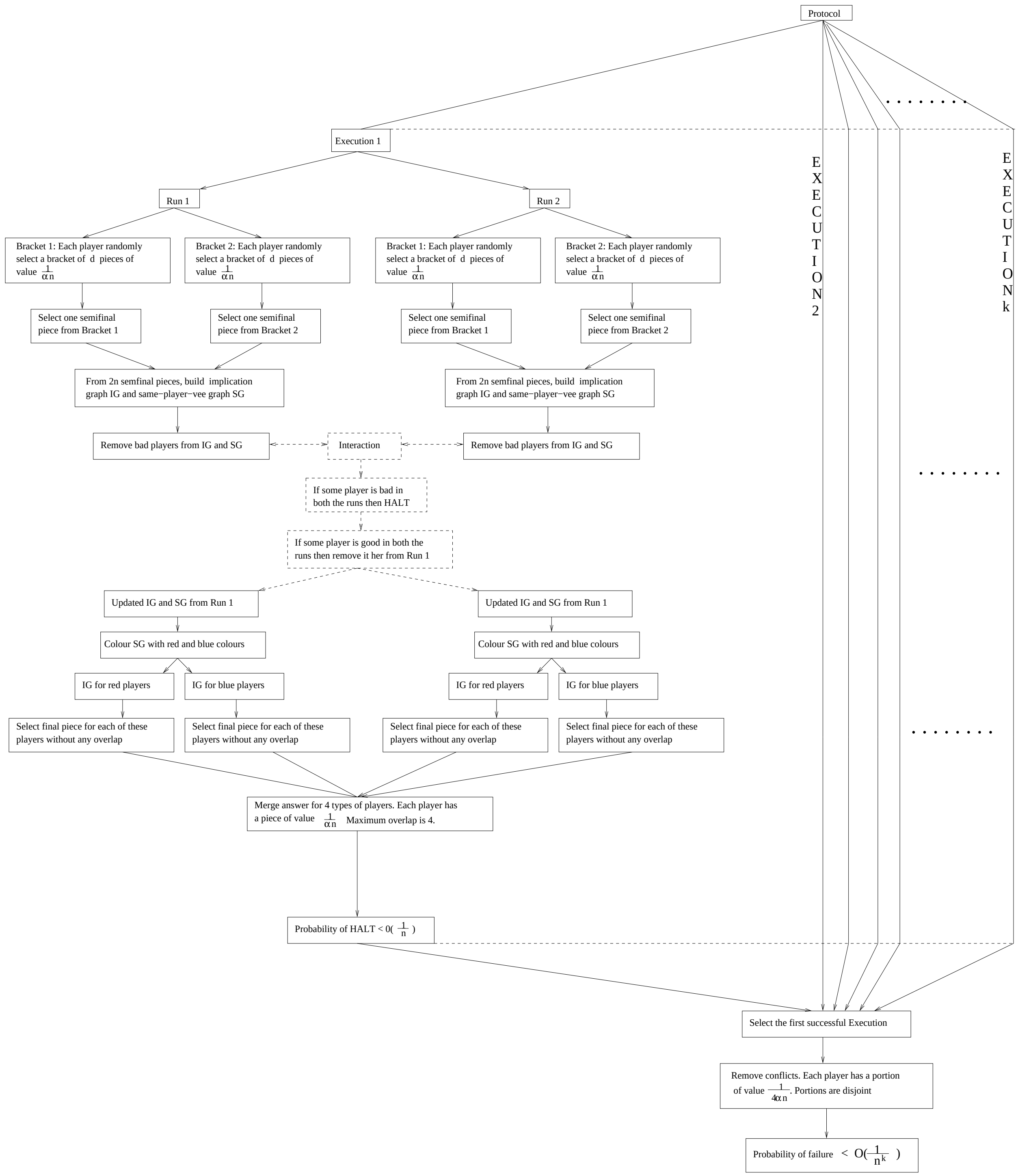

Figure 3.4: Flowchart of Our Protocol.

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

The probability that IG contains a pair path of length at least three starting with player p is at most

$$
\begin{aligned}
& \sum_{z=3}^{n} \sum_{V\in\mathcal{V}_p} \sum_{P\in\mathcal{P}_z(V)} \mathrm{Prob}[P \in IG] \\
\leq\ & \sum_{z=3}^{n} \sum_{V\in\mathcal{V}_p} \sum_{P\in\mathcal{P}_z(V)} \mathrm{Prob}[V \in IG] \cdot \Pi_{i=1}^{z-2}\mathrm{Prob}[I_i \in IG] && (3.1)\\
\leq\ & \sum_{z=3}^{n} \sum_{V\in\mathcal{V}_p} \mathrm{Prob}[V \in IG] \sum_{P\in\mathcal{P}_z(V)} \left(\frac{2d^2}{\alpha n}\right)^{z-2} && (3.2)\\
\leq\ & \sum_{z=3}^{n} \sum_{V\in\mathcal{V}_p} \mathrm{Prob}[V \in IG] \left(\binom{2n}{z-3}(z-3)!\right) \left(\frac{2d^2}{\alpha n}\right)^{z-2} && (3.3)\\
\leq\ & \sum_{z=3}^{n} (2n)^{z-3} \left(\frac{2d^2}{\alpha n}\right)^{z-2} \sum_{V\in\mathcal{V}_p} \mathrm{Prob}[V \in IG] \\
\leq\ & \sum_{z=3}^{n} (2n)^{z-3} \left(\frac{2d^2}{\alpha n}\right)^{z-2} \left(\frac{16d^3}{\alpha^2}\right) && (3.4)\\
\leq\ & \frac{8d^3}{\alpha^2 n} \sum_{z=3}^{n} \left(\frac{4d^2}{\alpha}\right)^{z-2} \\
\leq\ & \frac{8d^3}{\alpha^2 n} \left(\frac{4d^2}{\alpha}\right) \left(\frac{1}{1-4d^2/\alpha}\right) \\
=\ & \frac{32d^5}{\alpha^2(\alpha-4d^2)} \cdot \frac{1}{n}
\end{aligned}
$$

The inequality in line 3.1 follows from Lemma 30 (see Section A.3) and the inequality in line 3.2 follows from Lemma 27(see Section A.2). The inequality in line 3.3 holds since there are z-3 pieces in P that are not part of the vee V. The inequality in line 3.4 follows from Lemma 14. □

Now we show that, probability that same-player-vee graph SG has a path of length two starting with player p is at most $O(\frac{1}{n})$. Recall that we put the directed edge from p to q in the same-player-vee graph if one of player p's two semifinal pieces, namely $a_{\langle p,0\rangle}$ or $a_{\langle p,1\rangle}$, overlap with both of player q's two semifinal pieces, namely $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$. Hence, a path of length 2 consists of semi-final pieces $a_{\langle p_1,r_1\rangle}$, $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_2,1-r_2\rangle}$, $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ for three players p_1, p_2, and p_3, where both $a_{\langle p_2,r_2\rangle}$ and $a_{\langle p_2,1-r_2\rangle}$ overlap with $a_{\langle p_1,r_1\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$. We compute the probability of such paths by modifying the proof given in Lemma 37(see Section A.3).

