Design and Analysis of Algorithms | CSCI 5454
Crowd Source Lecture 20 | 13 April 2011

Aditya Sawhney and Vikrant Malhotra

1 Fair Division

Fair division is the problem of dividing one or several goods amongst two or more players in a way that satisfies a suitable fairness criterion. Fair division problems, such as chore division, inheritance allocation, and room selection have been extensively studied in philosophy, political science, economics, and mathematics for a long time, but are also relevant to computer science. These problems arise often in everyday computing when different users compete for the same resources. Typical examples of such problems are: job scheduling; sharing the CPU time of a multiprocessor machine; sharing the bandwidth of network connection; etc. The resource to be shared can be viewed as a "cake", and the problem of sharing such as resource is called ***cake-cutting***. The theory of fair division dates back to 1940's when it was originally formulated by Polish mathematician, Steinhaus [1].

1.1 Motivating problem

Consider a large size pizza which is half pepperoni - half mushroom.

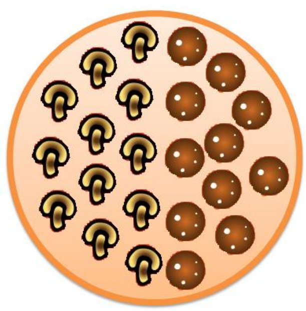


Figure 1.Pizza with different toppings

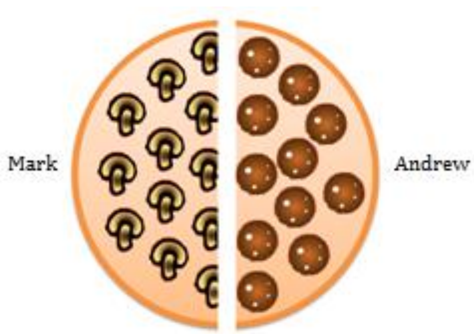
The pizza needs to be fairly divided between two people - Mark and Andrew, wherein a "fair" share for both would be half the value of the pizza according to *their own value system*.

Suppose that the value system of Mark and Andrew is as follows:

- Mark is a vegetarian so for him only the mushroom half is valuable and rest is worthless
- Andrew equally likes mushroom and pepperoni so for him whole of the pizza is valuable

Problem: Find a cut such that both Mark and Andrew feel that they have received a fair share.

Solution: One such possible cut which yields in a fair share division is,



Mark – gets a fair share as he gets half the pizza and all the mushrooms (that's what he is only interested in)

Andrew – gets a fair share as he gets half the pizza and all the pepperoni (as he is fine with anything)

1.2 Terminology

In formal terms, a cake-cutting/fair-division problem is represented as follows:

Let N = $\{1, \dots, n\}$ be set of *players* (or *agents*, *individuals*) who need to share a *cake* (or *goods*, *resources*, *items*, *objects*).

The *cake C* is represented by the unit interval [0, 1] and these players need to divide the cake amongst themselves by means of a series of cuts.

A *piece* is a finite union of subintervals of the full cake. These subintervals are not allowed to overlap (so pieces cannot be shared) and we are only interested in complete allocations, where every piece of cake is allocated to someone.

Each player $i \in$ N has a *utility* function u_i (or *valuation* function) to model their preferences and is used to determine how the player values a piece of cake.

An *allocation* A (or *agreement*) is a mapping of players to *pieces of cake* which results in a *utility vector* $\langle u_1(A), \dots, u_n(A)\rangle$.

A *fair division criterion* is a set of rules which define the conditions of fairness.

And the goal of the cake-cutting problem is to find an allocation where all the players feel that they have received a fair share based on the given fair division criterion.

The two most prominent fairness criterions are:

1. **Proportional:** Under this condition a division is said to be 'fair' when each of the n players feel (in their opinion) that they have received at least 1/n of the total value of C.

   Definition: An allocation is proportional if $u_i(A(i)) \geq \frac{1}{n} \cdot \hat{u}_i$ for every player $i \in$ N, where $\hat{u}_i$ is the utility given to the full cake by player i.

   Illustration: Considering the same pizza example, it can be deduced that according to Mark a fair share would be at least half portion of the mushroom half of the pizza (which is actually 1/4th of the pizza) and according to Andrew a fair share would be at least half portion of the whole pizza (as depicted below).

