1. Fair Division

Fair division is a method of dividing resources among number of participants in a way that they think that they have got a fair share. To express it in mathematical terms, we divide a set of resources S, S={S1, S2, ..., S_N} among N participants in such a way, that according to each participant, the share which they have received, is at least worth 1/N of total value of S to be considered as fair share.

To understand fair division, we make two assumptions:
1) Each player has/her own value system to assign a real, non-negative value to each share. Participants are unaware of each others' value system.
2)For any participant, sum of values of all shares is 1.

The value system of participant is a value function particular to that participant with which he evaluates his share in given resources.
Let's take a cake as an example of resources. Let's assume that this cake is half chocolate and half vanilla as shown in Fig 1a. We consider two participants, A and B. A likes both chocolate and vanilla part equally and B likes only vanilla part. He considers chocolate part to be worthless.
Now if A is asked to divide the cake into 2 equal pieces as shown in Fig 1b, then he divides it in a way, that it's half the original cake size, because he likes both chocolate and vanilla part equally. But if B is asked to divide the cake into 2 parts, then he will divide it in a way, that both pieces have same amount of vanilla in it. Chocolate content can vary in each piece as he considers it to be worthless. This is shown in Fig 1c.

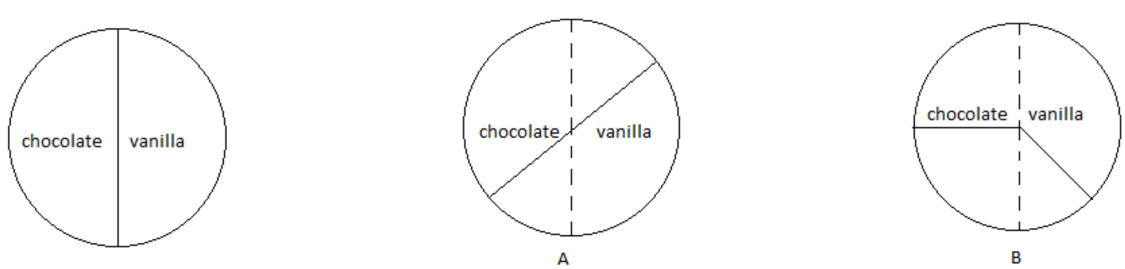

Fig 1: a) original cake b)division made by A c) division made by B

Depending on the nature of resources S, it can be classified into 3 categories:
1)Continuous: S can be divided in infinitely many ways, e.g. - cake, money
2)Discrete: It is a set of objects which cannot be divided, e.g. - houses, paintings, etc
3)Mixed: Most inheritances fall under this category. It is a combination of continuous and discrete objects, e.g.- houses + money+ paintings, etc.

Fair division algorithms are different in a way that there is no central control over execution of algorithm as divisions are done from participants' perspectives, as per their value systems. Also, the running time is measured in terms of number of cuts performed rather than no. of atomic operations which is usually the case.

1.1 Divide and Choose

This is the classic case of the “I cut you choose” method [2]. The problem is defined for 2 participants, where the first participant cuts and the other chooses.

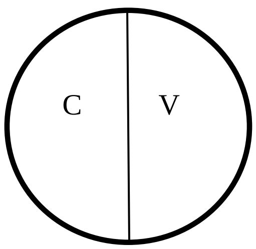

Figure 2: Initial Cake

Suppose A1 values only chocolate and not vanilla, A2 does not have preferences. Now, if we ask A1 to cut the cake into equal halves, he/she can cut it as follows:

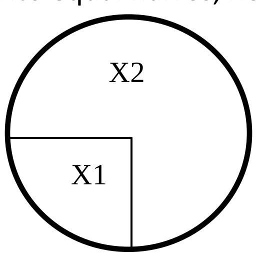

Figure 3: Cut made by A1

$f_{A1}(X_1) = ½$, and $f_{A1}(X_2) = ½$ …… (1)

Now, A2 gets to choose the bigger piece out of the two. Since A2 does not have any preferences,

$f_{A2}(X_2) = ¾$ and $f_{A2}(X_1) = ¼$ …… (2)

<u>Proof of Correctness</u>

If Xi is the piece allotted to Ai, the criteria for fair division is

$f_{Ai}(X_i) \geq 1/n$

from equations 1 and 2, the above criteria is satisfied.