Lemma 16. *Consider a player p_1. The probability that there are semi-final pieces $a_{\langle p_1,r_1\rangle}$, $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_2,1-r_2\rangle}$, $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ for three players p_1, p_2, and p_3, where both $a_{\langle p_2,r_2\rangle}$ and $a_{\langle p_2,1-r_2\rangle}$ overlap with $a_{\langle p_1,r_1\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$ is at most $\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}$.*

Proof. Let $\ell_{q,i}$ be the number of candidate pieces of the other players that overlap with the candidate piece $c_{\langle q,i\rangle}$ of player q. Without loss of generality, let us renumber q's candidate pieces in non-increasing order by $\ell_{\langle q,i\rangle}$, that is, $\ell_{\langle q,i\rangle} \geq \ell_{\langle q,i+1\rangle}$.

Let $R_{\langle p,i,r\rangle}$ be the event that the candidate $c_{\langle p,i\rangle}$ is selected to be the semifinal piece $a_{\langle p,r\rangle}$. It is proved in Lemma 28 (see Section A.2) that $\mathrm{Prob}[R_{\langle p,i,r\rangle}] = d \cdot (\frac{1}{\alpha n}) \cdot (\frac{i-1}{\alpha n})^{d-1}$. There are αn

choices for $a_{\langle p_1, r_1 \rangle}$, Thus by Lemma 36 (see Section A.4), our desired probability is at most

$$\begin{aligned}
& \sum_{i=1}^{\alpha n} \frac{d}{\alpha n} \left(\frac{i-1}{\alpha n} \right)^{d-1} \frac{4d^2 \ell_{\langle p_1, i \rangle}^2}{(\alpha n)^3} \cdot \left[1 + \frac{\alpha n}{d \ell_{\langle p_1, i \rangle}} \right] \\
\leq \; & \frac{4d^3}{(\alpha n)^{d+3}} \sum_{i=1}^{\alpha n} \ell_{\langle p_1, i \rangle}^2 (i-1)^{d-1} + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{\langle p_1, i \rangle} (i-1)^{d-1} \\
\leq \; & \frac{4d^3}{(\alpha n)^{d+3}} (\alpha n)^{d-2} (2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{\langle p_1, i \rangle} (i-1)^{d-1} \\
\leq \; & \frac{4d^3}{(\alpha n)^{d+3}} (\alpha n)^{d-2} (2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}} (\alpha n)^d \left(\frac{2\alpha n^2}{\alpha n} \right) \\
= \; & \left(\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} \right) \cdot \frac{1}{n}
\end{aligned}$$

The second inequality follows from Lemma 29 (see Section A.4). The third inequality is due to the fact that since the $\ell_{\langle p_1, i \rangle}$'s are non-increasing, the sum is obviously maximized if each $\ell_{\langle p_1, i \rangle}$ is equal. That is, each $\ell_{\langle p_1, i \rangle} = \frac{2\alpha n^2}{\alpha n}$. □

Lemma 17. *If we build the implication graph IG and same-player-vee SG graph then the probability that a player p is bad is at most $O(\frac{1}{n})$.*

Proof. Recall that a player p is bad if there exists a pair path of length three or more starting with it in IG or a path of length two starting with it in SG, i.e.,

$$\begin{aligned}
& \mathrm{Prob}[\textit{player } p \textit{ is bad}] \\
\leq \; & \mathrm{Prob}[\textit{player } p \textit{ is bad in } IG] + \mathrm{Prob}[\textit{player } p \textit{ is bad in } SG] \\
\leq \; & \frac{32d^5}{\alpha^2(\alpha - 4d^2)} \cdot \frac{1}{n} + \left(\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} \right) \cdot \frac{1}{n} \qquad (3.5) \\
\leq \; & \frac{32d^6}{\alpha^2} \cdot \frac{1}{n}
\end{aligned}$$

The inequality in line 3.5 above follows from Lemma 15 and Lemma 16. □

Lemma 18. *If we build two independent pairs* $(IG1, SG1)$ *and* $(IG2, SG2)$ *of the implication graph and the same-player-vee graph then probability that some player* p *is bad in both the pairs is at most* $O(\frac{1}{n})$.

Proof.

$$\begin{aligned}
& \text{Prob}[\mathit{some\ player\ p\ is\ in\ }(IG1, SG1)\ \mathit{and}\ (IG2, SG2)] \\
\leq\ & \sum_{p=1}^{n} \text{Prob}[\mathit{player\ p\ is\ bad\ in\ }(IG1, SG1)\ \mathit{and}\ (IG2, SG2)] \\
\leq\ & \sum_{p=1}^{n} \text{Prob}[\mathit{player\ p\ is\ bad\ in\ }(IG1, SG1)] \cdot \text{Prob}[\mathit{player\ p\ is\ bad\ in\ }(IG1, SG1)] \\
\leq\ & \sum_{p=1}^{n} (\frac{32d^6}{\alpha^2} \cdot \frac{1}{n})^2 \qquad (3.6) \\
\leq\ & \frac{1024d^{12}}{\alpha^4} \cdot \frac{1}{n}
\end{aligned}$$

The inequality in line 3.6 follows from Lemma 17. □

4 Initial Unsuccessful Approach

In this chapter we briefly describe our initial attempts in coming up with an improved randomized protocol for cake cutting. We needed some structure that appears in the implication graph IG with low probability and if it does not appear then we can solve our problem. Edmonds and Pruhs used *pair path* as their structure. If we carefully observe their Final Piece Selection Algorithm then we can find that pair path in one direction (i.e. having pair path from $a_{\langle p,r\rangle}$ to $a_{\langle p,1-r\rangle}$ but not from $a_{\langle p,1-r\rangle}$ to $a_{\langle p,r\rangle}$) is not really problematic. It just implies that we should select $a_{\langle p,1-r\rangle}$ for player p. We found that as long as we have pair paths in one direction only, we can solve the problem. However, pair paths in both directions will be problematic.

