The Hong Kong University of Science and Technology
School of Science - Department of Mathematics

MATH 4994

Capstone Project in Mathematics and Economics

Bounded, Discrete, Fair Allocation Procedure for 4 Players

Written Report - 2023/24 Fall

AU, Yik Hau | CHEUNG, Tsz Hin

Content

1. Introduction

1.1 Motivation

The cake-cutting algorithm is a metaphor for resource allocation, by cutting the cake "fairly", we can expect optimal allocation of the resource for individuals. Thus, we aim to find a fair algorithm that can achieve fair allocation, which can satisfy Pareto optimal, envy-freeness, and strategic-proof. Since an envy-free allocation is guaranteed to exist up to (n-1) cuts (F. E. Su. 1999), we would like to discover an algorithm to achieve boundness based on this result. Therefore, this report will utilize the discovery of [A Discrete and Bounded Envy-free Cake Cutting Prtocol for Four Agents] from Haris Aziz, Simon Mackenzie (2016) to discuss a discrete, envy-free algorithm protocol for 3 and 4 players with a bounded number of queries as well as cuts of the cake.

2. Three-Person Allocation Protocol

This protocol makes use of many similar concepts and techniques as the four-person protocol, which will serve as the warm-up for the four-person protocol.

2.1 Methodology and Terminology - Techniques to Achieve Envy-Freeness

1) Terminology

- Player i: (i)
- Complete Cake i: X_i
- Trimmed Cake: X'
- Cake i in stage j: X_i^j
- Residual cake: β
- Effective trim mark on a cake: | i j | (i.e. player i place a knife on the left of player j's knife)
- Bonus: γ (the additional cake from the receiver's trim mark and trimmer's trim mark)
- Player i's valuation on cake i: $V_i(X_i)$

2) Domination

For Domination, when one player (says j) dominates another player (say i), given they are given the cakes X_j and X_i respectively, player j will consider the player i's allocation is at least as good as his own allocation even if player i received an extra trimmed cake, denoted as β.

Then, we can denote player j will not envy player i, and denote that receiving an additional piece β will not affect the result is the irrevocable advantage of player j.

3) Ordinal

For Ordinal, when one player (says j) is ordinal with his own allocation and another player's (says i) allocation, it means that player j has a weak ranking over those pieces but he cannot tell the exact cardinal utility difference. In short, player j is indifferent to player i's allocation, thus he is envy-free of player i.

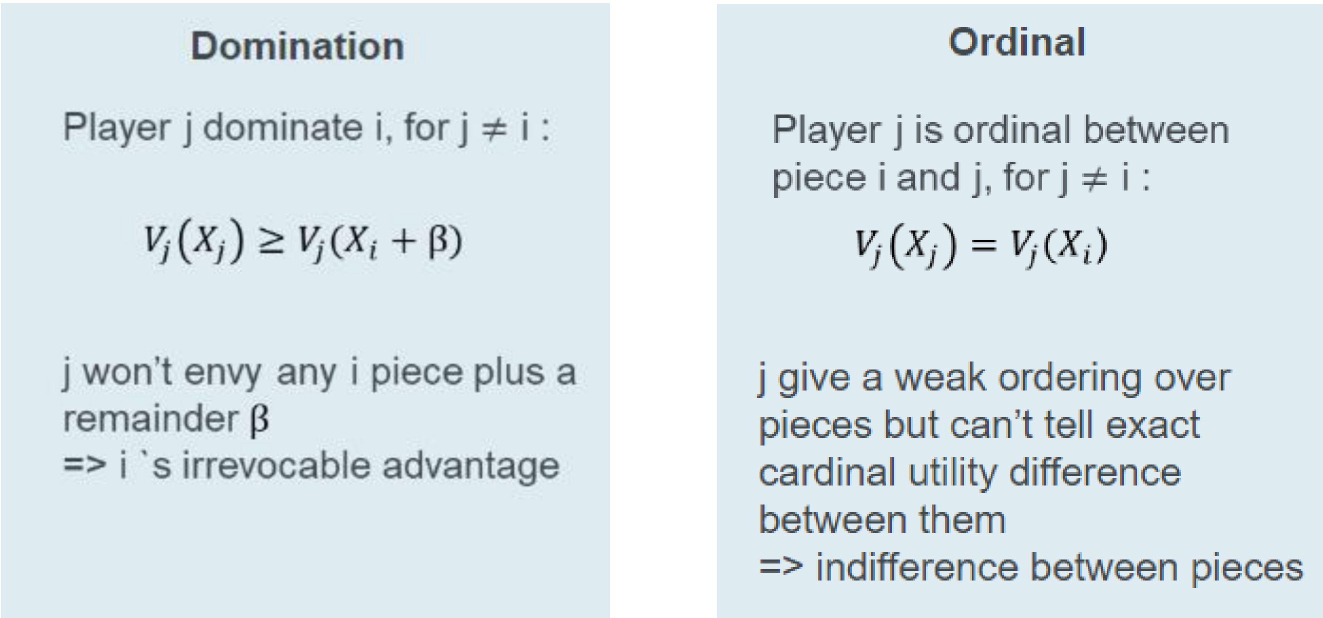

Figure 1: Definition of Domination and Ordinal

2.2 Main Algorithm

Core idea: Implement domination and ordinal among players to achieve envy-freeness

The algorithm has 4 stages:

Stage 0: Player 3 equally trim 3 pieces
Stage 1: Player 1 & 2 trim their most preferred piece
Stage 2: Player 1 & 2 trim their most preferred reminder
Stage 3: Player 1 & 2 exchange their holding

Stage 0:
Player 3 trims the cake into 3 pieces equally in his own valuation.

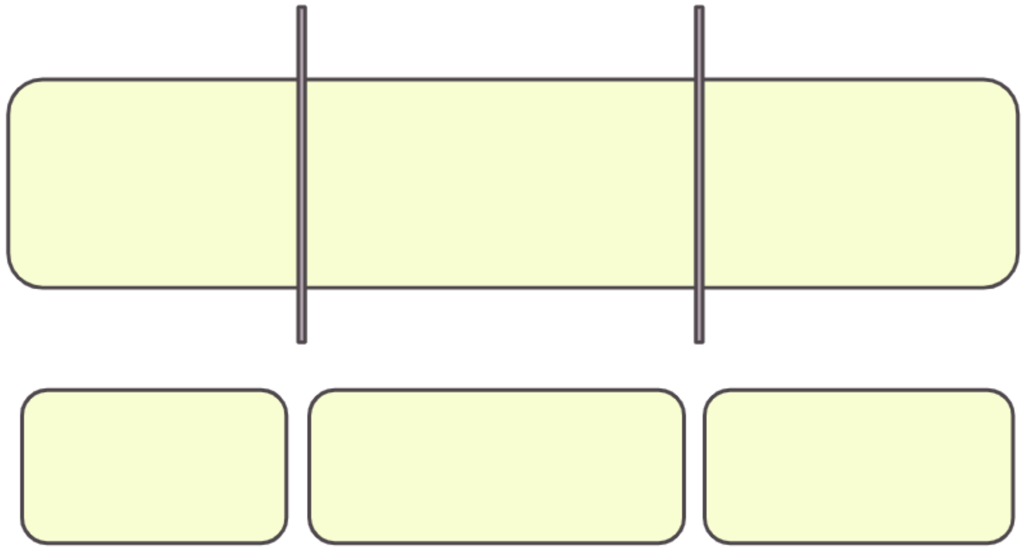

Figure 2: stage 0 for 3 person

Stage 1:
Players 1 and 2 will both hold a knife and place it onto their most preferred piece so that the cake will be trimmed such that its trimmed piece is equal to their 2nd most preferred piece ($V_1(X_1') = V_1(X_2)$, where X_i' denotes as trimmed piece, and cake 2 is player 1's 2nd most preferred piece)

Now we have two cases, which we denote as method 0 and method 1 respectively which will be applied again in the later chapter.

<u>Method 0 (M0):</u>
If players 1 and 2 both prefer different pieces, then they will receive the desired pieces accordingly with no trims needed. Each player received their most preferred piece, ordinal for all players.
All cake is allocated with no residual, the algorithm ends here.