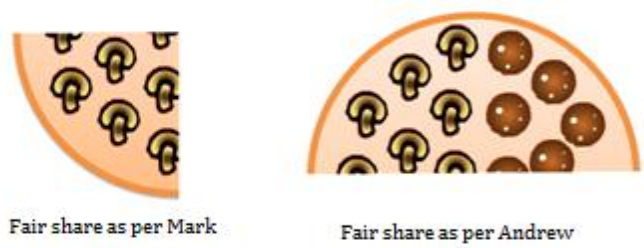


Figure 2.Proportional fair share as per Mark and Andrew

2. **Envy-free:** Under this condition a division is said to be 'fair' when no player feels that another player has a strictly larger piece.

   Definition: An allocation is envy-free if $u_i(A(i)) \geq u_i(A(j))$ for every pair of players $i, j \in$ N.

Also, it can be observed that envy-free has stricter fairness criteria, therefore, problems with envy-free criteria are harder to solve as compared to proportional problems.

A **cake-cutting algorithm** is a systematic procedure for solving a fair division problem. It must exhibit following properties and assumptions:

1. *Decisive* – each player should end up with a fair share.
2. *Internal* – division should be performed by players themselves (maybe using a mediator) as only the players really know how they value the pieces.
3. *Rationality* – each player should be cooperative and behave in a rational manner; meaning that they don't connive with other players and do not make emotional decisions.
4. *Privacy* – players don't know about other player's value system.

From **computational point of view**, we want to *minimize the number of cuts* needed, since this leads to a smaller number of computational steps performed by the algorithm. Intuitively, the number of cuts required in a solution can be used as a measure for the running time of an algorithm. The following sections cover a few of the most efficient fair division algorithms which aim at minimizing the number of cuts required for a fair division.

2 Cut and Choose Algorithm (n=2)

The origin of Cut & Choose method dates back to ancient Greek history and is still the most popular method for division between two players.

CUT-AND-CHOOSE ALGORITHM (2 players)

Step 1. One player *cuts* the cake in 2 pieces (which he/she considers to be of equal value).

Step 2. The other player *chooses* one of the pieces (the piece he/she prefers).

Note that the remarks in bracket are recommended winning strategies and not actual steps of the algorithm.

2.1 Illustration

Case 1: Mark is the cutter and Andrew is the chooser

Strategy

-Mark will cut the pizza in such a fashion that both pieces have equal value according to him (i.e. equal amount of mushrooms) as he doesn't know which piece Andrew would pick. This is also called risk-averse phenomena.

- Andrew will choose a piece which is at least half the value according to him (i.e. half the pizza).

Share analysis

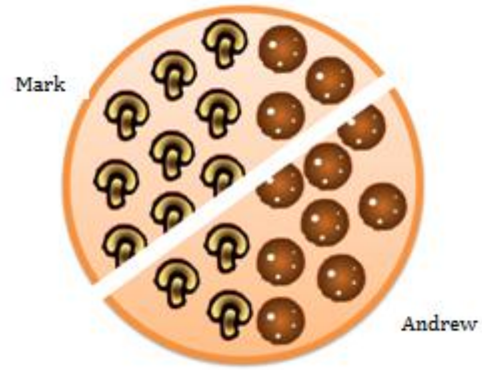
- Mark gets ½ portion of mushroom half which according to him is 50% the value of pizza.

- Andrew gets ½ the pizza which according to him is 50% of the value of pizza.

Case 2: Andrew is the cutter and Mark is the chooser

Strategy

- Andrew will cut the cake in equal halves in terms of portion size.
- Mark will choose the piece which has more mushrooms



Share analysis

- Mark gets 2/3rd portion of the mushroom half which according to him is 66.6% of the value of pizza.
- Andrew gets ½ the pizza which according to him is 50% of the value of pizza.

2.2 Properties

The cut and choose method satisfies two important properties:

- **Proportionality**

  Proof: Let A and B be the two players trying to share a cake and A be the cutter. No matter which piece B selects, A is going to get exactly ½ (by his measure) as he will always cut the cake in 2 equal pieces due to risk-averse phenomena. As for B, when it's time for him to choose, he sees two pieces, one worth W and the other worth 1-W (by his measure). It is guaranteed that either W ≥ ½ or 1-W ≥ ½ otherwise W + 1-W < 1 which is not possible. Therefore, B gets at least ½ by his measure. Thus, both A and B get at least ½ the value of cake by their own measure.

- **Envy-freeness**

  Proof: Player A will not be envy with any selection of B because as per his measure both the pieces are of equal value. As for B, he picks a piece which as per his measure is better than the other piece so he won't envy A getting the other piece. Thus, both A and B won't think that the other player gets a strictly larger piece.