The criteria for envy freeness is $f_{Ai}(X_i) \geq f_{Ai}(X_j)$ for all i and j
Now, $f_{A1}(X_1) \geq f_{A1}(X_2) = ½$ and $f_{A2}(X_2) \geq f_{A2}(X_1)$
Hence the division is also envy free.

1.2 Lone Chooser

We have seen the case of the divide and choose method for two players. Now let us extend it to n persons. This problem is also known as the Lone Chooser or the Successive Pairs algorithm. This works by performing divide and choose for n players. When a third person comes in, he shall be entitled to a third of the cake.
Both A1 and A2 trisect their respective pieces, obtained from the fair division for two people. Now, we let C choose one piece from the trisected pieces of A1 and one from the pieces of A2.
Proof of correctness:
We shall now prove that the algorithm is fair to all the three players involved
After division of the cake into two pieces, A1 and A2 have
$f_{A1}(X_1) = ½$, $f_{A2}(X_2) \geq ½$
Now, A trisects X1 into X11, X12, X13 and B trisects X2 into X21, X22, X23 such that
$f_{A1}(X_{11}) = f_{A1}(X_{12}) = f_{A1}(X_{13}) = 1/6$ and $f_{A2}(X_{21}) = f_{A2}(X_{22}) = f_{A2}(X_{23}) \geq 1/6$ …. (3)

A3 gets to choose the largest piece form part of A1 and part of A2. Suppose A3 chooses X_{13} and X_{23}. Since A3 has got to choose the largest piece, by his valuation, he has got at least 1/3 of the total cake.
$f_{A3}(X_{13}+X_{23}) \geq 1/3$
Through equation (3)
$f_{A1}(X_{11} + X_{12}) = 1/3$ and $f_{A2}(X_{21} + X_{22}) = 1/3$
Hence, each person has got 1/3 (fair share) of the cake by their own evaluation.
Now we extend this for n players, A1…An. The parts of the cake allotted to the players are X1…Xn respectively. If a piece Xi is split into n pieces, it is denoted by Xik, k=1 to n. The pseudo code is as follows:
At the end of the division,
$f_{Ai}(X_i) \geq 1/n$ … each player believes he has got his fair share. The pseudo code for the algorithm is as follows:

```
LoneChooser
{
 X1 = 1
   for i : 2 to n
```

```
    for j : 1 to (i-1)
        Aj.divide (i)           //Splits Xj into I pieces
         Ai.choose(j)           //Chooses a piece from {Xjk}..k=1 to (i-1)
        Aj.recombine()          //Sets Xj = ∑Xjk except piece chosen by Ai
    end
  end
}
```

Running Time:

The running time for fair division algorithms is not measured in the amount of atomic operations performed. It is instead measured in the amount of cuts that are made on the piece of cake. So T(n) represents the number of cuts made for n players.

$T(n) = n \times (n – 1) + T(n-1)$

$= n \times (n-1) + (n-1) \times (n-2) + ... + 3 \times 2 + 1$

$= O(n^3)$

2. Envy Free Division

An algorithm is termed as envy free if every participant values their own piece more than he values all the other pieces. It follows the following inequality:

$f_{Ai}(X_i) \geq f_{Ai}(X_j)$ for all i and j

The algorithm explained above is not envy free, as it is not possible for A1 to influence how A2 divides his share into three parts. If A1 thinks that A2 divided the portions unequally, A3 or A2 might have an advantage in A2's eyes. This is a classic case where an algorithm is fair but it is not envy free.

Most of the fair division algorithms are not envy free, unless there is a way for the player to influence the outcome of the division after he has taken his fair piece.

Selfridge Conway algorithm:

This algorithm is designed for 3 people. Consider 3 players, A1, A2 and A3. Any one of the players is told to cut the cake into three equal pieces. Let it be A1.

If A2 thinks that at least two are tied for the largest, he simply passes the pieces to A3. Else he trims one of the pieces to create a two-way tie for the largest piece and passes them to A3. Now, A3 chooses from all the pieces. A2 chooses next, and if there is a piece that A2 has trimmed and is available, A2 has to choose that piece. Finally, A1 takes the third piece.

The person who received the trimmed piece is called as the cutter (say A2) and the other as the non-cutter (A3).

Now we partition the left over piece. The partition is such that the non-cutter trisects the leftover piece. The players then choose in the order cutter, A1, non-cutter.

Correctness:

Before we partition the leftover piece, A1, A2 and A3 all believe they have the largest piece and are not envious of each other. After partition of the leftover piece, the following relations come up.