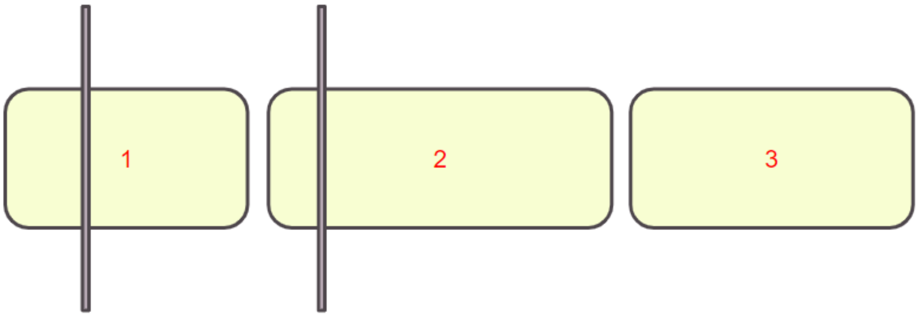

Figure 3: M0 for 3 person

<u>Method 1 (M1):</u>
If both players 1 and 2 prefer the same piece (says cake 1 is their 1st preferred, cake is their 2nd most preferred), the player with the second rightmost trim will trim the cake in his knife position and receive his second most preferred cake instead of the cake he trimmed ($X_1 \text{->} X_1 + \beta^1$). We denote this

player as (-i), and another player will be denoted as (i) who will receive X_2. Player 3 will receive the remaining complete (untrimmed) cake.

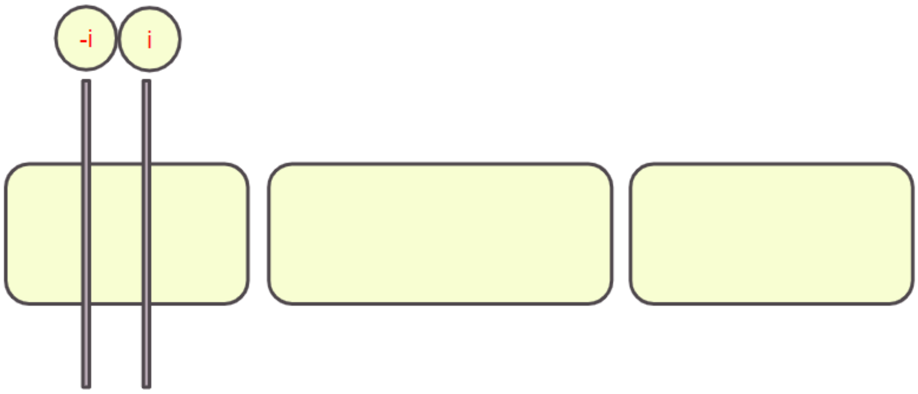

Figure 4: M1 for 3 person

Now the allocation is as follows:

Player	Cake	Valuation
(-i)	X_1'	≥ 2nd most preferred - Dominate (3), (i)
(i)	X_2	2nd most preferred - Dominate (3), (-i)
(3)	X_3	Most preferred - Dominate (-i) - Ordinal (i)
Unallocated	β^1	

Table 1: stage 1 allocation

Note that, cake X_1 is cut at (i) position but is assigned to (-i), thus (-i) received a bonus from the cake 1 between the knife position (i) and (-i), i.e. (-i) receive a cake at least as good as his 2nd most preferred piece, straightly better than X_2, and X_3 is the worst for him, so he dominates other two players.

Since there exists an unallocated cake, we will then proceed to the next stage

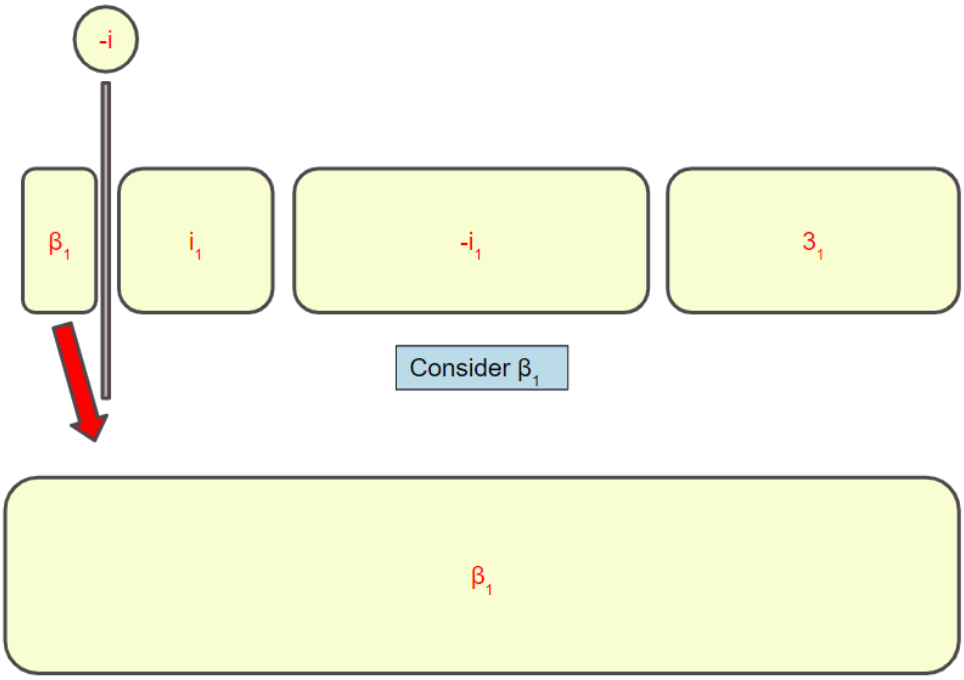

Figure 5: stage 1 M1 for 3 person

Stage 2:

Now we consider β^1 and we re-run the algorithm again.

1. (3) trim the cake into 3 pieces equally in his own valuation.
2. Player 1 and 2 will both hold a knife and place it onto their most preferred piece so that the cake will be trimmed such that its trimmed piece is equal to their 2nd most preferred piece
3. Execute M0 or M1
4. Proceed to the next stage if M1 is executed

Stage 3:

Consider the M1 again, there exists a β^2 unallocated this time. We will consider whether (i) and (-i) in stages 1 & 2 are the trimmer or not (second rightmost knife position) to decide whether a permutation is required.

Case 1: The trimmer in stage 2 is (i) instead of (-i),

We execute the same algorithm again, (i) will receive his 2nd most preferred cake (a complete cake 2)

Now the allocation is as follows:

Player	Cake	Valuation
(-i)	$X_1^{1'}$, X_2^2 (1 complete, 1 trimmed)	≥ 2nd most preferred - Dominate (3), (i)
(i)	X_2^1 , $X_2^{2'}$ (1 complete, 1 trimmed)	≥ 2nd most preferred - Dominate (3), (-i)
(3)	X_3^1 , X_3^2 (2 complete cake)	Most preferred - Dominate (-i), (i)
Unallocated	β^2	

Table 2: stage 3 case 1 allocation for 3 person

Then, β^2 is allocated using cut & choose between (i) and (-i), no one will envy each other because they are dominating each other.

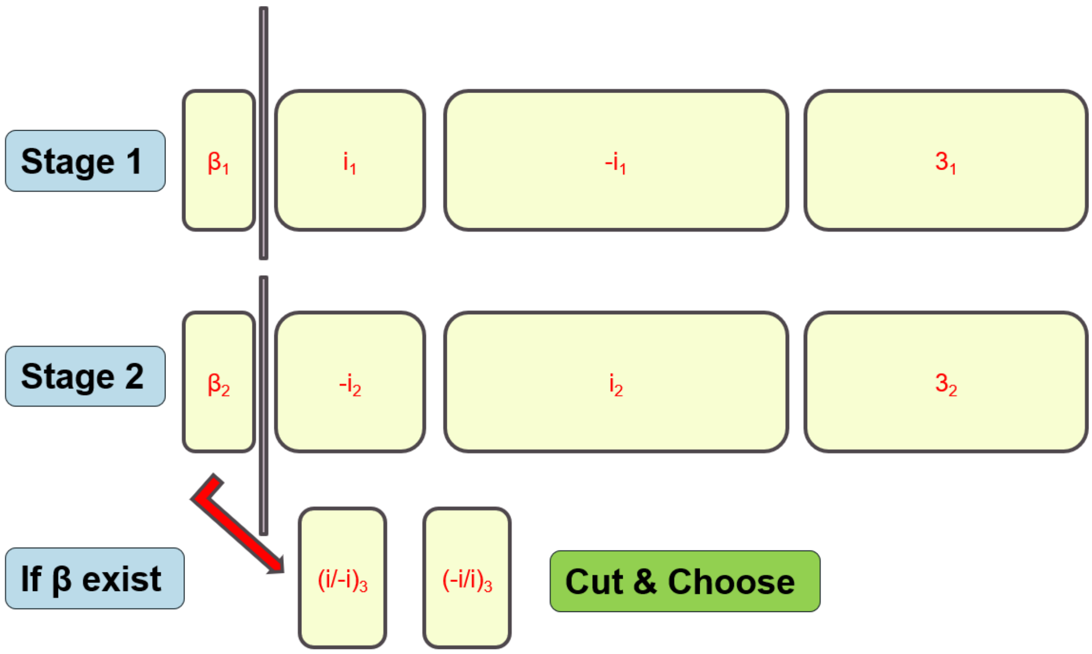

Figure 6: case 1 for stage 3

Case 2: The trimmer in stage 2 is (-i) again,
We execute the same algorithm again, (i) will receive his 2nd most preferred cake (a complete cake 2)