Notice that in this algorithm the divider *always* comes out with a fair share while the chooser *sometimes* comes out with more than a fair share. So, to be really fair, the chooser should be picked using a fair coin.

3 Divide and Conquer Algorithm (n>2)

Divide and conquer method was proposed by Even and Paz [6] in 1984 and is the most efficient general case *proportional* cake-cutting algorithm known for more than two players.

DIVIDE-AND-CONQUER ALGORITHM (n players)

Step 1. Ask each player to indicate a mark which cuts the cake in $\left\lfloor\frac{n}{2}\right\rfloor : \left\lceil\frac{n}{2}\right\rceil$ ratio.

Step 2. Associate the part to the left of the $\left\lfloor\frac{n}{2}\right\rfloor th$ mark with the players who made the leftmost $\left\lfloor\frac{n}{2}\right\rfloor$ marks (group 1), and the rest with the others (group 2).

Step 3. Recursively, apply the same procedure to each of the two groups, until only a single player is left.

The pseudo-code for the algorithm is as follows:

```
DIVIDE_CONQUER(G)
    playerCount = size(G)
    if playerCount > 1:
        marks = requestMarks(G)                       Step 1
        cut = determineCut(marks)
        G1 = createGroup(G, 0, cut)                    Step 2
        G2 = createGroup(G, cut, totalPlayers-1)
        DIVIDE_CONQUER(G1)                             Step 3
        DIVIDE_CONQUER(G2)
```

3.1 Illustration

Consider n = 5, then in step (1) each player is asked to indicate a point x on the cake such that [0, x] has value 2/5 and [x, 1] has value 3/5. Then we cut the cake at the 2nd mark and associate the 2 players who made the 2 leftmost marks with the left part of the cake, and the remaining 3 players with the right part of the cake. This kind of local procedure is iterated for each part, until we are down to the level of individual pieces for every player.

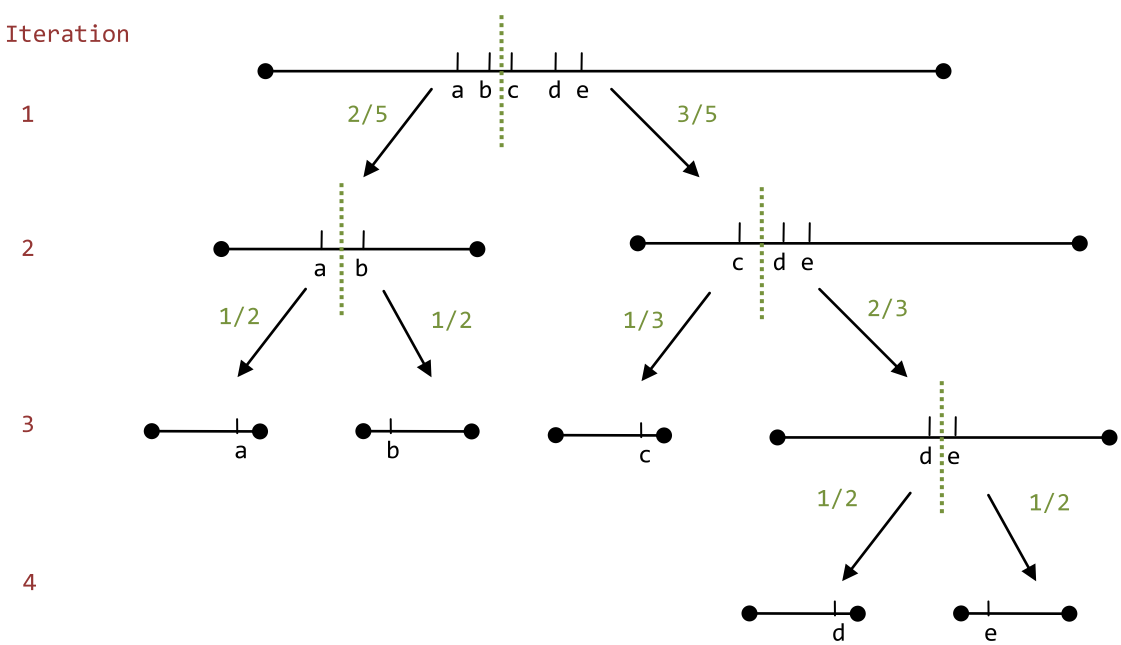


Figure 3.Illustration of Divide and Conquer for n=5

3.2 Properties

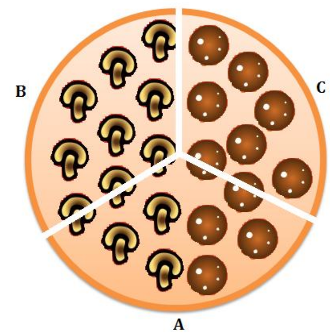
The divide and conquer method exhibits following properties:

- **Proportionality**

  Observing the example above it can be easily seen that the algorithm exhibits proportionality property as all the players are guaranteed to get at least 1/n of the cake.

