E0 337: Topics in Advanced Cryptography 6/Sep/23

Lecture 5

Instructor: Dr. Bhavana Kanukurthi, Dr. Chaya Ganesh *Scribe: Shubh Prakash*

1 Welcome to E0 337

1.1 Course Information

Contact information and office hours:

- **Lecturer:** Bhavana Kanukurthi (bhavana@iisc.ac.in), Chaya Ganesh (chaya@iisc.ac.in).
- **Presenter:** Girisha B Shankar (girishabs@iisc.ac.in).

1.2 Lecture 5 topics

The lecture will cover the following topics:

- Fair Division
- Matching
- Transaction Fee Mechanisms

2 Recap

In the last two lectures, mechanism design was introduced and some basic terminology and notation related to it was discussed:

- Had discussed direct and indirect mechanisms and had stated a result which shows their equivalence, called **the revelation principle**
- Had mentioned various kinds of real life settings, where mechanism design is important. For example:
  - Auctions
  - Fair Division
  - Elections
  - Matching
- Had discussed auctions in some detail, talking about both *English auctions* and *sealed bid auctions*
- Saw another classification: *First price* and *second price* auctions and how the strategies for the parties differ in these 2 kinds of auctions
- These kinds of Auctions are examples of **Bayesian games**
- The notion of incentive compatibility was also discussed

3 Fair Division

In this section, we consider only divisions of items which are *continuous* and *homogeneous*.

3.1 Cake Cutting Problem- 2 Parties

This is a problem we had seen in the very first example of mechanism design.

Problem Statement- There is a cake(continuous, homogeneous). There is a parent and 2 kids. Parent wants to divide the cake into 2 parts in such a way that both kids are satisfied. What is a mechanism to achieve this.

Solution- Pick one of the kids at random. Say kid 1. Ask him to divide the cake into 2 pieces as he wants. Ask kid 2 to pick one of the 2 pieces whichever he prefers. Kid 1 gets the piece which was left.

Reason why this works- Suppose total utility of cake is 100 for each player and it is additive.

Kid 1 will divide the cake in such a way that *both pieces have equal utility according to his utility function-both 50-50*(if not, then he risks kid 2 choosing the piece of higher utility according to kid 1, leaving kid 1 with the piece which he deems having lesser utility)

Kid 2 will certainly choose the piece which it deems to have higher utility.

Thus Kid 2 will get a piece which has utility ≥ 50 according to it
Kid 1 anyway gets a piece of utility 50 according to it.

Kid 1 was indifferent between the 2 pieces so of course he is satisfied.
Kid 2 is satisfied because it got to pick the piece which had higher utility according to it.

This was an elementary example of fair division

3.2 More general fair division

Without loss of generality, we assume that the item to be divided is a 1-D interval, call it M.
Let parties be $N = \{1, 2, \cdots, n\}$
Let $X = \{X_1, X_2, \cdots, X_n\}$ be the pieces of M which are pairwise disjoint and have union M
That is,

$$\bigcup_{i=1}^{n} X_i = M$$

$$X_i \cap X_j = \phi \, \forall i \neq j$$

Each player has valuations for all the pieces. Let player i have valuation function v_i.
v_i is a map from X to $\mathbb{R}$
The valuation function is additive, that is:

$$v_i(X_m \cup X_n) = v_i(X_m) + v_i(X_n)$$

3.2.1 Cake Cutting problem-2 parties

Using this notation, we can represent the cake division problem in the form of a table:

	X_1	X_2
1	30	70
2	50	50

From the table, it is clear that player 2 divided the cake(as both pieces are 50-50 for it)
Player 1 chose the piece of higher utility for it. So, it chose X_2. Player 1 got X_1.

As discussed, both players are happy. This is because both got the best piece they could have got and each got a piece of utility ≥ 50.

3.2.2 Cake Cutting problem- Extending to 3 parties

Example 1. *Consider the following table:*

	X_1	X_2	X_3
A	90	80	130
B	100	100	100
C	60	90	150

We can see that player B divided the cake. Naturally, he divided the cake in a way that all 3 pieces have equal utilities according to it.

B is indifferent between the pieces. But A and C both prefer X_3 maximum.
This is a problem because we want any fair division mechanism to satisfy **proportionate property**:

Definition 1 (Proportionate)**.** For all parties $i \in N$, let Y_i be the subset of M that player i ends up with. Division mechanism is said to be proportionate if:

$$v_i(Y_i) \geq \frac{v_i(M)}{n}$$

In the above example, we can see $v_i(M) = 300$. Thus, all parties should get piece of valuation atleast 100. But, it doesn't seem possible as whoever of A and C doesn't get X_3, he will get something of valuation less than 100.

Solution to this problem:
Each party is asked what it wants. A replies $(X_3, 130)$, C replies $(X_3, 150)$, B is indifferent.
So, clearly B can be given X_1 or X_2 randomly, say X_2.
So X_1, X_3 are left.
Merge these pieces together, that is new $M = X_1 \cup X_2$
Use this new M as the new cake for 2 player cake division with the parties A and C.
Now clearly, as valuations left is 220 for A and 210 for B, both players are guaranteed to get more than 100 valuation, which is what was required.

3.2.3 Extension to 4 parties and more

Example 2. *Consider the following table for fair division among 4 parties:*

	X_1	X_2	X_3	X_4
A	21	18	34	27
B	30	21	23	26
C	22	21	42	15
D	25	25	25	25

Clearly D divided the cake. So it is indifferent.
1 approach: Proceed as before:
A says $(X_3, 34)$
B says $(X_1, 30)$
C says $(X_3, 42)$

So, clearly no one has X_2 as their best preference. So give X_2 to D. No one except B has X_1 as first preference.
Now X_3, X_4 are left for A and C. Both want X_3 maximum, but for player A, even X_4 is valued at ≥ 25. But for C, X_4 is valued at 15 which is unacceptable.
So we can give X_4 to A and X_3 to C.