We can correlate this with the 2-SAT problem. In the 2-SAT problem we want to check whether given 2-SAT formula F is satisfiable or not. To solve this problem we build a directed graph G from the given 2-SAT formula F. For each variable x_i, we put two vertices x_i and $\bar{x}_i$ (not of x_i) in G. The clauses then correspond to edges. Clause $(x \vee y)$ gives edges from $\bar{x}$ to y and $\bar{y}$ to x. If $\bar{x}$ is true then y must be true to make clause $(x \vee y)$ true. Graph G has the property that F is satisfiable if and only if there is no directed cycle in G containing x and $\bar{x}$ for some variable x. We try to find out the satisfying assignment for F by selecting one of the vertex x_i or $\bar{x}_i$ (setting variable 1 or 0 respectively). Note that if we select x_i then we should not select $\bar{x}_i$ or vice versa. This procedure to select the vertices is almost similar to the Final Piece Selection Algorithm except that it works even if we have path from x_i to $\bar{x}_i$ but not the other way around for some i. So the procedure fails only if G contains a directed cycle.

Pair Cycle:A pair cycle in the implication graph is a directed cycle containing both semifinal pieces $a_{\langle p,0\rangle}$ and $a_{\langle p,1\rangle}$ for some player p. Note that a pair cycle is same as having a directed path in both directions, from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$ and $a_{\langle p,1\rangle}$ to $a_{\langle p,0\rangle}$ for some player p. For example, see Figure 4.1. It shows a pair cycle which contains both the semifinal pieces of player 1. Note that it has two pair paths.

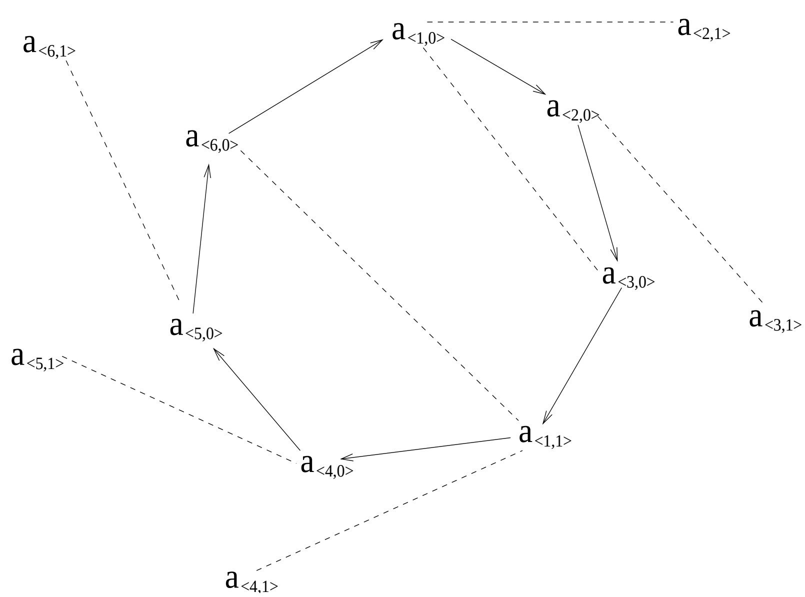

Figure 4.1: The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph

Next we prove that if the implication graph IG does not contain pair cycle then the following algorithm selects a final piece for each player in such a way that these final pieces are disjoint.

Modified Final Piece Selection Algorithm: We repeatedly pick an arbitrary player p that has not selected a final piece. If there is no pair path from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$ then we select the piece $a_{\langle p,0\rangle}$ as the final piece for p. Otherwise we select the piece $a_{\langle p,1\rangle}$ as the final piece for p. Further, we pick as final pieces all those pieces in IG that are reachable from the selected piece $a_{\langle p,r\rangle}$ in IG.

Lemma 19. *If an implication graph IG of the semifinal pieces does not contain a pair cycle, then the Modified 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 piece to player p. There are two possible cases depending on the piece assigned to p:

- **Case 1:** Assigned piece is $a_{\langle p,0\rangle}$. This happens only when there is no directed path from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$. This iteration will force the assignment of at most one piece to any player because if there is a player q such that both $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$ are reachable from $a_{\langle p,1\rangle}$ then by Lemma 24 there will be a directed path from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$.

- **Case 2:** Assigned piece is $a_{\langle p,1\rangle}$. This happens only when there is a directed path from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$. This iteration will force the assignment of at most one piece to any player because if there is player q such that both $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$ are reachable from $a_{\langle p,1\rangle}$ then by Lemma 24 there will be directed path from $a_{\langle p,1\rangle}$ to $a_{\langle p,0\rangle}$. In other words we will have a pair cycle in IG.

Similarly, if this same iteration forces player q to be assigned say to $a_{\langle q,r\rangle}$, then we need to prove that she has not already been assigned $a_{\langle q,1-r\rangle}$ during an earlier iteration. If assigning $a_{\langle p,s\rangle}$ forces $a_{\langle q,r\rangle}$, then there is a path from the one to the other. Hence, by Lemma 23, there is a path from $a_{\langle q,1-r\rangle}$ to $a_{\langle p,1-s\rangle}$. Hence, if $a_{\langle q,1-r\rangle}$ had been previously assigned, then player p would have been forced to $a_{\langle p,1-s\rangle}$ 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. □

4.1 Probability of Pair Cycle

Having solved the problem when there are no pair cycles, what remains is to prove that pair cycles do not occur with high probability in the implication graph IG. We anticipated this probability to be less than $O(\frac{1}{n})$. To understand our anticipation, let us assume that edges in IG occur independently and with probability $O(\frac{1}{\alpha n})$. Let us calculate the probability of pair paths in IG. Any pair path of length $z \geq 2$ requires z edges and $z+1$ vertices but only z players. We get the following probability calculation for implication graph having pair paths.

$$\begin{aligned}
&\mathrm{Prob}[IG\ contain\ pair\ paths] \\
\leq\ & \sum_{z=2}^{n} \binom{n}{z} \left(\frac{1}{\alpha n}\right)^z \\
\leq\ & \sum_{z=2}^{n} n^z \left(\frac{1}{\alpha n}\right)^z \\
\leq\ & \sum_{z=2}^{n} \frac{1}{(\alpha)^z} \\
\leq\ & \frac{1}{\alpha(\alpha-1)}.
\end{aligned}$$

Now if we consider any pair cycle of length $z \geq 4$ then it requires z edges, z vertices and at most $z-1$ players. Then probability that implication graph has pair cycle is given by,

$$
\begin{aligned}
& \text{Prob}[IG\ contain\ pair\ cycles] \\
\leq\ & \sum_{z=4}^{n} \binom{n}{z-1} \left(\frac{1}{\alpha n}\right)^{z} \\
\leq\ & \sum_{z=2}^{n} n^{z-1} \left(\frac{1}{\alpha n}\right)^{z} \\
\leq\ & \frac{1}{n} \sum_{z=2}^{n} \frac{1}{(\alpha)^{z}} \\
\leq\ & O(\frac{1}{n}).
\end{aligned}
$$

The only problem with the above calculation is that we have assumed that edges occur independently, but we know that edges do not occur independently (recall the cake distribution given in Figure 2.2). Nevertheless, Edmonds and Pruhs were able to prove the results for pair path. Motivated from their result of Lemma 31 we made the following conjecture.