Now the allocation is as follows:

Player	Cake	Valuation
(-i)	$X_1^{1'}$, $X_2^{2'}$ (1 complete, 1 trimmed)	>= 2nd most preferred - Dominate (3), (i)
(i)	X_2^1 , X_2^2 **(2 complete cake)**	>= 2nd most preferred - Dominate (3), (-i)
(3)	X_3^1 , X_3^2 (2 complete cake)	Most preferred - Dominate (-i) - Ordinal (i), will then envy (i) if cut & choose is applied
Unallocated	β^2	

Table 3: stage 3 case 2 allocation for 3 person

Here, we can't use cut & choose for β^2 because (3) is not dominating (i), if (i) receive 2 complete cake, his allocation is equal to (3)'s in (3) valuation, adding a fraction of β^2 will result in (3) envy (i). Therefore, we will permutate (i) and (-i)'s allocation based on the following rule:

<u>Permutation rule:</u>
Denote the cake between (i) and (-i)'s knife position as γ, which represents the additional gain of the receiver for the trimmed cake other than just 2nd most preferred piece.
(i) will then pick a less preferred γ then exchange the trimmed cake with (-i)'s complete cake in that stage so that both (i) and (-i) will not have 2 complete cakes plus a small fraction of residual to prevent (3) from being envy.

E.g. $V_i(\gamma_2) < V_i(\gamma_1)$, then (i) exchange to give out his allocation and (-i)'s allocation in stage 2

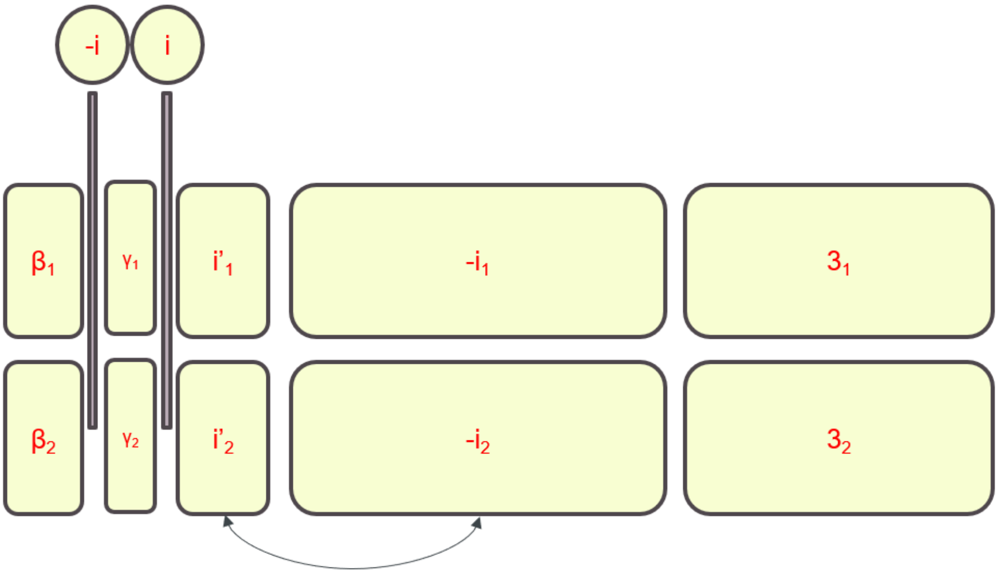

Figure 7: illustration of permutatuion for 3 person

Now, β^2 can be allocated using cut & choose between (i) and (-i), no one will envy each other because they are dominating each other.

3. Four-Person Allocation protocol

3.1 Introduction

This protocol is a combination of 4 algorithms (Core Protocol, Permutation Protocol, Post Domination Protocol and the Overall Protocol) that will be applied recursively to achieve discrete, boundedness and envy-freeness of the fair allocation.
Some observations in the three-person allocation protocol are integrated in the four-person allocation protocol.
The following are descriptive of different protocols:
The Core Protocol aims at achieving a domination of 2 persons for players with an additional bonus in their allocation per algorithm, also it achieves partial envy-freeness.
While the Permutation Protocol will handle the case the domination is over the same person for all the protocol gone through in the Core Protocol stage. It permutates the pieces allocated to different players while preserving the domination with the consequence of reducing the bonus received by players.
Next, the Post Double Domination Protocol will allocate the remaining unallocated cake and achieve envy-freeness.
Lastly, the above 3 protocols formulate the Overall Protocol and return a discrete, bounded, and envy-free fair allocation.

3.2 Post Double Domination Protocol

The Post Double Domination Protocol requires that all 4 persons dominate 2 of the others and return an envy-free allocation after the protocol.

Without loss of generality, we assume that player 1 dominates players 3 and 4. The following will be divided into 3 cases.

Case 1: Player 2 dominates Players 3 and 4
As both players 1 and 2 dominate players 3 and 4, the remaining cake can be divided between players 3 and 4 by cut and choose procedure.

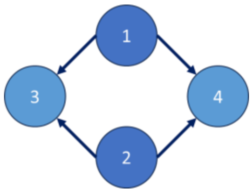

Figure 8: Case 1 for Post-Double Domination Protocol

Case 2: Player 2 dominates Players 1 and 4, while Player 3 also dominates Player 4
All players dominate player 4, therefore Player 4 gets all the remaining cake.

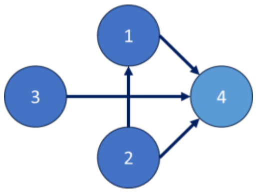

Figure 9: Case 2 for Post-Double Domination Protocol

<u>Case 3: Player 2 dominates players 1 and 4, while player 3 dominates players 1 and 2</u>

We can observe that in between players 1, 2, and 3, player 1 is dominated by 2 persons while player 1 is dominated by 1 person. Regardless of who player 4 dominates, we can assign player 4 as the cutter and cut the remaining part into 4 equal pieces in the view of player 4, and a choosing procedure in the sequence that player 1 is the first to choose, player 2 the second and player 3 the third, such that player 1, 2 and 3 will be envy-free while player 4 is indifference.

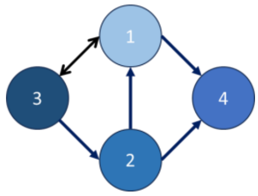

Figure 10: Case 3 for Post-Double Domination Protocol

3.3 Core Protocol

The Core Protocol assigns a cutter who will receive a complete piece and others may get partial pieces. The main technique is to trim the cake using the 3 methods in different scenarios to achieve envy-freeness.

The protocol is expected to be applied on the unallocated piece (β) recursively so that the cutter will eventually dominate two agents.

<u>**1) Brief introduction of the Core Protocol**</u>

Basis of the Core Protocol:

Input:	Cutter (says player 4)
Step:	• (4) cut the cake into 4 equal pieces • (1),(2),(3) each place a knife on their 1st and 2nd most valuable piece respectively to match the value of their 3rd valuable piece (i.e. At least 2 cakes will receive at most 3 knives)
Output:	• Partial envy-free allocation (at least two complete pieces for player (4) and another player) • (Possibly) unallocated piece