- **NOT envy-free**

  Since the algorithm restricts at least one player to receive exactly 1/n of the cake, it is bound to be not envy-free because some players may envy others for receiving what they perceive to be more valuable.

  For example, consider a 3 player case where A is vegetarian and B & C are fine with anything. Now, say after step (1) A is part of group 1 and B & C are part of group 2. In this scenario, A is forced to take 1/3 portion of cake. Now, when B and C were to divide the cake amongst themselves as shown then A will perceive B's pieces to be more in value than his piece thereby making him envious.

3.3 Analysis

The number of cuts required in the above algorithm can be represented with following recurrence expression:

$$T(n) = n + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right)$$

As for given number of players (n) there are n number of marks (interpreted as cuts) corresponding to each player and the problem is broken down into two sub-problems of sizes $\left\lfloor \frac{n}{2} \right\rfloor$ and $\left\lceil \frac{n}{2} \right\rceil$ respectively. The recurrence expression roughly evaluates to $\boldsymbol{O(n \log_2 n)}$ which is thus the running time of the algorithm. Also, notice that the recurrence relation is the same as the one for merge-sort and quick-sort.

4 An envy free approach based on Sperner's Lemma [5]

4.1 Sperner's Lemma [14]

So far we have seen algorithms which are proportional and do not provide an envy free solution. The approach of Divide and Conquer, although gives the lower bound of $O(n \, log \, n)$, does not provide a complete envy-free solution. Procedures proposed by Selfridge and Conway [9], and Stormquist [9] are envy free but present the limitation on the number of players they deal with. Lately a procedure presented by Brams and Taylor [12] is an envy free procedure and extends to an arbitrary number of players. However, the number of cuts produced by this procedure becomes arbitrarily large and procedure itself becomes cumbersome because of the number of steps involved.

This turns us to look for other envy free approaches to cake cutting. An approach presented by Forest Simmons is based upon a simple combinatorial lemma proposed by Sperner in 1928. This approach presents an "approximate" envy free cake division and produces minimal number of cuts. In addition to proving the existence of an envy free cake division, it also presents a constructive way to find a solution. In the following we present this approach.

Some Definitions

1. ***n-simplex:*** An n-dimensional figure consisting of $n+1$ points such that each point is connected to every other point. It is like a mesh network in n-dimensional space. For instance, 0-simplex is a point, 1-simplex is a line, 2-simplex is a triangle, 3-simplex is a tetrahedron and so on. Thus, it can be considered to be an n-dimensional tetrahedron—a convex hull of $n+1$ affinely independent[1] points in $\mathbf{R}^m$, for $m > n$.
2. ***k-face*:** A k-face of an n-simplex is a k-simplex formed by the span of any subset of $k+1$ vertices.
3. ***Triangulation:*** A triangulation of an n-simplex S is a collection of distinct smaller n-simplices such that their union is S and have the property that any two of them either intersect in a face common to both, or not at all.
4. ***Facet:*** Any face of an n-simplex S, consists of first n of the $n+1$ vertices is called a facet. For instance, facets of a line are its end-points, facets of a triangle are its sides and facets of a tetrahedron are its triangular faces.

Sperner Labeling*:* Given an n-simplex S. We first triangulate it into smaller n-simplices, called *elementary simplices*, and then label it. The labeling obeys the following rule: We label each vertex of S with a distinct number in 1, 2, ….. n, and any other vertex in the interior of a k-face is labeled with one of the vertex numbers that span that face. For instance, in the following figure the Sperner labeling of a tetrahedron is shown in the figure below.