Conjecture 20. *The probability that for some player p, we have pair paths of length at least three from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$ and from $a_{\langle p,1\rangle}$ to $a_{\langle p,0\rangle}$ in the implication graph IG is at most $O(\frac{1}{n})$.*

Recall that pair cycle requires that at least one player's both semifinal pieces have to be present in it. To prove Conjecture 20, we started with the case where exactly one player's both semifinal pieces are present in the pair cycle. This case looks simple and our intuition said that the probability of having more than one player repeating in a pair cycle is less than the case when exactly one player repeats. More repeats will reduce the number of distinct players in a pair cycle but the number of edges will remain the same. We were able to prove the probability calculation for this case.

Lemma 21. *The probability that for some player p we have pair paths of length at least three from $a_{\langle p,0\rangle}$ to $a_{\langle p,1\rangle}$ and from $a_{\langle p,1\rangle}$ to $a_{\langle p,0\rangle}$ in the implication graph IG and except player p there is no common player involved in both the pair paths, is at most $O(\frac{1}{n})$.*

Proof. We will provide highlights of the proof. Consider a simple pair paths

$$P_0 = \left\langle a_{\langle p,0\rangle}, a_{\langle p_1,r_1\rangle}, \ldots a_{\left\langle p_{z_0-1}, r_{z_0-1}\right\rangle}, a_{\langle p,1\rangle} \right\rangle \text{ of length } z_0 \geq 3 \text{ and}$$
$$P_1 = \left\langle a_{\langle p,1\rangle}, a_{\langle q_1,r_1\rangle}, \ldots a_{\left\langle q_{z_1-1}, r_{z_1-1}\right\rangle}, a_{\langle p,0\rangle} \right\rangle \text{ of length } z_1 \geq 3.$$

Let V_0 be the vee with center $a_{\langle p,0\rangle}$ and bases $a_{\langle p_1,1-r_1\rangle}$ and $a_{\left\langle p_{z_0-1}, r_{z_0-1}\right\rangle}$, and V_1 be the vee with center $a_{\langle p,1\rangle}$ and bases $a_{\langle q_1,1-r_1\rangle}$ and $a_{\left\langle q_{z_1-1}, r_{z_1-1}\right\rangle}$. For $i \in [1, z_0-2]$, let $I_{0_i} \in IG$ be the event that semifinal pieces $a_{\langle p_i,r_i\rangle}$ and $a_{\langle p_{i+1},1-r_{i+1}\rangle}$ intersect. For $j \in [1, z_1-2]$, let $I_{1_j} \in IG$ be the event that semifinal pieces $a_{\langle q_j,r_j\rangle}$ and $a_{\langle q_{j+1},1-r_{j+1}\rangle}$ intersect. Since both the pair path P_0 and P_1 have all different players except player p, event I_{0_i} and I_{1_j} are independent for all possible values of i and j. Player p contributes to both the pair paths but for the probability computation it is counted only once. We can slightly modify the proof of Lemma 31 and get the desired result. Instead of considering set $\mathcal{V}$ of all possible vees, we consider the set $\mathcal{V}'$ of possible pairs (V_0, V_1) of vees where center piece of V_0 and V_1 is $a_{\langle p,0\rangle}$ and $a_{\langle p,1\rangle}$ respectively for some player p. ☐

Note that above lemma consider the case when length of both pair paths is at least three and only one player repeats. In general, more than one player can repeat in a pair cycle and pair path can be of length two in one of the direction or both. Recall that whenever we have pair path of length two, we get self-vee among semifinal pieces. We proved following lemma to handle the case when one of the pair path has length two.

Lemma 22. *Let $\alpha \geq 10$ be some sufficiently large constant. Each of n players has a partition* *of the unit interval* $[0, 1]$*, or cake, into* αn *disjoint candidate subintervals/pieces. Each player independently picks* $d' = 3d = 6$ *of her pieces uniformly at random, with replacement. Then there is an efficient method that, with probability at least* $\Omega(1 - \frac{1}{n})$*, chooses three of the* d' *pieces for each player and then narrow down two pieces for each player, so that same-player-vee graph build from these chosen pieces can be coloured by at most two colour.*

Proof. We will provide highlights of the proof. Same-player-vee graph will have path of length two starting with player p if one of the semifinal piece $a_{\langle p,r \rangle}$ of player p overlaps with both the semifinal pieces $a_{\langle p_1,0 \rangle}$ and $a_{\langle p_1,1 \rangle}$ of some player p_1 and one of the piece $a_{\langle p_1,r_1 \rangle}$ overlaps with both the semifinal pieces $a_{\langle p_2,0 \rangle}$ and $a_{\langle p_2,1 \rangle}$ of some player p_2. So piece $a_{\langle p,r \rangle}$ is not good for player p. We proved that probability of any two pieces (out of three chosen pieces) being bad for some player p is at most $O(\frac{1}{n})$. This is done by modifying the proof given for Lemma 16. So we can select with probability at least $\Omega(1 - \frac{1}{n})$ two pieces for each player such that same-player-vee graph does not have path of length two. □

Then to handle the case where more than one player can repeat in pair cycle, we tried many proof techniques and then found the following counterexample.

Counterexample:

Figure 4.2 shows the counterexample to Conjecture 20. Corresponding cake distribution is shown in Figure 4.3. It also shows the semifinal pieces received by the corresponding players. Note that once player 1 and player 2 select their leftmost piece as the semifinal piece, with constant probability we get the dashed square shown in Figure 4.2. The probability that both player 1 and

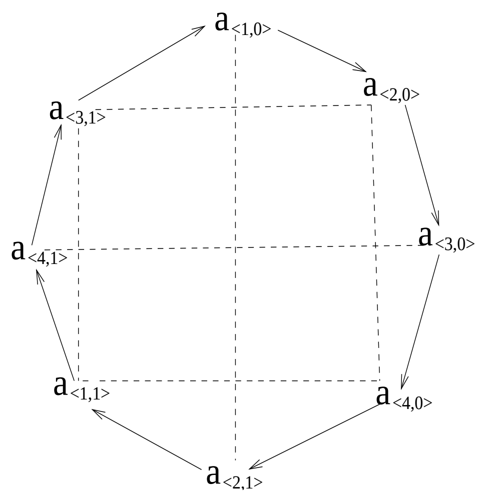

Figure 4.2: The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph.