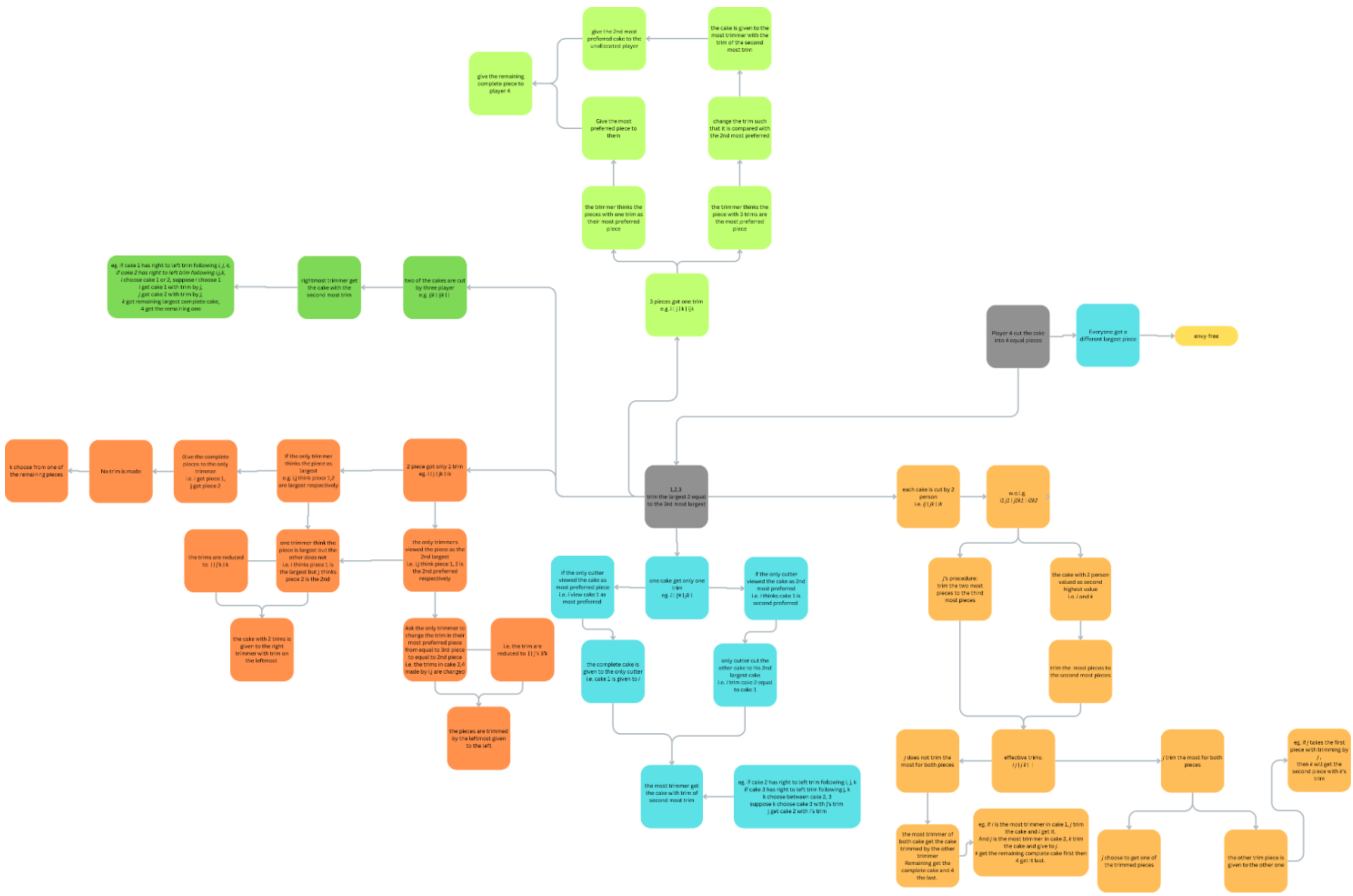

Figure 11: Complete mind map for the Core Protocol

There are a total of 6 different cases:
Case 0:
They all have different 1st preferred pieces

Case 1:
3 pieces received 2 knives respectively
Suppose, WLOG, denote as $i_1j_1 \mid j_2k_1 \mid i_2k_2 \mid$, $\{1,2,3\} = \{i,j,k\}$

Case 2:
2 pieces received 3 knives respectively
Suppose, WLOG, denote as $i_1j_1k_1 \mid i_2j_2k_2 \mid \mid$, $\{1,2,3\} = \{i,j,k\}$

Case 3:
3 pieces received 1 knife, 3 knives, 2 knives respectively
Suppose, WLOG, denote as $i_1 \mid i_2\, j_1k_1 \mid j_2k_2 \mid$, $\{1,2,3\} = \{i,j,k\}$

Case 4:
4 pieces received 1 knife, 1 knife, 2 knives, 2 knives respectively
Suppose, WLOG, denote as $i_1 \mid j_1 \mid j_2k_1 \mid i_2k_2 \mid$, $\{1,2,3\} = \{i,j,k\}$

Case 5:
4 pieces received 1 knife, 1 knife, 1 knife, 3 knives respectively
Suppose, WLOG, denote as $i_1 \mid j_1 \mid k_1 \mid i_2\, j_2k_2 \mid$, $\{1,2,3\} = \{i,j,k\}$

2) Methodology for Core Protocol

Method 0 (M0):
If players i, j, k all prefer different pieces, then they will receive the desired pieces accordingly with no trims needed. Each player received their most preferred piece, ordinal for all players.
All cake is allocated with no residual, the algorithm ends here.

Method 1 (M1):
If there are two players (says i and j) who prefer the same piece (says cake 1 is their 1st preferred, cake is their 2nd most preferred), the player with the second rightmost trim (says j) will trim the cake in his knife position and another player (i) will receive this trimmed cake.
E.g. Effective trim on one cake:
| | -> | j i | -> cake trim at (j)'s position and then assign to (i)

Method 2 (M2):
For trimmer (say player j) who trimmed his 2nd most preferred piece as a solo cutter (i.e. that cake only has his own trim mark) will be given one knife only and use it to re-trim his 1st preferred cake so that this will equal to his 2nd most preferred piece and ignore his original trim mark.
Then we execute M1 on the cake with multiple trim mark
E.g. Effective trim on two cakes:
| j | j k | -> | | j' k | (j' trim on (j)'s 1st preferred piece such that it trimmed to match his 2nd most preferred piece) -> (k) receive cake trimmed up to (j)' position

3) Details for Core Protocol Algorithm

Player 4 cut the cake into 4 equal pieces according to his valuation
Player 1,2,3 values those pieces

(Case 0)
If they have a different complete most-preferred piece **then**
Execute M0
(assign the cake to them accordingly & assign the remaining complete piece to player 4)
return envy-free allocation
End if

Player 1,2,3 will trim the left-hand side of their 1st and 2nd most valuable piece equally valuable to their 3rd most preferred piece

(Case 1)
If we are in case 1 (2,2,2,0 trim for 3 piece) i.e. $i_1 j_1$ | $j_2 k_1$ | $i_2 k_2$ **then**
i & k put new trim on their most preferred piece which equal their second most valued piece; j same trim (effective trim: i j | j k | | |)

If j does not trim the most for both piece **then**
(consider $i_1 j_1$ | $j_2 k_1$ | | |**)**
Execute M1 + M0:
- rightest trimmer receive the cake trimmed up to 2nd rightest trim ;
- remaining out of {i,j,k} choose one complete cake ;

- (4) take remaining complete ;

else if j trim the most for both piece **then**
(consider $i_1\, j_1 \mid k_1\, j_2 \mid \mid |$**)**

- j pick one of the trimmed cake that i or k trimmed ;
- i or k who trim the other cake receive the trimmed ;
- i or k who didn't trim the other cake pick a complete cake and (4) pick the remaining complete ;

End if

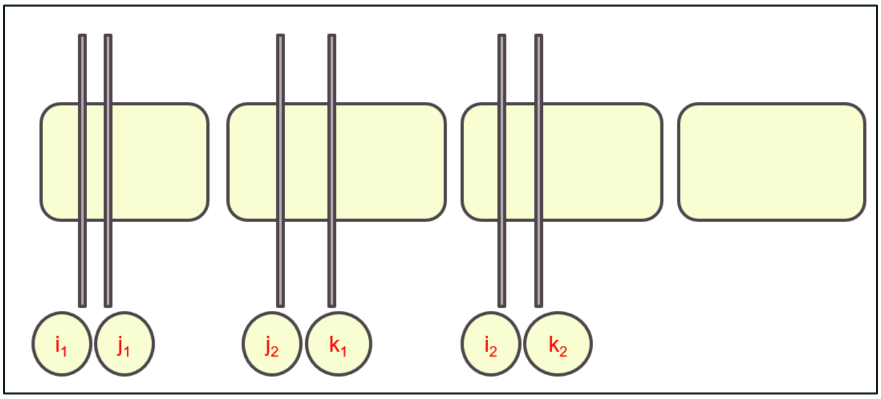

Figure 12: case 1 for Core Protocol

(Case 2)

If we are in case 2 (3,3,0,0 trim for 2 piece) i.e. $i_1 j_1 k_1 \mid i_2 j_2 k_2$ **then**
Execute M1 + M0:

- the cake will be trimmed in the 2nd knife position and rightmost player will receive it
- player who didn't receive trimmed cake will pick a complete piece ;
- (4) take remaining complete ;