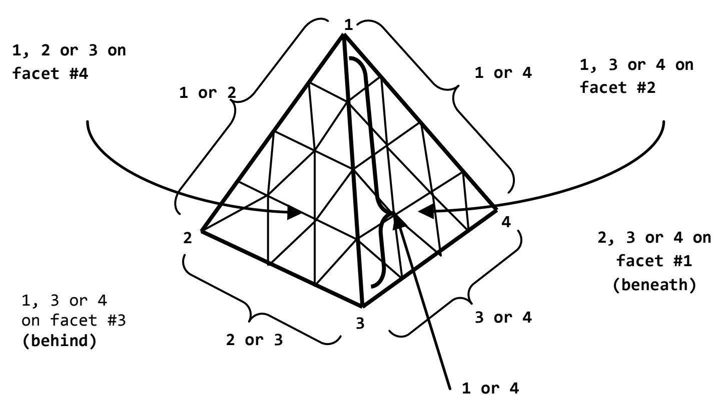


Figure 4. Sperner labeling on a triangulated tetrahedron

[1]Let $u_0, \dots, u_k \in R^n$. Then $x = \sum_{i=0}^{k} \lambda_i u_i$, $\lambda_i \in R$, is called an affine combination of u_i if $\sum_{i=0}^{k} \lambda_i = 1$
The points u_i are affinely independent if for any $x = \sum_{i=0}^{k} \lambda_i u_i$ and $y = \sum_{i=0}^{k} \mu_i u_i$ we have $x = y$ iff $\lambda_i = u_i$

From the above figure we can see that each of the vertices, lying on the edges of the tetrahedron, take one of the two numbers as are taken by one of the two end points of the corresponding edge; each of the vertices, on the interior of a triangular face, take one of the 3 numbers as taken by one of the vertices of the corresponding triangular face; and so on. Hence, it follows Sperner's labeling.

Sperner's Lemma: *Any Sperner-labelled triangulation of a n-simplex must contain an odd number of fully labeled elementary n-simplices. In particular, there is at least one.*

Where an elementary simplex of a triangulation is fully labeled if its vertices are labeled with distinct labels in 1, 2, 3,…,n.

Several approaches have been followed in proving Sperner's Lemma. Many of these are based upon *parity arguments* but are non-constructive. Here we discuss about a constructive approach, which along with proving the argument, helps get to a solution, i.e. to the fully labeled elementary n-simplices.

Proof: The proof proceeds by induction argument. First we analyze its validity for the case with $n = 1$. This is shown in the figure below. We can easily verify that there are an odd number of switches from 1 to 2 and hence an odd number of (1, 2)-elementary simplices. We then assume that the theorem is true all the way up to $n - 1$. If we then show that the theorem is true for a triangulated, Sperner-labeled n-simplex with (n+1) points, we are done!!

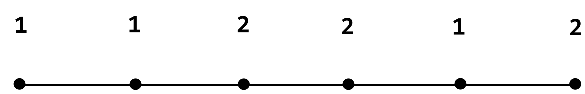


Figure 5. Sperner labeling for n=1

For the sake of simplicity, we take the case with $n = 2$, and move along with an arbitrary n. For n=2, the simplex S has a triangular shape. The triangulated simplex is shown in the figure below:

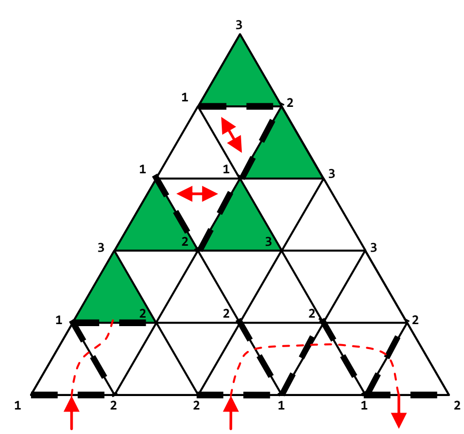


Figure 6. A "house" with fully labeled "rooms" (green) and their reachable paths

We can think of the simplex S as a “house” with elementary simplices—the triangles—as “rooms”. Also, we can assume the *facets* of “rooms” as “doors” but with a condition. A facet of an elementary n-simplex can be a door if it contains the first n of the $n + 1$ labels. So, for n=2, the triangular rooms will have (1, 2)-edges as doors. Similarly, for $n = 3$, the tetrahedral rooms will have (1, 2, 3)-facets as doors. These doors can be present on the facets or in the interior of the simplex S.

We then argue that the number of doors on the boundary of the simplex-S is odd. This is because the doors can only be present on the facet numbered “n”, i.e. the one containing the first n of the $n + 1$ labels. Since, we have already assumed that the lemma is true for (n-1)-simplex (in which case it is a “line”), there will be an odd number of fully labeled elementary (n-1)-simplices (doors) on that facet.