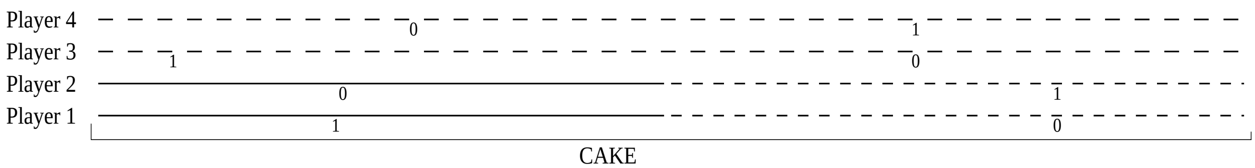

Figure 4.3: Cake distribution of 4 players with their selected semifinal pieces.

2 will select their leftmost piece as the semifinal piece is at least $\Omega(\frac{1}{n^2})$. Therefore, the probability that we get dashed square is at least $\Omega(\frac{1}{n^2})$. The remaining two dashed edges can occur with probability at least $\Omega(\frac{1}{n^2})$. So the probability that this particular counterexample exists is at least $\Omega(\frac{1}{n^4})$. Since we have at least $\Omega(n^4)$ possible choices for 4 players having cake distribution shown in Figure 4.3, probability that this sort of counterexample exists in the implication graph is at least $\Omega(1)$.

5 Summary and Future Directions

In this chapter we summarize our contributions and provide directions for future work by listing the open problems in the area of cake cutting.

Sgall and Woeginger [13] showed that every exact fair protocol (deterministic or randomized) has complexity $\Omega(n \log n)$ if every portion given to players is restricted to be a contiguous piece of the cake. Edmonds and Pruhs [4] extended the lower bound to apply even when the protocol need to guarantee only *approximate fairness* and not necessarily assign contiguous portions. They proved that every deterministic approximate fair protocol for cake cutting has complexity $\Omega(n \log n)$. They also proved that every randomized approximate fair protocol has complexity $\Omega(n \log n)$ if answers to queries asked by protocol are approximations to actual answers.

In [5], Edmonds and Pruhs gave a randomized approximate fair protocol of complexity $O(n)$ with $O(1)$ success probability. Their protocol need not assign contiguous portions to players. We have improved on this by giving a randomized approximate fair protocol with complexity $O(n)$ with *high* probability of success. The probability of success of our protocol is $O(1 - \frac{1}{poly(n)})$. An outline of our protocol follows.

- Independently run twice, the protocol by Edmonds and Pruhs upto the formation of *implication graph* and *same-player-vee graph*.

- Delete players that are *bad* for the run from the corresponding graphs.
- Each run is then completed by narrowing down to one final piece for every player in the run.
- Merge results from the two runs. This results into a situation where each point of the cake has at most $4 = 2 \times 2$ players (at most 2 players from each run) wanting it.

We also proved that it is unlikely that some player will be deleted from both the runs by proving that the probability of existence of a bad player is less than or equal to $O(\frac{1}{n})$. This implies that the probability that our protocol will not work is also $O(\frac{1}{n})$.

One can execute the above two-run protocol k times and select the first successful execution. This improves the probability of success because the probability that none of the k executions would work is less than or equal to $O(\frac{1}{n^k})$. For each of the k executions of the two-run protocol, no point in the cake is wanted by more than 4 players and since we select only one successful execution from k executions, the number of players sharing any point in the cake is at most 4. We get the same probability of success if we run Edmonds and Pruhs algorithm $k+1$ times instead of two times, but the number of players wanting any point of the cake could be as much as $2k+2$. As with the Edmonds and Pruhs protocol, our protocol may not assign contiguous pieces to players.

There are several lines of further inquiry. One could try to determine if linear complexity is obtainable for cake cutting if either exact fairness or contiguous portions were required for either constant or high probability of success.

5.1 Open Problems

The open problems in the area of cake cutting are as follows.

- A Linear randomized protocol which is fair or assigns contiguous pieces, with either constant or high probability of success.

- Lower bound on the probability of success for linear randomized protocols.

- Let α be some large constant. Each of the n players has a partition of the unit interval $[0,1]$, or cake, into αn disjoint candidate subintervals/pieces. Each player independently picks d of her pieces uniformly at random, with replacement. Each player comes one by one. Can we get an efficient method that, with either constant or high probability, picks one of the d subintervals for each player, so that every point on the cake is covered by at most $\theta(\frac{\log \log n}{\log d})$ pieces ? This is the generalization of Theorem 2 for cake cutting.

Bibliography

[1] Y. AZAR, (1998) On-line Load Balancing, *Online Algorithms - The State of the Art*, Springer, 178-195.

[2] Y. AZAR, A. BRODER, A. KARLIN, AND E. UPFAL, (2000) Balanced Allocations, SIAM Journal of Computing 29, 180–200.

[3] S.J. BRAMS AND A.D. TAYLOR (1996). *Fair Division – From cake cutting to dispute resolution*. Cambridge University Press, Cambridge.

[4] J. EDMONDS AND K. PRUHS, Cake cutting really isn't a piece of cake, *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2006)*.

[5] J. EDMONDS AND K. PRUHS, Balanced Allocations of Cake, *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'2006)*, 623-634.

[6] S. EVEN AND A. PAZ (1984). A note on cake cutting. *Discrete Applied Mathematics 7*, 285–296.

[7] 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.

[8] M. MITZENMACHER, A. RICHA, AND R. SITARAMAN 200 The power of two random choices: A survey of the techniques and results, *Handbook of Randomized Computing*, editors: P. Pardalos, S. Rajasekaran, and J. Rolim, Kluwer.

[9] N. JOHNSON AND S. KOTZ (1977). Urn Models and Their Application. *John Wiley and Sons*.

[10] 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.

[11] 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.

[12] J.M. ROBERTSON AND W.A. WEBB (1998). *Cake-cutting algorithms: Be fair if you can*. A.K. Peters Ltd.

[13] 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.

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

[15] 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.

A Proofs by Edmonds and Pruhs

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.

A.1 Final Piece Selection Algorithm

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

Lemma 23. *If there is a path in G from $a_{\langle p,r\rangle}$ to $a_{\langle q,s\rangle}$ then there must be a path from $a_{\langle q,1-s\rangle}$ to $a_{\langle p,1-r\rangle}$ in G.*

Lemma 24. *If both the pieces $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$ are reachable from a piece $a_{\langle p,r\rangle}$ in the implication graph G, then G has a pair path.*

Lemma 25. *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 $a_{\langle p,0\rangle}$ to player p. This iteration will force the assignment of at most one piece to any one player because by Lemma 24 there can not be a player q such that both $a_{\langle q,0\rangle}$ and $a_{\langle q,1\rangle}$ are reachable from $a_{\langle p,0\rangle}$. Similarly, if this same iteration forces player q to be assigned say to $a_{\langle q,0\rangle}$, then we need to prove that he has not already been assigned $a_{\langle q,1\rangle}$ during an earlier iteration. If assigning $a_{\langle p,0\rangle}$ forces $a_{\langle q,0\rangle}$, then there is a path from the one to the other. Hence, by Lemma 23, there is a path from $a_{\langle q,1\rangle}$ to $a_{\langle p,1\rangle}$. Hence, if $a_{\langle q,1\rangle}$ had been previously assigned, then player p would have been forced to $a_{\langle p,1\rangle}$ 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. $\square$

A.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 $a_{\langle p,r\rangle}$ and two *base* pieces $a_{\langle q,s\rangle}$ and $a_{\langle q',s'\rangle}$, with the property that the center piece intersects both of the base two pieces.