End if

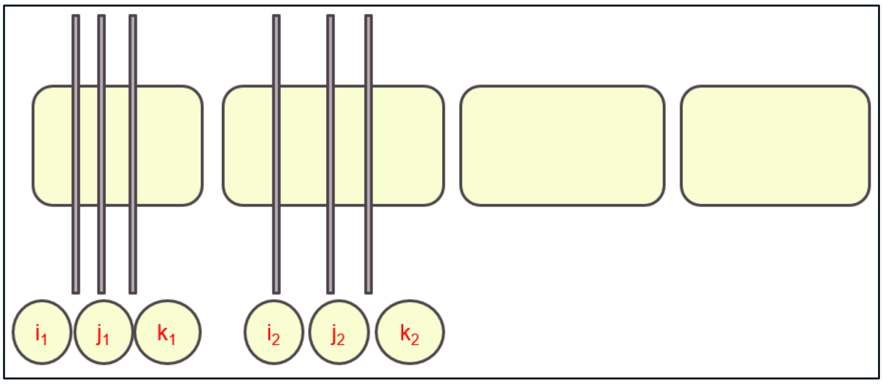

Figure 13: case 2 for Core Protocol

(Case 3)

If we are in case 3 (1,3,2,0 trims for 3 piece) i.e. $i_1 \mid i_2\, j_1 k_1 \mid j_2 k_2$ **then**
If the solo cutter (i) consider the cake as most preferred piece **then**

- (i) will receive the complete version of this piece ;
- rightmost trim player receive the trimmed cake up to second rightmost trim ;

else ((i) view it as second most preferred, says cake1)

- (i) cut cake 1 so it equally valuable to his trim in cake 2 ;
- rightmost trim player receive the trimmed cake up to second rightmost trim ;

End if

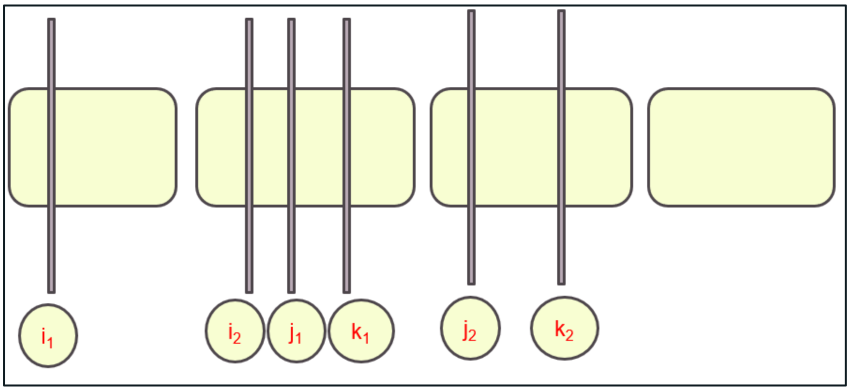

Figure 14: case 3 for Core Protocol

(Case 4)
If we are in case 4 (1,1,2,2 trims for 4 piece) i.e. $i_1 | j_1 | j_2k_1 | i_2k_2$ **then**
If one of the solo cutter (says i or j) view it as most preferred **then**
If both view it as most preferred **then**

- no trim and (i, j) receive the complete version the piece, k take the remaining,
- Then (4) take the remaining;

else (only one of them view like this, says i)
execute M2 + M0:

- (j) re-trim his most preferred to his second most preferred piece (says i | j'k | k |) for cake with 2 trim (cake 2);
- rightmost trim player receive the trimmed cake up to the leftmost trim;
- remaining complete give to the remaining player, Then (4) take the remaining complete cake;

End if

else (one of solo cutter view cake with muti-trim as most preferred)
If only one solo cutter view it that way **then excute M2**
else (Both solo cutter view it that way)

- he re-locate the trim so that it is equal to his 2nd most preferred piece instead, then this cake 3,4 is assigned to rightmost trimmer with the trim at the 2nd rightmost trim;

End if
End if

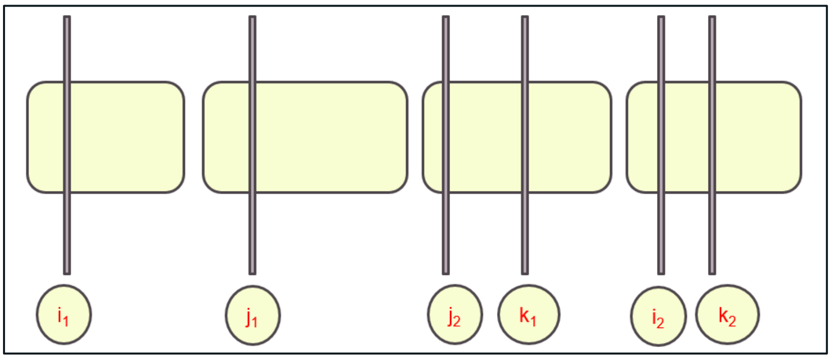

Figure 15: case 4 for Core Protocol

(Case 5)
If we are in case 5 (1,1,1,3 trims for 4 piece) i.e. $i_1 | j_1 | k_1 | i_2 j_2k_2$ **then**
If the solo cutter (says i or j or k) view it as most preferred **then**
no trim for his piece and receive the complete version the piece;
else (solo cutter says j view cake 4 as most preferred)

Execute M2 + M0:

- (j) re-allocate the trim so that it is equal to his 2nd most preferred piece instead, then this cake 4 is assigned to rightmost trimmer with the trim at the 2nd rightmost trim;
- (j)'s 2nd most preferred cake give to the unallocated;

End if

Then (4) take the remaining;

End if

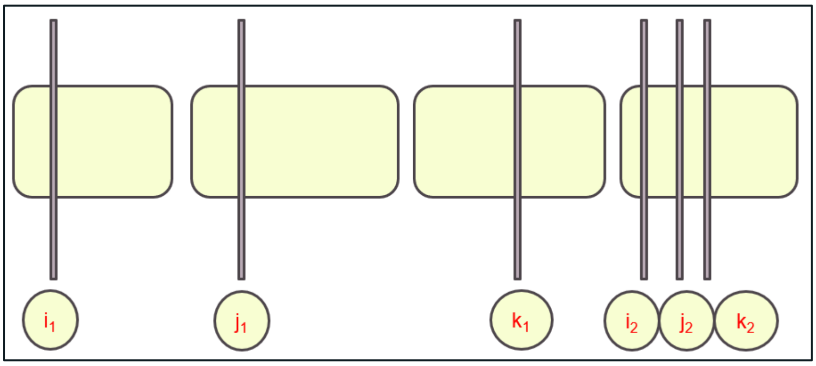

Figure 16: case 5 for Core Protocol

Return Envy-free allocation & possibly unallocated piece

4) Envy-freeness

As shown above, the cutter will always receive a complete cake while other players will at most receive a complete. As a result, the cutter will dominate or ordinal with the other 3 players.

3.4 Permutation Protocol

The Core Protocol requires repeating iteration to achieve double domination.
However, it may result that the cutter keeps dominating the same player, Permutation Protocol aims to achieve Double Domination with higher efficiency (i.e. fewer iterations)

1) Brief Introduction of Permutation Protocol

Basis of the Permutation Protocol:

Input:	Remaining cake from the output of Core Protocol, say (4) is the cutter and (1) receive the significant piece
Output:	Allocation that each player gets a piece equal to the value he trimmed to in the Core Protocol and (2) or (3) receive the significant piece

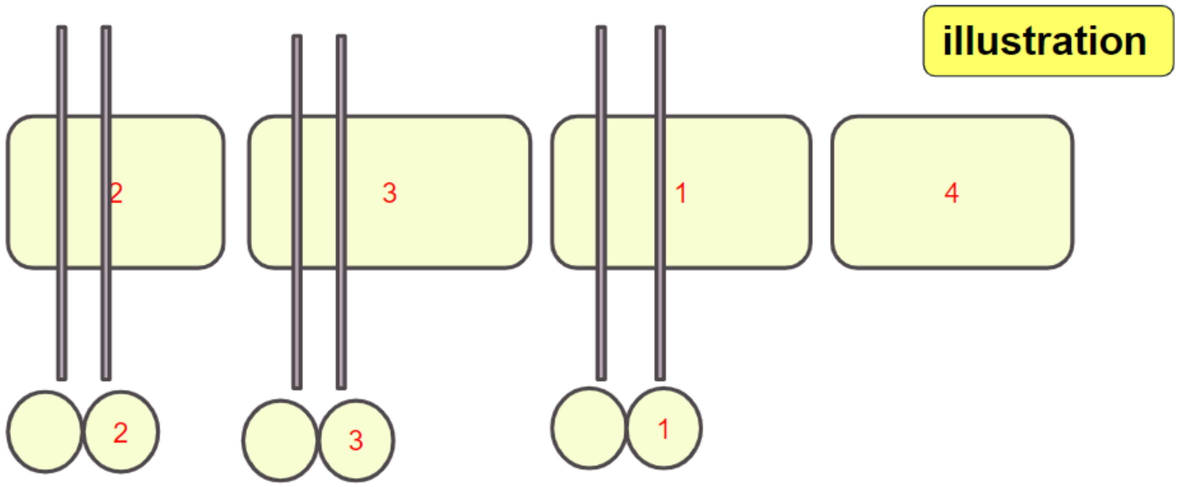

Figure 17: Illustration of the set-up of Permutation Protocol

<u>Terminology:</u>

1. Significant piece: if (i) receive a significant piece (says X_i), meaning that X_i is his 1st preferred piece which is trimmed, therefore (i) was the rightest trimmer in X_i during Core Protocol
2. Competition: if (i) was completing with someone, meaning that (i)'s allocation has had another player's trim mark during Core Protocol
3. Bonus: The additional cake player (i) received from the cake between the trimmer's trim mark and the receiver's trim mark, denoted as γ