These boundary doors can be used to locate the fully labeled “rooms” in the interior of the simplex S. This approach is called “trap-door” argument. If we note, we will find that the rooms of the simplex S can be of 3 types: having no doors, one door and two doors. From the figure, the rooms which do not have corner-1 have no doors; the ones that have all the labels—fully labeled—have one door; and those which have repeated labels—have two doors. Also, no room can have 3 doors, since that would require us to have all 3 corners labeled with a same number, and since doors must have a (1, 2)-edge, there will be a contradiction. In a similar way we can verify that a tetrahedral room with facets labeled with {1, 2, 3} are doors.

Now that we have found boundary doors, we use them to find the fully labeled rooms. We follow a strategy: We start at some boundary door and move into the adjoining room. There are two possibilities: the room we just entered is fully labeled. In that case we have found one!! If it is not, we will escape through that room by moving out of its other door—the “trap-door”. We continue walking through the doors until we hit a fully labeled room or we get out of house through some boundary door. One thing that needs to be noticed is that a path does not contain a loop, i.e. it does not traverse a room more than once. And also, since there are a finite number of rooms in the house, this procedure terminates. The termination occur when we reach either a fully labeled room or by moving out of the house through some boundary door.

The boundary doors which led us to the outside of the house pair up and thus are even in number. Other than these are the ones which led us to the fully labeled “rooms”. Since, the total number of boundary doors is odd and the number of trap-doors is even, the number of boundary doors having a path to fully labeled rooms must be odd. Apart from these reachable fully labeled rooms, there can also be other fully labeled rooms, not reachable from the boundary doors. These unreachable rooms form pairs through their trap-door paths. Hence, they are also even in number. So, the total number of the fully labeled rooms, reachable or unreachable, is odd in number. This completes the proof.

In general, the above proof gives us a way to find the fully labeled rooms in an arbitrary n-simplex in the following way: We start from the lowest dimension, i.e. a point on an elementary (1, 2)-edge, and traverse through the higher dimension simplices. At each stage $'k'$ we assume a simplex in k^{th} dimension as a room and $k - 1$ dimension as a door. So, essentially we start from a point (door) on an edge and get to a fully labeled elementary-edge (room in 1-dimension); start from an elementary edge (door) and move to a fully labeled triangle (room in 2-dimensions); start from a elementary triangle (door) and move towards a fully labeled tetrahedron (room in 3-dimensions) and so on, until we get to a room in the highest dimension. Once we get there we have found the solution.

4.2 Simmon's approach to cake cutting

Now that we have presented Sperner's lemma, we show how we can use it to find an envy-free cake cutting solution. The approach is as follows. We start with mapping the problem of finding a fully labeled room in the "trap-door" problem, discussed above, to the problem of finding an envy-free cake-division problem. As we will see, the vertices of rooms represent pieces of the cake. Fully labeled rooms represent space formed by closely related cake divisions. And the problem of finding a fully labeled room reduces to a problem of finding an 'approximate' envy-free cake division. We will then show the existence of envy free cake division and discuss how to find such a division.

Suppose we have to divide a cake among n people (we call these players, as they are going to perform some action), who have different utility functions about the valuation of the cake. The cake can be divided using $n - 1$ knives as shown in the following figure. We can assign piece x_i to a person-i. The size x_i is the physical size of a piece and has no relation to the valuation given by the person-i. The set formed by these cuts is called a cut-set and is completely determined by the relative size of the pieces. Since each of these cut-sets has $n - 1$ cuts we can assume these to form an (n-1)-simplex in an n-dimensional real space-$\mathbf{R}^n$.

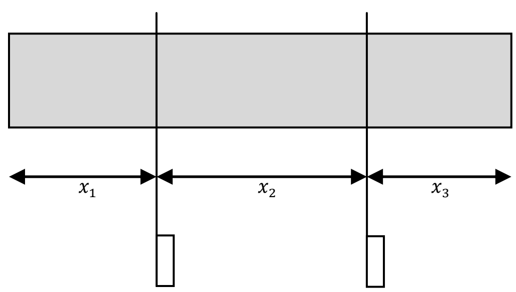


Figure 7.Division of the cake with* $\mathbf{n-1}$ *knives

We assume that for a given cut set, a player always prefers a given piece x_i, if according to him there is no other piece better than x_i. Also, these choices depend upon the cut-sets and the players' preferences only, and are independent from the choices of other players.