Lemma 26. *Assume that n players have partitioned their cake into αn pieces each. Let $\ell_{p,i}$ be the number of pieces of the other players that overlap with piece i of player p. Then for any player p, $\sum_{i=1}^{\alpha n} \ell_{p,i} \leq 2\alpha n^2$.*

Lemma 27. *The probability that semifinal piece $a_{\langle p,r\rangle}$ overlaps with semifinal piece $a_{\langle q,s\rangle}$ is at most $\frac{2d^2}{\alpha n}$.*

Lemma 28. *The expected number of vee's in G is at most $\frac{16d^3}{\alpha^2}n$.*

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

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

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

$$
\begin{aligned}
E[x_{\langle p,r\rangle}] &= \sum_{i=1}^{\alpha n} \mathrm{Prob}[R_{\langle p,i,r\rangle}] \cdot E[x_{\langle p,r\rangle} \mid R_{\langle p,i,r\rangle}] \leq \sum_{i=1}^{\alpha n} \left(\frac{d}{\alpha n}\right)\left(\frac{i-1}{\alpha n}\right)^{d-1} \cdot 2\ell^2_{\langle p,i\rangle}\left(\frac{d}{\alpha n}\right)^2 \\
&\leq \left(\frac{2d^3}{(\alpha n)^{d+2}}\right) \cdot \sum_{i=1}^{\alpha n} i^{d-1}\ell^2_{\langle p,i\rangle}
\end{aligned}
$$

Lemma 26 bounds that $\sum_{i=1}^{\alpha n} \ell_{\langle p,i\rangle} \leq 2\alpha n^2 = M$. The next lemma then bounds $\sum_{i=1}^{m} i^{d-1}\ell^2_{\langle p,i\rangle} \leq m^{d-2}M^2$.

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

By linearity of expectation, the expected number of vees over all is $\sum_{p=1}^{n}\sum_{r=0}^{1} E[x_{\langle p,r\rangle}] \leq 2n \cdot \frac{8d^3}{\alpha^2}$. □

Lemma 29. *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 $M = \sum_{i=1}^{m} \ell_i = \sum_{i=1}^{m} i s_i$. Then let $t_i = i s_i$ so that $M = \sum_{i=1}^{m} t_i$. Now using basic algebra we conclude that

$$\begin{aligned}\sum_{i=1}^{m} i^{d-1}\ell_i^2 &= \sum_{i=1}^{m} i^{d-1}\left(\sum_{j=i}^{m} s_j\right)^2 = \sum_{i=1}^{m} i^{d-1}\sum_{j=i}^{m}\sum_{k=i}^{m} s_j s_k = \sum_{j=1}^{m}\sum_{k=1}^{m} s_j s_k \sum_{i=1}^{\min(j,k)} i^{d-1} \\ &\leq \sum_{j=1}^{m}\sum_{k=1}^{m}\frac{t_j t_k}{jk}\min(j,k)^d \leq m^{d-2}\sum_{j=1}^{m}\sum_{k=1}^{m} t_j t_k = m^{d-2}\left(\sum_{j=1}^{m} t_j\right)^2 = m^{d-2}M^2\end{aligned}$$

□

A.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 of more. Recall that if the semifinal pieces $a_{\langle p,r\rangle}$ and $a_{\langle q,s\rangle}$ intersect, then there is an directed edge in the implication graph G from $a_{\langle p,r\rangle}$ to $a_{\langle q,1-s\rangle}$ and from $a_{\langle q,s\rangle}$ to $a_{\langle p,1-r\rangle}$ and that a *pair path* is a directed path between the two semifinal pieces for the same player, i.e. from some $a_{\langle p,r\rangle}$ to $a_{\langle p,1-r\rangle}$. Note that we need to only bound the probability of *simple* pair paths (pair paths where all the players are different except starting and ending player in the path). The next lemma is best understood by studying Figure A.1.

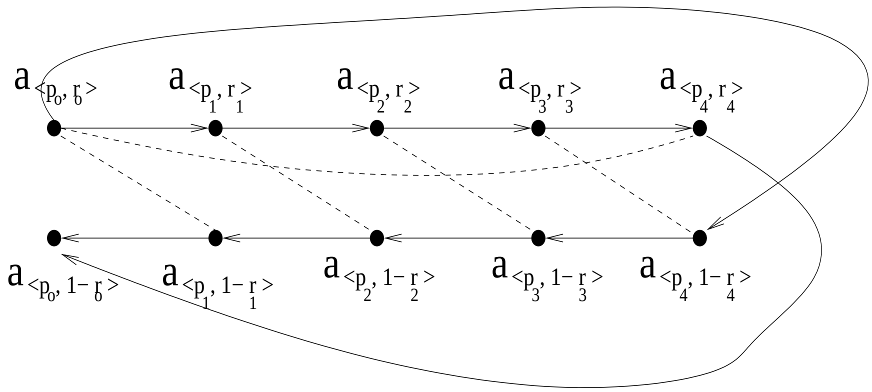

Figure A.1: The dotted edges are between semifinal pieces that overlap. The solid directed edges are the resulting edges in the implication graph.

Lemma 30. *Consider a simple pair path* $P = \left\langle a_{\langle p_0,r_0\rangle}, a_{\langle p_1,r_1\rangle}, \ldots a_{\langle p_{k-1},r_{k-1}\rangle}, a_{\langle p_0,1-r_0\rangle}\right\rangle$ *of length* $k \geq 3$*. Let* V *be the vee with center* $a_{\langle p_0,r_0\rangle}$ *and bases* $a_{\langle p_1,1-r_1\rangle}$ *and* $a_{\langle p_{k-1},r_{k-1}\rangle}$*. For* $i \in [1, k-2]$*, let* $I_i \in G$ *be the event that semifinal pieces* $a_{\langle p_i,r_i\rangle}$ *and* $a_{\langle p_{i+1},1-r_{i+1}\rangle}$ *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. Assume $P \in G$. The edges from $a_{\langle p_0,r_0\rangle}$ to $a_{\langle p_1,r_1\rangle}$ and from $a_{\langle p_{k-1},r_{k-1}\rangle}$ to $a_{\langle p_0,1-r_0\rangle}$ mean that $a_{\langle p_0,r_0\rangle}$ intersect with both $a_{\langle p_1,1-r_1\rangle}$ and $a_{\langle p_{k-1},r_{k-1}\rangle}$. Hence, the vee V occurs. The edge from $a_{\langle p_i,r_i\rangle}$ to $a_{\langle p_{i+1},r_{i+1}\rangle}$ means that $a_{\langle p_i,r_i\rangle}$ and $a_{\langle p_{i+1},1-r_{i+1}\rangle}$ 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 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 $A_{\langle p,0\rangle}$ and the selection of p's semifinal piece $a_{\langle p,0\rangle}$ within this bracket is independent of this process for his other semifinal piece $a_{\langle p,1\rangle}$. $\square$