2) Details for Permutation Protocol Algorithm

If (2) competed with some player and receive the trimmed cake (i.e. he is rightmost trimmer) **then**

<u>(Case 1)</u>

If the second rightmost trim on (2)'s piece is (1) **then**

- Exchange cakes (X_1) & (X_2) for players (1) & (2)

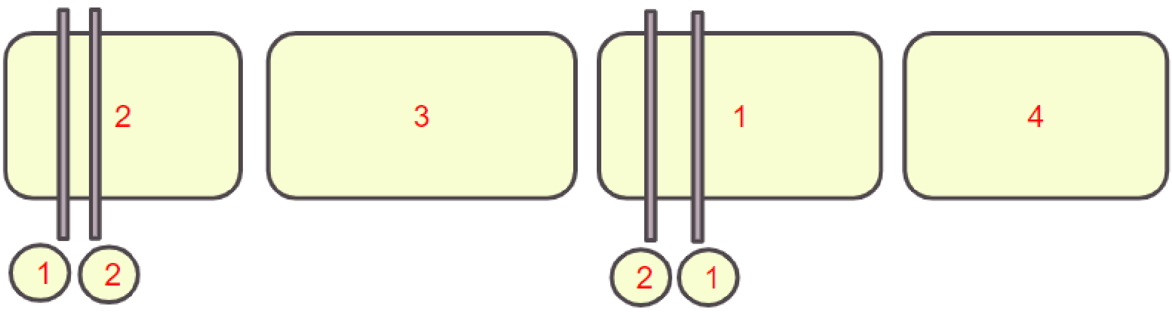

Figure 18: case 1 of Permutation Protocol

<u>(Case 2)</u>

Else if the second rightmost trim on (2)'s piece is (3) **then**

- cake (X_2) give to player (3)

- cake (X_1') (trimmed) give to player (2)

- player (1) choose between cakes (X_3) and (X_4) (both cakes are complete)

- player (4) receive the remaining (complete piece)

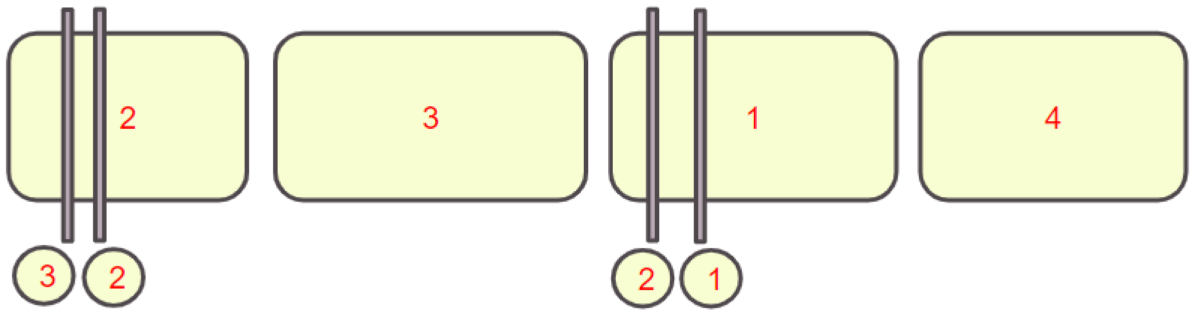

Figure 19: case 2 of Permutation Protocol

(Case 3)
Else if (2) received a complete piece and he did not compete with any player **then**
If his piece have a trim mark by (1) (i.e. (2) hold (1)'s 2^{nd} most preferred piece) **then** permute (1) & (2)

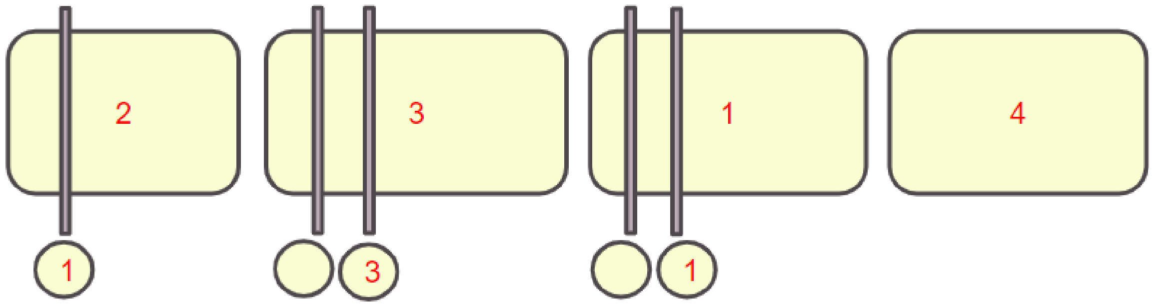

Figure 20: case 3 of Permutation Protocol

(Case 4)
Else if (2) received a complete piece and he is not competing with some player **then**
If (4) has a trim mark by (1) (i.e. (4) hold (1)'s 2^{nd} most preferred piece) **then** permute in the following order

- cake (X_2) -> player (4)
- cake (X_4) -> player (1)
- cake (X_1) -> player (2)

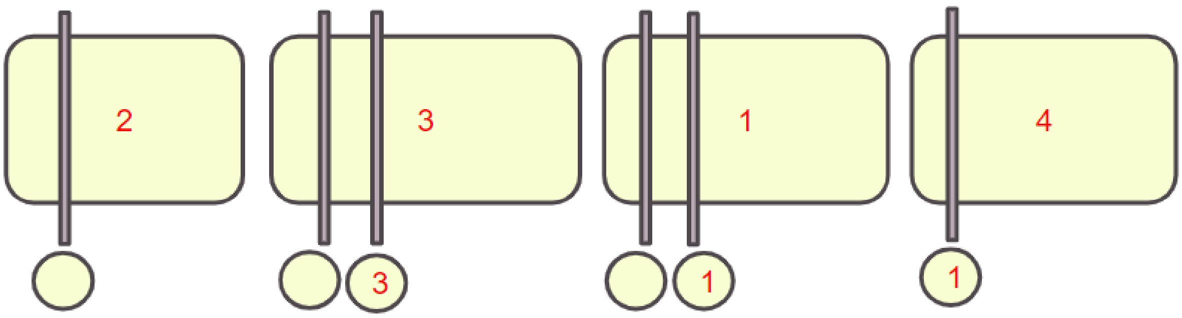

Figure 21: case 4 of Permutation Protocol

(Case 5)
Else if (2) received a complete piece and he did not compete with some player **then**
If (X_3) is complete while it has a trim mark by (1) (i.e. (3) hold (1)'s 2^{nd} most preferred piece) and (3) is indifferent between two pieces among his top 3 pieces **(guaranteed to happen after core protocol) … [Appendix 1]**
then permute in the following order

- cake (X_3) -> player (1)

- cake (X_1) -> player (2)
- player (3) choose between cake (X_2) and (X_4)
- player (4) get the remaining complete piece

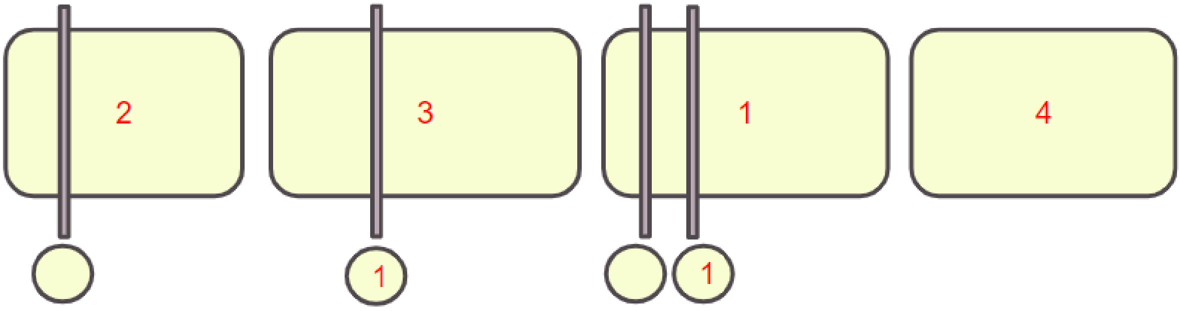

Figure 22: case 5 of Permutation Protocol

End if

3) Domination Target

As shown above, player (1)'s allocation will be exchanged from the significant piece to another complete piece or his 2nd most preferred piece. Therefore, the cutter will change from dominating (1) to ordinal (1) and thus dominating other players instead. As a result, the player who has the significant piece will "be balanced" such that he doesn't have the significant piece anymore, hence his bonus from the significant piece is lowered.