A1 is envy free of A2 as the trimmed piece, with leftover added to it would still be as equal as A1's share. A1 is envy free of A3 as at step __ both have equal pieces and out of the leftover piece, A1 gets to choose first.

A2 is envy free of A1 and A3 as he believes he has the largest piece from the left over pieces.

A3 is envy free of A1 and A2 as he believes everyone got equal share of the leftover piece.

For n ≥ 4

When we try to extend this algorithm to four people, we run into a problem. If A1 makes four cuts, A2 has to create a 3-way tie for the largest piece. A3 trims one piece to create a 2-way tie. Now if A4 chooses one of the un-trimmed pieces, then A1 will have to choose a piece that has been trimmed by either A2 or A3. This would result in the algorithm being unfair and would lead to envy [3].

The solution to this problem lies in the first player cutting the cake into more number of pieces than there are people. For example, if A1 cut the cake into 5 equal pieces and then A4 chooses an untrimmed piece, A1 is still left with an untrimmed piece at the end.

The question arises, what do we do with the extra piece and the trimmed leftovers? For n ≥ 4 there is no way to deal with the trimming leftover piece to ensure an envy free division. We just pass the piece through the algorithm again. Again A1 partitions the piece into 5 pieces (for four people) and this continues. The assumption is that at infinite time, the algorithm would ensure an envy free division for all.

To put a limit on the algorithm, we discuss a special case of envy free algorithms. Here we define a constant such that

$f_{Ai}(X_i) + \varepsilon \geq f_{Ai}(X_j)$ for all i and j

This is an interesting case of envy freeness. Here, a person is not envy free, if his/her piece is ε smaller than the largest piece given to anyone. This puts a limit on the running time of the

algorithm, since whenever the size of the leftover (extra) piece becomes less than Ɛ, we can randomly divide it among the people or just assign it to a particular person.
Even if the entire piece is given to one person, others will not be envious of that person due to the condition explained above.

3. Fair Division Using Sperner's Lemma

There is another technique for fair division which makes use of Sperner's Lemma. A paper was written by Francis Su in 1999[1], where he proved that Sperner's Lemma can be used for dividing resources fairly among n participants.

3.1 Sperner's Lemma

Sperner's Lemma is performed on n-dimensional simplex. Here simplex is an n-dimensional geometrical structure. Hence a 0-simplex is a vertex, 1-simplex is a line, 2-simplex is a triangle, 3-simplex a tetrahedron and so on. As mentioned in paper, "an n-simplex is a convex hull of n+1 affinely independent points". In case of 2-simplex, we have 3 independent points. Let the points be A, B and C. A convex hull of 3 points then form a 2-D triangle embedded in 3-D Euclidian space as shown in Fig 2.

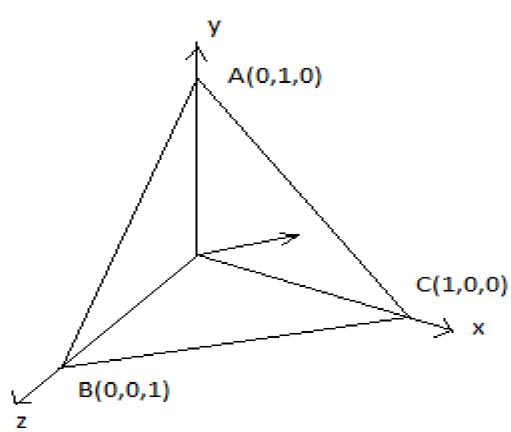

Fig 4: 2-simplex in 3-D Euclidian space

In this way, an n-simplex can be embedded in (n+1)-D space.
An n-simplex can be built by combining (n+1) number of (n-1) simplices as its facet. Hence a 2-simplex triangle is built by combining three 1-simplex lines and a 3-simplex tetrahedron is built by combining four 2-simplex triangles as its facet and so on.

Let's consider a special case where n=2. Hence the geometrical structure will form a triangle. Let's call this triangle as T.

We triangulate the triangle T into many smaller triangles, called elementary triangles such that they have either a vertex or edge common between them. The vertices are labeled by 1's, 2's and 3's.

The labeling follows two conditions:

1) All the main vertices of T have different labels.

2) The label of vertex along any edge of T should match one of the labels spanned by that edge. Labels of interior vertices are arbitrary.