Lemma 31. *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 $\mathcal{V}$ be the set of all 3-tuples representing all possible vee's in G and for $V \in \mathcal{V}$ 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

$$\sum_{k=3}^{n} \sum_{V\in\mathcal{V}} \sum_{P\in\mathcal{P}_k(V)} \mathrm{Prob}[P \in G] \tag{A.1}$$

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

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

$$\tag{A.4}$$

$$\leq \sum_{k=3}^{n} \sum_{V\in\mathcal{V}} \mathrm{Prob}[V \in G] \left(\binom{2n}{k-3}(k-3)!\right) \left(\frac{2d^2}{\alpha n}\right)^{k-2} \tag{A.5}$$

$$\leq \sum_{k=3}^{n} (2n)^{k-3} \left(\frac{2d^2}{\alpha n}\right)^{k-2} \sum_{V\in\mathcal{V}} \mathrm{Prob}[V \in G] \tag{A.6}$$

$$\leq \sum_{k=3}^{n} (2n)^{k-3} \left(\frac{2d^2}{\alpha n}\right)^{k-2} \left(\frac{16d^3}{\alpha^2} n\right) \tag{A.7}$$

$$\leq \frac{8d^3}{\alpha^2} \sum_{k=3}^{n} \left(\frac{4d^2}{\alpha}\right)^{k-2} \leq \frac{8d^3}{\alpha^2} \left(\frac{4d^2}{\alpha}\right) \left(\frac{1}{1-4d^2/\alpha}\right) = \frac{32d^5}{\alpha^2(\alpha-4d^2)} \tag{A.8}$$

The inequality in line A.2 follows from Lemma 30 and line A.3 from Lemma 27. The inequality in line A.5 holds since there are $k-3$ pieces in P that are not part of the vee V. The inequality in line A.7 follows from Lemma 28. □

A.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 32. *The probability that the same-player-vee graph is not $w = 2$ colourable is at most* $\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}$.

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 $a_{\langle p,0\rangle}$ or $a_{\langle p,1\rangle}$, overlap with both of player q's two semifinal pieces, namely $a_{\langle q,0\rangle}$ and $a_{\langle p,1\rangle}$. Hence, a path of length 2 consists of semi-final pieces $a_{\langle p_1,r_1\rangle}$, $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_2,1-r_2\rangle}$, $a_{\langle p_3,0\rangle}$, and $a_{\langle p_3,1\rangle}$ for three players p_1, p_2, and p_3, where both $a_{\langle p_2,r_2\rangle}$ and $a_{\langle p_2,1-r_2\rangle}$ overlap with $a_{\langle p_1,r_1\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$. We will consider the probability of such paths starting backwards.

Lemma 33. *Suppose we are considering a set of $\widehat{\ell}$ candidate pieces for the semi-final pieces $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$. The probability that some player gets both of his semi final pieces from this set is at most* $\min((\frac{d\widehat{\ell}}{\alpha n})^2, 1)$.

Consider some candidate piece $c_{\langle p_1,i\rangle}$ that potentially might be $a_{\langle p_1,r_1\rangle}$. Let $\ell_{\langle p_1,i\rangle}$ denote the number of other candidate pieces of overlapping it. Consider some player p_2. Let $c_{\langle p_2,j_l\rangle}, c_{\langle p_2,j_l+1\rangle}$,

$\ldots, c_{\langle p_2, j_r\rangle}$ be the candidate pieces of player p_2 that overlap with piece $c_{\langle p_1,i\rangle}$. Let $\ell_{\langle p_2,j\rangle}$ denote the number of other candidate pieces overlapping $c_{\langle p_2,j\rangle}$. Consider some player p_3. Define $\ell_{\langle p_2,j,p_3\rangle}$ to be the number of player p_3's candidate pieces that overlap $c_{\langle p_2,j\rangle}$. Note that if $\ell_{\langle p_2,j,p_3\rangle} = 1$, then it is impossible to have both of player p_3's semi-final pieces overlap $c_{\langle p_2,j\rangle}$. Hence, we can ignore player p_3 when considering $c_{\langle p_2,j\rangle}$ as being $a_{\langle p_2,r_2\rangle}$ (we need $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ both to overlap with $a_{\langle p_2,r_2\rangle}$, giving $\ell_{\langle p_2,j,p_3\rangle} \geq 2$). Hence, define $\widehat{\ell}_{\langle p_2,j,p_3\rangle}$ to be $\ell_{\langle p_2,j,p_3\rangle}$ if $\ell_{\langle p_2,j,p_3\rangle} \geq 2$ and zero otherwise. Define $\widehat{\ell}_{\langle p_2,j\rangle} = \sum_q \widehat{\ell}_{\langle p_2,j,q\rangle}$. Note this is the number of pieces that overlap $c_{\langle p_2,j\rangle}$ excluding those pieces whose player only has one piece overlapping $c_{\langle p_2,j\rangle}$.

Lemma 34. *Then* $\sum_{j=j_l+1}^{j_r-1} \widehat{\ell}_{\langle p_2,j\rangle} \leq 2\ell_{\langle p_1,i\rangle}$.

Lemma 35. *Consider a candidate piece $c_{\langle p_1,i\rangle}$ such that there are $\ell_{\langle p_1,i\rangle}$ other candidate pieces overlapping it and some other player p_2. The probability that there are semi-final pieces $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_3,0\rangle}$, and $a_{\langle p_3,1\rangle}$ for some player p_3, where $a_{\langle p_2,r_2\rangle}$ overlaps with $c_{\langle p_1,i\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$ is at most* $\frac{4d}{\alpha n} \cdot \left[\frac{d\ell_{\langle p_1,i\rangle}}{\alpha n} + 1\right]$.