Now we make two assumptions:

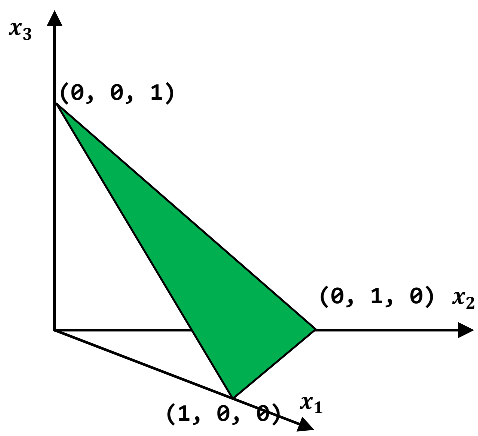
1. *The players are hungry:* A player always prefers a piece that has some mass than an empty piece.

2. *Preference sets are closed:* This means that a piece which is preferred for a convergent sequence of cut-sets is preferred at the limiting cut-set. We can note that this condition eliminates the situation where a single point of a cake can be assumed to be of some positive value.

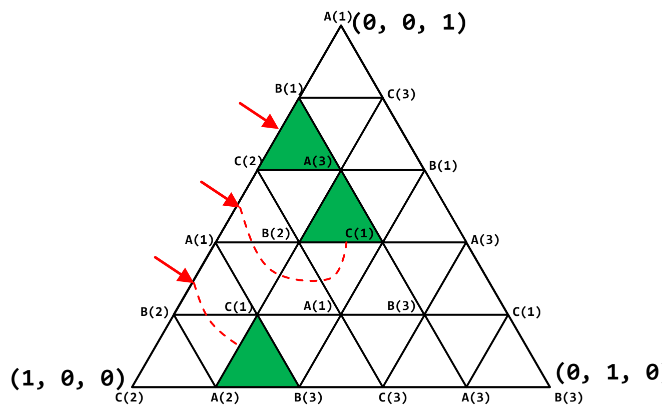
Theorem

For hungry players with closed preference sets, there exists an envy-free cake division, i.e. a cut set for which each person prefers a different piece.

We analyze the theorem for $n = 3$, and state that it can be extended to an arbitrary n. Let the players are A, B and C and they are assigned pieces x_1, x_2 *and* x_3 by the actual sizes. We can then write $x_1 + x_2 + x_3 = 1$ with every $x_i \geq 0$. The solution space of these equations lie on a plane intersecting with the three axes x_1, x_2 *and* x_3, in the first octant, as shown in the figure below. It is a triangle. Each of the point on this triangle is a possible cut-set of the cake. Now we triangulate this triangle into smaller elementary triangles, and assign "ownership" on their vertices by A, B and C in such a way, that each elementary-triangle has a distinct owner on each of its vertices. We then do an auxiliary labeling on these smaller triangles by 1, 2, and 3 as follows: For each elementary simplex we ask the owner of each of the vertices the following question: "If the cake were divided with *this* cut-set which piece would you prefer?" Whatever the answer is, 1, 2, or 3, we label the vertex with that number. An example of such an auxiliary labeling is shown in the figure 8(b).



a)



b)

Figure 8 a) solution space of cut-sets b) Labeling by ownership and preferred piece number (in brackets). The fully labeled triangles and their reachable paths are also shown.

A number in a bracket represent a preferred piece by the owner of that vertex. The claim is: the labeling that results is Sperner labeling!! This is because of the following observation: Each of the vertices of the simplex-*S*, i.e. (1, 0, 0), (0, 1, 0) and (0, 0, 1) has one full piece and two pieces of '0' size. Whosoever is the owner of that vertex will obviously prefer that full piece. Also any arbitrary point on any edge of the simplex *S* is also devoid of one piece, thus would be labeled with one of the two end points of the edge. The interior points of the simplex *S* can be labeled with any of the labels 1, 2 or 3. But we know that these are the conditions for Sperner labeling. Hence the solution space triangle is a Sperner labeled triangle. What we can infer from this is that there will be an odd number of fully labeled ABC elementary triangles in the interior of the simplex *S.* In particular there will be at least one. These fully labeled triangles are shown in the figure 8(b). Hence, we have found

3 very similar cut sets (vertices of any fully labeled elementary triangle) in which every player will choose a different piece.

Now if we try to decrease the size of these elementary triangles smaller and smaller, in the limit the vertices of elementary triangles approach to a single point. This point represents the existence of 'a' cut-set where every player will choose a different piece.

However, as the elementary triangles become smaller and smaller, the number of fully labeled triangles becomes larger and larger and the paths to get to these triangles become longer and longer (in terms of number of rooms traveled along a path). Since, length of a path represent the number of steps needed or the complexity of a procedure, we are interested in paths of reasonable lengths that can led us to an approximate envy-free solution.