This indeed satisfies proportionate property. However, it has a problem.

Definition 2 (Envyfree-ness)**.** A fair division mechanism is said to be *envy free* if for all $i \in N$:

$$v_i(X_i) \geq v_i(X_j)$$

Clearly, the above solution **is not envy free** for A. This is because A got X_4 which had a valuation less than X_3 for it. But it didn't get X_3

So, this kind of naive solution may not work.
In general, one may need to potentially run the cake cutting for k-1 times if total parties are k. Basically, one person divides and he is given the least wanted piece. Then the $k-1$ parties do the cake cutting game on whatever is left. Again the divider gets the least wanted piece. The remaining $k-2$ participate in another cake cutting game and so on.

After atmost $k-1$ cake cutting games, we get the final division which will surely be proportionate and envy free.

Definition 3 (Equitable)**.** Suppose $Y_1, Y_2, \cdots, Y_n$ are the pieces that players $1, 2, \cdots, n$ get in the mechanism. A fair division mechanism is said to be equitable if:

$$v_1(Y_1) = v_2(Y_2) = \cdots, = v_n(Y_n)$$

This property is hard to achieve and it may not be possible to achieve this in every case.

4 Matching:

4.1 Some definitions

Definition 4 (Matching in a graph)**.** In an undirected graph $G = (V, E)$, a matching is a subset M of E which is such that no 2 edges in the subset share a common vertex. That is :

$$\{u, v\} \cap \{x, y\} = \phi \forall (u, v), (x, y) \in M$$

Definition 5 (Bipartite Graph)**.** A graph $G = (V, E)$ is said to be bipartite if V can be divided into V_1 and V_2 such that $V_1 \cup V_2 = V$ and there are no edges between vertices in V_1 and no edges between vertices in V_2

Thus, in bipartite graphs, matching is basically joining each element of V_1 with a unique element of V_2(if possible) and vice versa. This is the case which is mostly considered.
A bipartite graph is called complete if for any $v_1 \in V_1$ and any $v_2 \in V_2$, $(v_1, v_2) \in E$

We will consider matching in such graphs. We interpret the edges to mean preferences.
Each edge (v_1, v_2) will have 2 weights on it, one weight gives preference of v_2 for v_1 and other weight gives preference of v_1 for v_2

4.2 The setting:

$$A = a_1, a_2, \cdots, a_n$$
$$B = b_1, b_2, \cdots, b_m$$

We can interpret A and B as V_1 and V_2 of a complete bipartite graph.

Each a_i has preference order of all the b_j and vice versa (preferences given by the edge weights)

Objective is to create a matching between A and B which is **stable**

Definition 6 (Stable Matching)**.** Let A and B and preference orders be as above. A matching M of A and B is said to be *stable* if for any $a \in A$ and any $b \in B$ either a prefers its current partner in M atleast as much as it prefers b or b prefers its current partner in M atleast as much as it prefers a

Example 3. *Say* $A = \{x, y, z\}$
$B = \{a, b, c\}$
Preference orders:

- x: $a > b > c$
- y: $b > c > a$
- z: $a > c > b$
- a: $y > z > x$
- b: $y > z > x$
- c: $x > y > z$

Consider $M = \{(x, a), (y, b), (z, c)\}$ This is not a stable matching as z prefers a more than c and a prefers z more than x.

On the other hand, consider $M = \{(x, c), (y, b), (z, a)\}$
It is easy to see that this is a stable matching

Designing mechanism for this while making sure it happens without any parties preferences becoming leaked is an important use case for cryptography.

5 Transaction Fee Mechanisms

5.1 Background

Such mechanisms are useful in blockchains, specifically in cryptocurrency like bitcoin.
We will stick to bitcoin.
The miners, which mine the block, in addition to getting bitcoin for mining the block successfully and solving the problem by utilizing their computational resources, also charge a small transaction fee to the people who want their transactions to be recorded in that block.

Transactions fees are essential to eliminate competition and for the miner to decide in a logical way which transactions to include in the current block (usually there is a size limit to the number of transactions that can be recorded on a block).

Usually the way it works is that everyone who wants their transaction to be included in the block bids a transaction fee that they are willing to pay to the miner.
So it is like an auction.

Note- In this setting miner is also a player. He wants to maximise his earnings.

5.2 The setting:

We assume there is 1 miner who has successfully solved the problem and he has the block. He wants to add transactions to the block in such a way that he maximises his transaction fee.

Suppose n parties want their transaction to be included. Let us say there is space of only $k < n$ transactions in a block.
We can think of 2 cases:

5.2.1 First price auction

Here it is extremely straightforward. Miner just includes the transactions corresponding to the k highest bids.

5.2.2 Second Price Auction

This is a more interesting case. Consider that $n = 6$ with the following bids:

$$10,\ 9,\ 8,\ 6,\ 5,\ 4$$

Suppose $k = 3$
If miner is honest, he will include the transactions corresponding to the largest 3 bids and will get $3 \times 6 = 18$ as the transaction fees. However miner can try to be clever. He can include a fake transaction of bid 7. Then, it will still include the same 3 transactions, but will get $3 \times 7 = 21$ in this case.

Thus, such issues have to be kept in mind when designing mechanisms.

5.2.3 Ideal Requirements

Ideally we would want a mechanism to incentivize the following:

- All the people who want their transactions included are truthful and bid according to their valuation
- Miner is honest and doesn't include fake transactions for example
- No collusion happens between the miner and the people or between people

  It turns out that all 3 are not possible together. The only mechanisms which enable all 3 to happen together are the ones where miner gets no transaction fees.