3.5 Overall Protocol

1) Details for Overall Algorithm

i = 0;
While (

- Some player (i) does not dominate two players; and
- There exists some unallocated piece

) do

Run Core Protocol 4 times on the unallocated piece with (i) as the cutter [(i) will dominate one player after two iterations by "Single-Domination Protocol lemma" (see section 4.3 for details and proof)]

If there exists a player (say j) get the significant piece in each of the 4 iterations **then**

Identify which iteration that the non-cutter receive less bonus than the sum of the bonuses in the other iteration
(e.g. says player (3) received bonus in iteration 2 less than sum of his bonus in all other iteration, i.e. $\gamma_3^2 < \gamma_3^1 + \gamma_3^3 + \gamma_3^4$)

Run Permutation Protocol on this iteration with (i) as a cutter, (j) get the significant piece

end if

Run Core Protocol for the unallocated residual if there is any

i += 1

End while

If there is some unallocated cake then

Run Post Double Domination Protocol on the remaining cake

End if

Return envy-free complete allocation

2) Envy-Freeness

By running the Core Protocol 4 times, (i) will be dominating at least one player.
If (i) was dominating the same player (j) for all 4 times (i.e. (j) received the significant piece every time), we permute his allocation for iteration that "damage" his allocation the least to keep the domination relationship strong and applied to two players other than the cutter.

3) Boundedness

3.1 Number of cuts

The maximum number of cuts required for each protocol:

Protocol	Maximum number of cut each iteration
Core Protocol	10
Post Double Domination Protocol	3

Table 4: Count of the number of cuts for different protocols

The maximum total number of cuts required is:

$$10 * 5(iteration) * 4\,(players) + 3 = 203\,(cuts)$$

3.2 Number of Queries

Core protocol for each iteration:

Stage	Number of queries per player	Number of player	Sub-total
Cutter is asked for the value of the whole cake	1	1	1
Cutter cut the cake into 4 equal pieces	3	1	3 =(3*1)
Non-cutters are asked to value the piece	4	3	12 =(4*3)
Non-cutters are asked to trim their 1st and 2nd most preferred pieces	2	3	6 (=2*3)
Re-trim or Cut & Choose (M2)	1 or 1	2 or 4	4 (=max(1*4,1*2)
Total			26

Table 5: Count of the number of quiries for different protocols

Permutation Protocol each iteraion:

=3 (queries)

Post-Domination Protocol:
=16 (queries)

The maximum total number of cuts required is:

$$(26 * 5(iteration) + 3 * 4 (players)) * 4(players) + 16 = 203 (cuts)$$

Note: We run the permutation and core protocol for 4 players to achieve double domination.

3.6 Strategic Prove of Four-person Allocation Protocol

As the players of the allocation are assumed to be risk averse, we will observe that if a player tells any "lie" in different stages of the protocol will lead to a loss in value received by the player in the worst situation of different stages, the player will not conduct the new strategy and thus the protocol is strategic-proof.
The following protocols are in reverse order as stages, at last, will prevent players from performing some strategies at an earlier stage.

3.6.1 Post-Double Domination Protocol
During the Post-Double Domination Protocol, the domination relationship is not confirmed as it is affected by the choice of other players.
In cases 1 and 2, the worst situation is that the player does not receive any cake, i.e. indifference to the previous stage.
In case 3, the cutter will not tell "lie" as he/she is the last to choose. Also, the choosers will also choose the largest piece to maximize their value received.
Therefore, this protocol is strategic-proof.

3.6.2 Permutation Protocol
The permutation protocol does not involve strategies that the players can conduct, as all the procedures are done following the result of the core protocol which cannot be controlled by any one of the players by itself.
Also, the Protocol will always return a value to every player that is lower or equal to the result of the previous stage.
Therefore, this protocol is strategic-proof.

Remark: The player may receive a cake that was allocated to other players originally which states that lowering the value of the cake received by others in the Core Protocol has a chance of bringing a loss to the player.

3.6.1 Core Protocol
Consider single strategies that can be conducted to tell "lie",
i.e. moving the knife to the right/left and exchanging the preference over the pieces.
Remark: The cases are constructed with the aim of producing the worst scenario for the player who chooses to tell "lie".

Case 1:The player moves the knife to the right

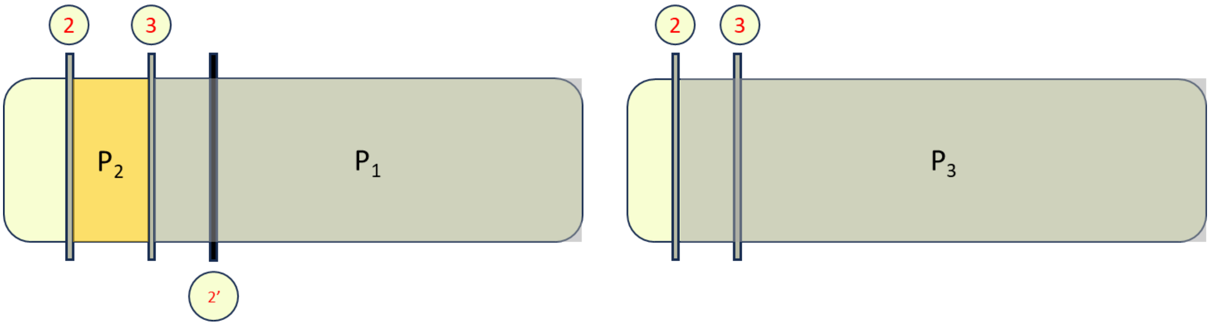

Figure 23: case 1 of lie about knife position for Core Protocol

Original allocation: Player 2: (P_1+P_2), Player 3: (P_3)
New allocation: Player 2: (P_1), Player 3: (P_3)
Player 2 is worse off.

Case 2: The player moves the knife to the left
Assume that $V_2(P_4) > V_2(P_6)$ and $V_3(P_4+P_5) > V_3(P_6) > V_3(P_4)$,

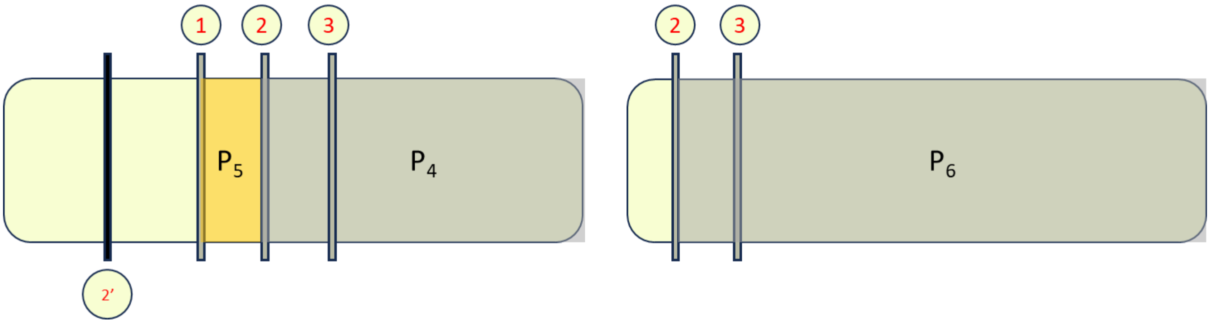

Figure 24: case 2 of lie about knife position for Core Protocol

Original allocation: Player 2: (P_4), Player 3: (P_6)
New allocation: Player 2: (P_6), Player 3: (P_4+P_5)
Player 2 is worse off.

Case 3: The player exchange his preference of 1st and 2nd piece
Suppose the following is in case 5 of the protocol,
i.e. there are 3 cuts on the same cake while there is one cut on the other three cakes.
Assume that *Cake 1* is originally the 2nd preferred while *Cake 2* is the 1st preferred for player 1.
Therefore, $V_1(P_8) > V_1(P_7)$.