Any triangulation satisfying this labeling is said to be following Sperner's Labeling as shown in Fig 3.

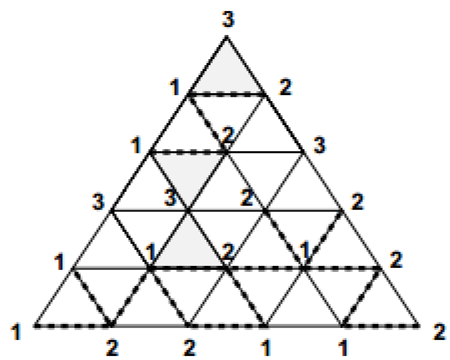

Fig 5: Sperner's Labeling of 2-simplex triangle (Figure from Alex Wright's paper)

Then Sperner's Lemma states that, "a Sperner-labeled triangulation of T must contain odd number of elementary triangles possessing all the labels. In particular, there is at least one".

This can be extended for n-simplex triangulation inductively. In that case, Sperner's Lemma states that, "Sperner-labeled triangulation of n-simplex will have at least one elementary simplex possessing all the labels and at most odd number of such fully labeled simplices".

3.2 Algorithm for locating a fully labeled simplex

We need to locate a fully labeled simplex in the triangulation. We continue with our example of triangle. Before that, we need to know a basic fact:

The number of fully labeled edge segments on that corresponding edge is always odd. For example, the number of 1-2 edge segments on 1-2 main edge of T will be odd.

This can also be proved using Sperner's lemma itself. consider the edge to be a 1-simplex line. Since it satisfies the labeling criteria of Sperner's labeling, the number of elementary edges in it will be odd.

Now we consider the main triangle T as a house, its elementary triangles to be rooms and each 1-2 edge segment as door. We consider a path through triangle starting from outside the triangle. We enter a triangle room and exit it only through 1-2 edge segment door.
This path will terminate only in two cases:
1) It leaves the main triangle again.
2) It finds a fully labeled triangle.
This is shown in Fig 4.

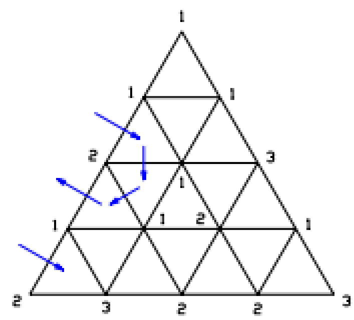

Fig 6: Possible Paths (Figure from Alex Wright's paper)

This path is uniquely determined since no triangle has three 1-2 edge segments as its sides. This also means that no 2 paths can end up in same triangle.
Now, how can we be sure that such path exists?
If a path terminates by exiting the triangle, it means that a pair of 1-2 edge segments is used up. Since the number of edge segments is odd, there is one edge segment left which leads us to fully labeled triangle.

The above explanation can be extended to n-simplex inductively.

The following algorithm [4] follows the same concept to find a fully labeled elementary simplex in n-simplex triangulation.

get triangulated n-simplex T^n
for each fully labeled elementary 1-simplex t^1 in T^1 // T^1 is the 1-simplex labeled 1-2
  set k to 2
  while k<n
    call compute-k-simplex with t^{k-1} returning t^k //t^{k-1} is labeled from (1,2,..k)
    if t^k is fully labeled // i.e., t^k is labeled from (1,2,..k,k+1)
      set t^{k-1} to t^k
      increment k

```
        else
           set tk-1 to the other facet of tk with label (1,2,...,k,k+1)
        endif
    endwhile
endfor
```

The pseudocode for method compute-k-simplex is as follows:

get k-1 simplex t^{k-1}
if t^{k-1} lies on T^{k-1} // t^{k-1} is a facet of T^n and lies on its surface
  set t^k to null
endif
set t^k to the elementary k-simplex which has t^{k-1} as its facet
set NewVertex to the vertex in t^k but not in t^{k-1}
if label of NewVertex is not one of the (1,2,...,k)
  set t^k to null
endif
return t^k

3.3 Application of Sperner's Lemma In Finding Fair Division

Let's see how Sperner's lemma can be applied to produce fair division of resources.
We will take cake as the example of resources here.
First, we make following assumptions:
1) Total size of cake is 1.
2) Physical size of i^{th} piece is x_i and is absolute measure, independent of participants' preferences.
Hence, if there are n participants, then
$x_1+x_2+...x_n=1$ and $0 \leq x_i \leq 1$
3) Participants are hungry - It means that, participants will choose a piece with some mass rather than empty pieces.
The space S of possible partitions forms a standard (n-1) simplex in Rn. Each point in S corresponds to a partition of cake by a set of cuts in the cake which is called a cut-set.