With this discussion we conclude that there exists an envy free cake division (represented by the cut) in which each player prefers a different piece. And, to find an 'approximate' cake-division we can follow the trap-door strategy mentioned in the previous section. Figure 8(b) shows paths to get to the approximate solutions.

The above proof can be easily extended to the case with $n > 3$. However, in that case, the triangulation of the simplex S becomes an issue. As we saw earlier, there is a step where we assign ownership to every vertex of each elementary simplex in S. When $n > 3$, that does not generalize easily. For any arbitrary n, we then take the help of what is called $barycentric\ subdivision$, to do the triangulation. For more details about this procedure refer to [5][13].

5 Further Readings

In this lecture, we have discussed several approaches which are classified as *continuous* fair division algorithms wherein the good(s) can be divided in infinitely many ways (e.g. piece of land, large sum of money, cakes). Few more interesting procedures in this category are:

a. Dubins-Spanier's Moving Knife [9] - although most efficient (n-1 cuts) but cannot be translated into discrete sequence of steps and hence are not of much interest.
b. Steinhaus's Lone Divider [1] - is just an extension of the Cut-Choose procedure to more than two players.
c. Banach-Knaster's Last Diminisher [4] [9] – is similar to Divider-Conquer procedure but requires $O(n^2)$ cuts.

Then there are *discrete* fair division algorithms wherein the good(s) are indivisible (e.g. houses, boats, valuable paintings). Few popular procedures in this category are:

a. Sealed bids [4] [10] – is an old and much used method where cash value is associated with each item by the players.
b. Method of markers [4] [10] – has similarities to Last Diminisher and works well for something like bag of Halloween candies.

During the research, we came across at least three books published on this topic and the one which is most relevant from computer science perspective is written by Robertson and Webb [3].

Moreover, many variations on fair division have been seen in the literature where in addition to the 'goods' there are 'bads' an undesirable entity which nobody wants to share, or everybody wants to minimize. One such example is *chore-division* by Marin Gardner, where there is a need to divide an undesirable entity *chore* fairly among *n* people. *Rental harmony* [5] is a problem proposed by Francis Edward Su which talks about dividing the house rent in such a way that every person prefers a different room.

6 References

1. H. Steinhaus. The problem of fair division. Econometrica, 16:101-104, 1948.
2. Fair Division article at Wikipedia. http://en.wikipedia.org/wiki/Fair_division.
3. Jack Robertson, William Webb. Cake-Cutting Algorithms: Be Fair if You Can.
4. Peter Tannenbaum. Excursions in Modern Mathematics, Sixth Edition.
5. Francis Edward Su. Rental Harmony: Sperner's Lemma in Fair Division. Appeared in: Amer. Math. Monthly, 106(1999), 930-942. Retrieved from http://www.math.hmc.edu/~su/papers.dir/rent.pdf.
6. S.Even and A.Paz. A note on cake cutting. Discrete Applied Mathematics, 7:285-296, 1984.
7. Francis Edward Su. Review of Cake-Cutting Algorithms: Be Fair if You Can. Appeared in: Amer. Math. Monthly, 107(2000), 185-188. Retrieved from http://www.math.hmc.edu/~su/papers.dir/review.pdf.
8. Malik Magdon-Ismail, Costas Busch, and Mukkai S.Krishnamoorthy. Cake-Cutting is Not a Piece of Cake. Retrieved from http://www.cs.rpi.edu/~magdon/ps/conference/cake_conf.pdf.
9. Ulle Endriss. Tutorial on Fair Division. University of Amsterdam. Retrieved from http://staff.science.uva.nl/~ulle/teaching/cost-adt-2010/endriss-fair-division-slides-cost-adt-2010.pdf.
10. The Discrete Mathematics Project, CU Boulder. Fair Division Problems and Fair Division Schemes. Retrieved from http://www.colorado.edu/education/DMP/fair_division.html.
11. Steven J. Brams, Michael A. Jones, Christian Klamler. Divide-and-Conquer: A Proportional, Minimal-Envy Cake-Cutting algorithm. Retrieved from http://mpra.ub.uni-muenchen.de/22704/1/MPRA_paper_22704.pdf.
12. Steven J. Brams, S. D. Taylor. An Envy Free Cake Division Protocol. Retrieved from http://www.jstor.org/stable/2974850?origin=JSTOR-pdf.
13. J. W. Vick, Homology Theory. Springer-Verlag, New York, 1994
14. Jonathan Huang. On the Sperner Lemma and its Applications. Retrieved from http://www.cs.cmu.edu/~jch1/research/old/sperner.pdf