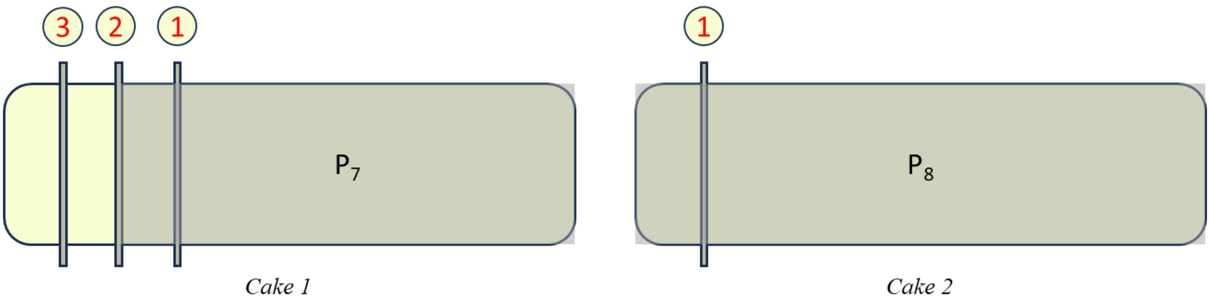

Figure 25: case 3 of lie about preference for Core Protocol

Original allocation: Player 1: (P_8)
New allocation: Player 1: (P_7)
Player 1 is worse off.

Case 4: The player exchange preference to the 3rd most preferred piece and 4th most preferred piece

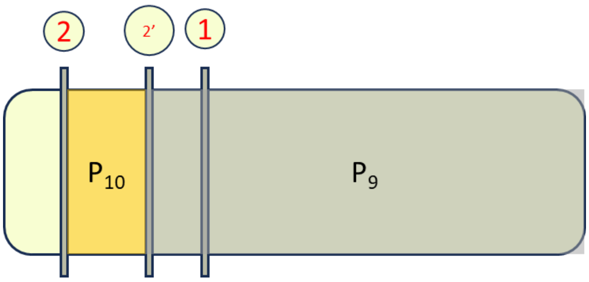

Figure 26: case 4 of lie about preference for Core Protocol

The cake received by player 1 is reduced from (P_9+P_{10}) to (P_9) which might lead to loss in the stage of Permutation Protocol for Player 2.
Therefore, Player 2 may be worse off.

Case 5: The player exchanges preference to the 3rd most preferred piece and 1st/2nd most preferred piece
After the strategy is performed, the player will randomly put a cut on the 3rd most preferred piece, and there is a chance that a partial part of the 3rd most preferred piece is given to the player. Originally either the complete piece of the 3rd most preferred piece or another piece with a value of larger than the 3rd most preferred piece is given to the player.
Therefore the player is worse off.

Case 6: A mixed strategy of the above
In cases 1 to 5, we can always find a situation where after switching the preference of cake, it falls into one of the scenarios of case 1 or 2.

In conclusion, the core protocol is strategic-proof.

As the 3 protocols are strategic-proof separately, therefore combining them as the core protocol will still preserve strategic-proof.

3.7 Pareto Optimal

The Overall Protocol is **not** Pareto optimal.

During the Permutation Protocol, the pieces are reallocated to different players. The bonus allocated to the players in the Core Protocol might be reduced due to the exchange of pieces. Comparing the final allocation that follows the Overall Protocol and the allocation that combines the result before the Permutation Protocol and residue allocated the same as that in the Overall Protocol, the second one will give a higher individual value without reducing other players' value.
Therefore, the Overall Protocol is not Pareto efficient.

4. Major Findings

4.1 Double Domination Lemma

Suppose we have a bound protocol that takes (i) and an unallocated piece as input, and outputs a partial envy-free allocation such that(i) dominates 2 other players,
Then we can extend this to 4 players envy-free bounded protocol

Proof: As demonstrated in Post Double Domination Protocol

4.2 Core Protocol Lemma

For n=4, there exists a discrete and bounded protocol which returns partial envy-free allocation in which one player cuts the cake into 4 equally preferred pieces and the cutter as well as at least one player gets one of those complete pieces

Proof: As demonstrated in Core Protocol

4.3 Single Domination Lemma

For n = 4, there exists a discrete and bounded protocol which returns an envy-free partial allocation in which one player dominates another player

Proof:
Assume player 4 is the cutter in the core protocol, let β_1, β_2 be the residue and β_1 be the residue of the significant piece in the valuation of player 4. (The following valuation is from the perspective of player 4)
<u>Stage 1:</u> In the core protocol, at most 2 of the cake is trimmed and the residue is at most ½ of the total cake in total. Also as $\beta_1 > \beta_2$, the residue has an upper bound of min($2\beta_1$, ½).
<u>Stage 2:</u> Consider running the Core Protocol again, the residue is again at most ½ of the residue in stage 1, i.e. residue< $2\beta_1 * 1/4 * 2 = \beta_1$
Therefore, if the residue in stage 2 is given to the person who gets the significant piece, the cutter(player 4) will still not envy the person, and achieve single domination.

4.4 Permutation Lemma

There exists a discrete and bounded protocol for 4 players that returns a partial envy-free allocation in which one player dominates two other players

Proof: As demonstrated in Permutation Protocol

5. Conclusion

The main idea of the algorithm is to exploit the bonus cake with permutation so that one player can dominate other players and thus achieve envy-freeness. Therefore, there are 3 different sub-algorithms to do so, namely core protocol, permutation protocol, and post-double domination protocol.

This Protocol provides some useful insight into cutting the cake fairly with a bounded number of cuts and queries, such as partial cut, domination and ordinal, permutation, etc. We showed that by combining those techniques, we can expand the algorithm from a 3-person to a 4-person case, we can then expect that it is possible to generalise the idea to a 5 or even n-person fair allocation algorithm.

We can also observe that the protocol is good overall since it can achieve envy-freeness and strategic proof while boundedness is accomplished. However, it is not perfect, we showed that the algorithm is not Pareto optimal as it may decrease some players' utility during the permutation protocol.

Another area for improvement could be having 584 as the number of queries might be too high for 4 players, considering on average each player will be asked 146 times at most during the algorithm, which might be too demanding in practice. Therefore, one may seek some other protocol with higher efficiency.

6. References

This material is mainly based on the result from the paper **[A Discrete and Bounded Envy-free Cake Cutting Protocol for Four Agents]** by Haris Aziz, Simon Mackenzie (2016). Additional material cited is as follows.

Reference list:

[1]: F. E. Su. Rental harmony: Sperner's lemma in fair division. American Mathematical Monthly, 10:930–942., 1999.

7. Appendix

[1]: (Prove that player (3) is indifferent between two pieces among his top 3 pieces after Core Protocol, statement used in the permutation protocol)
Assume that player 3 is not indifferent among any of his 3 most preferred pieces.

(a). We want to prove that the most preferred piece of Player 3 is the most significant piece.

The piece received by player 1 is a complete piece. Therefore player 3 does not compete on this piece. By partial envy-freeness achieved in the core protocol, the piece held by players 2 and 4 will also not be the most preferred piece of player 3 before trimming.
It implies that the most preferred piece of Player 3 is the significant piece.

(b). We want to prove that player 3 is allocated with a piece that is equal to his 2nd most preferred piece.

Assume that player 3 has put trim on other pieces such that it equals the 3rd most preferred piece and receives a piece equal to his 3rd most preferred piece. Either player 2 or 4 will get a complete piece, which indicates player 3 does not make a trim on it, also, either player 2 or 4 will get player 3 most preferred piece. This implies player 3 will be envious of either player 2 or 4. Therefore, a contradiction.
Assume player 3 makes trims which equals his 2nd most preferred piece. Therefore, player 3 will guarantee receiving a piece that is equal to the 2nd most preferred.

(c). We want to show that the piece received by player 3 is also the 2nd most preferred piece of player 1.

If the piece of player 2 or 4 was the most preferred piece of player 1, we would not be in this track of the Permutation Protocol.
In sum, we now have the piece received by player 3 as the 2nd most preferred piece of both players 1 and 3. Since, we assume the piece received by player 1 is the significant piece in the Permutation Protocol, and is the 2nd most preferred piece, player 1 should put a proper trim onto this piece. While, player 3 also thinks the piece is the 2nd most preferred piece, therefore player 3 will put a trim on the same piece as player 1. If a piece has 2 trims on it, it cannot be allocated completely to any of the players.
Therefore, it contradicts the statement that player 3 has a different valuation on the 3 pieces.