Theorem: For hungry participants, there exists a fair division, i.e., a cut-set for which each participant prefers a different piece.

Let's prove this for n=3. So we have three participants, A, B and C. In this case, the geometrical structure used will be a triangle for (n-1), i.e., 2-simplex.

We ask these three participants to the cake of size 1 into three pieces of sizes x_1, x_2 and x_3.
Hence, $x_1+x_2+x_3=1$ and $0 \le x_i \le 1$
Now, we triangulate S and assign ownership to each vertex of triangles as shown in Fig. Assignment should be such that each elementary triangle is an ABC triangle. This is the ownership labeling. This is shown in Fig 5.

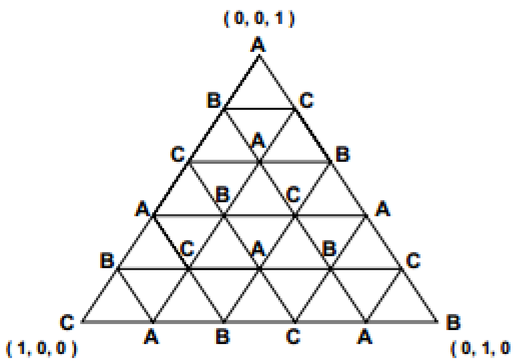

Fig 7: Ownership Labeling (Figure from Alex Wright's paper)

We obtain a new auxiliary labeling of triangulation by 1's, 2's and 3's by doing following: We ask the owner of each vertex to choose a piece. We then label that vertex with the number of that piece chosen. In this way, the auxiliary labeling obtained is a Sperner's labeling.

So how is it a Sperner's labeling?

A cut-set is represented in the form of sizes of pieces, (x_1, x_2, x_3). At vertex (1, 0, 0) of S, one of the pieces contains the entire cake and the other pieces have zero mass. Hence, by hungry participant assumption, the owner of (1, 0, 0) will take piece 1. Therefore, that vertex will be labeled as 1. Similarly (0, 1, 0) vertex will be labeled as 2 and (0, 0, 1) vertex will be labeled as 3. In this way, the main vertices of triangle have different labels, 1, 2 and 3. This satisfies first condition of Sperner's labeling.

On the sides of triangle, there will always be a piece of cake which will be devoid of any mass. Therefore, due to hungry participant assumption, this piece will never be chosen by any participant. Hence labeling corresponding to that piece will be always missing on the edge. In this way, the labels of all vertices lying on main edge of triangle with match one of the labels spanned by that edge. Thus, the second condition of Sperner's labeling is satisfied.

Then by Sperner's lemma, there will be at least one (1,2,3) elementary triangle in this triangulation. And existence of such fully labeled triangle means that there is at least one possibility that all participants prefer different pieces.
We now apply the algorithm to locate such fully labeled elementary triangle to find cut-set corresponding to a fair division.

To show that such single cut-set exists that satisfies everyone with different pieces, we carry out the same procedure on that triangle with finer and finer triangulations, each time yielding smaller and smaller (1,2,3) elementary triangles. Eventually, after continuously decreasing the size of triangles, it converges to a single point and such a point corresponds to a cut-set in which participants are not only satisfied with different pieces but also not envious of each other hence leading to an envy-free solution. This can be extended to (n-1) simplex inductively for n participants.

The worst case running time of Sperner's lemma is $O(N)^k$ where k is the dimension of simplex and N is the number of vertices on its facet, i.e., it depends upon the granularity level of triangulation.

4. References

1. Su, F. E. "Rental harmony: Sperner's lemma in fair division." American Mathematical Monthly 106: 930-942, 1999.

2. Steven J. Brams and Allan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, New York, NY, 1996.

3. Steven J. Brams and Allan D Taylor."An envy-free cake division protocol." Am. Math. Monthly, 102:9–18, 1995.

4. Ongaro J.N. and G.P Pokhariyal. "Application of Sperner's Lemma in Fair division", School of Mathematics, University of Nairobi.

5. Alex Wright, "Sperner’s Lemma And Brouwer’s Fixed Point Theorem"