Proof. Consider a candidate piece $c_{\langle p_2,j\rangle}$ that overlaps with $c_{\langle p_1,i\rangle}$. The probability that candidate piece $c_{\langle p_2,j\rangle}$ is a semi-final piece for player p_2 is at most $\frac{2d}{\alpha n}$. By Lemma 33, the probability that there are semi-final pieces $a_{\langle p_3,0\rangle}$, and $a_{\langle p_3,1\rangle}$ for some player p_3 which both overlap with $c_{\langle p_2,j\rangle}$ is at most $\min((\frac{d\widehat{\ell}_{\langle p_2,j\rangle}}{\alpha n})^2, 1)$. It follows that the required probability is at most

$$\sum_{j=j_l}^{j_r} \frac{2d}{\alpha n} \cdot \min\left(\left(\frac{d\widehat{\ell}_{\langle p_2,j\rangle}}{\alpha n}\right)^2, 1\right) \leq \frac{2d}{\alpha n} \cdot \left[1 + \left[\sum_{j=j_l+1}^{j_r-1} \min\left(\left(\frac{d\widehat{\ell}_{\langle p_2,j\rangle}}{\alpha n}\right)^2, 1\right)\right] + 1\right].$$

By Lemma 34, $\sum_{j=j_l+1}^{j_r-1} \widehat{\ell}_{\langle p_2,j\rangle} \leq 2\ell_{\langle p_1,i\rangle}$. Hence, because of the quadratics in the sum, our sum is maximized by having a few $\widehat{\ell}_{\langle p_2,j\rangle}$ as big as possible. But because of the $\min$, there is no reason to make a $\widehat{\ell}_{\langle p_2,j\rangle}$ bigger than $\frac{\alpha n}{d}$. Hence, the sum is maximized by setting $\frac{2d\ell_{\langle p_1,i\rangle}}{\alpha n}$ of the values $\widehat{\ell}_{\langle p_2,j\rangle}$ to $\frac{\alpha n}{d}$ and the rest to zero. This gives the result

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

□

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

Lemma 36. *Consider a candidate piece $c_{\langle p_1,i\rangle}$ such that there are $\ell_{\langle p_1,i\rangle}$ other candidate pieces overlapping it. The probability that there are semi-final pieces $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_2,1-r_2\rangle}$, $a_{\langle p_3,0\rangle}$, and $a_{\langle p_3,1\rangle}$ for two players p_2 and p_3, where both $a_{\langle p_2,r_2\rangle}$ and $a_{\langle p_2,1-r_2\rangle}$ overlaps with $c_{\langle p_1,i\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$ is at most $\frac{4d^2\ell^2_{\langle p_1,i\rangle}}{(\alpha n)^3} \cdot \left[1 + \frac{\alpha n}{d\ell_{\langle p_1,i\rangle}}\right]$.*

Proof. The probability that a particular candidate piece $c_{\langle p_2,j\rangle}$ is player p_2's semi-final piece $a_{\langle p_2,1-r_2\rangle}$ is at most $\frac{d}{an}$. Denote the number of player p_2's candidate pieces $c_{\langle p_2,j_l\rangle}, c_{\langle p_2,j_l+1\rangle}, \ldots, c_{\langle p_2,j_r\rangle}$ that overlap with piece $c_{\langle p_1,i\rangle}$ to be $q_{p_2} = j_r - j_l + 1$. Because these all overlap with $c_{\langle p_1,i\rangle}$, we have that $\sum_{p_2} q_{p_2} = \ell_{\langle p_1,i\rangle}$. Using Lemma 35, we get that the required probability is at most

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

□

We will now add the requirement that $c_{\langle p_1,i\rangle}$ 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 37. *The probability that there are semi-final pieces $a_{\langle p_1,r_1\rangle}$, $a_{\langle p_2,r_2\rangle}$, $a_{\langle p_2,1-r_2\rangle}$, $a_{\langle p_3,0\rangle}$, and $a_{\langle p_3,1\rangle}$ for three players p_1, p_2, and p_3, where both $a_{\langle p_2,r_2\rangle}$ and $a_{\langle p_2,1-r_2\rangle}$ overlap with $a_{\langle p_1,r_1\rangle}$, and both $a_{\langle p_3,0\rangle}$ and $a_{\langle p_3,1\rangle}$ overlap with $a_{\langle p_2,r_2\rangle}$ is at most $\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}$.*

Proof. As in the proof of Lemma 28, let $R_{\langle p,i,r\rangle}$ be the event that the candidate $c_{\langle p,i\rangle}$ is selected to be the semifinal piece $a_{\langle p,r\rangle}$. Recall that $\mathrm{Prob}[R_{\langle p,i,r\rangle}] = d \cdot (\frac{1}{\alpha n}) \cdot (\frac{i-1}{\alpha n})^{d-1}$. There are n choices for player p_1. Thus by Lemma 36, our desired probability is at most

$$\begin{aligned}
& n\left(\sum_{i=1}^{\alpha n} \frac{d}{\alpha n}\left(\frac{i-1}{\alpha n}\right)^{d-1} \frac{4d^2\ell^2_{\langle p_1,i\rangle}}{(\alpha n)^3} \cdot \left[1 + \frac{\alpha n}{d\ell_{\langle p_1,i\rangle}}\right]\right) \\
\leq\ & n\left(\frac{4d^3}{(\alpha n)^{d+3}} \sum_{i=1}^{\alpha n} \ell^2_{\langle p_1,i\rangle}(i-1)^{d-1} + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{\langle p_1,i\rangle}(i-1)^{d-1}\right) \\
\leq\ & n\left(\frac{4d^3}{(\alpha n)^{d+3}}(\alpha n)^{d-2}(2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}} \sum_{i=1}^{\alpha n} \ell_{\langle p_1,i\rangle}(i-1)^{d-1}\right) \\
\leq\ & n\left(\frac{4d^3}{(\alpha n)^{d+3}}(\alpha n)^{d-2}(2\alpha n^2)^2 + \frac{4d^2}{(\alpha n)^{d+2}}(\alpha n)^d \left(\frac{2\alpha n^2}{\alpha n}\right)\right) = \frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2}
\end{aligned}$$

The second inequality follows by Lemma 29. The third inequality follows from noting that, given that the $\ell_{\langle p_1,i\rangle}$'s are non increasing, the sum is obviously maximized if each $\ell_{\langle p_1,i\rangle}$ is equal. That is, each $\ell_{\langle p_1,i\rangle} = \frac{2\alpha n^2}{\alpha n}$. ☐

A.5 Computing the Probability of Failure

The probability that the total same-player-vee graph is not 2-colourable 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 $\frac{32d^3}{\alpha^2(\alpha-d^2)}$. 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 α to be sufficiently large, one can make this probability arbitrarily small. Hence, the probability that our caking cutting algorithm is not at least 2α-fair is at most $\frac{16d^3}{\alpha^3} + \frac{8d^2}{\alpha^2} + \frac{32d^3}{\alpha^2(\alpha - d^2)}$.