CS 6002: Selected Areas of Mechanism Design **Jan-Apr 2025**

Lecture 8: Indivisible Item Allocation

Lecturer: Swaprava Nath *Scribe(s): Rishabh*

Disclaimer: *These notes aggregate content from several texts and have not been subjected to the usual scrutiny deserved by formal publications. If you find errors, please bring to the notice of the Instructor.*

8.1 Generalizing the Even-Paz Algorithm

In the previous lecture, we had seen how the Even-Paz algorithm works when n is a power of 2. But what if this isn't the case. We make a small modification, instead of each agent marking the point where the valuation is half, they mark where valuation is $\frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor$ (Note that this is $\frac{1}{2}$ when n is even, slightly less than $\frac{1}{2}$ when n is odd). We divide the cake based on the $\left\lfloor\frac{n}{2}\right\rfloor^{th}$ division from the left, and recurse on the left piece with the left $\left\lfloor\frac{n}{2}\right\rfloor$ agents, and the right piece with the remaining agents.

Theorem 8.1 *The Even-Paz algorithm (when n need not be a power of 2) is proportional*

Proof: The proof is by induction on the number of times an agent participates in game divisons.

Base Case: If a player participates in a game division only once before they get their share, either $n = 2$, or $n = 3$ and the agent performed the leftmost cut. When $n = 2$ this is clearly proportional as it just becomes the "I cut, you choose" allocation. When $n = 3$, the leftmost agent gets $\frac{1}{3}\left\lfloor\frac{3}{2}\right\rfloor = \frac{1}{3}$ of the cake according to their valuation, hence this is proportional.

Induction Step: For all the $\left\lfloor\frac{n}{2}\right\rfloor$ agents on the left, the value of the left half is at least $\frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor$. From the induction assumption all these agents will get valuation at least $\frac{1}{\left\lfloor\frac{n}{2}\right\rfloor}$ of the left half. So overall their valuation will be at least $\frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor \times \frac{1}{\left\lfloor\frac{n}{2}\right\rfloor} = \frac{1}{n}$. Every agent on the right values the left half less than $\frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor$ hence values the right half more than $1 - \frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor = \frac{n-\left\lfloor\frac{n}{2}\right\rfloor}{n}$. From the induction assumption all these agents will get valuation at least $\frac{1}{n-\left\lfloor\frac{n}{2}\right\rfloor}$ of the right half. So overall their valuation will be at least $\frac{n-\left\lfloor\frac{n}{2}\right\rfloor}{n} \times \frac{1}{n-\left\lfloor\frac{n}{2}\right\rfloor} = \frac{1}{n}$.

■

8.2 Envy Free Cake Cutting

8.2.1 Envy Free Division for 3 Agents

When there are 2 agents, envy free cake cutting can be done by the "I cut, you choose" protocol. However even for 3 agents, the algorithm is a bit non-trivial. Assume there are 3 agents, A, B, C.

Phase 1:

- A cuts the cake into 3 pieces such that they value all pieces equally ($\frac{1}{3}$ each)

- B identifies their first and second most favourite piece, trims their favourite piece such that they value it as much as their second favourite piece
- Ignoring the trimmed portion, C picks their favourite piece, followed by B, and followed by A
- We make sure that A doesn't get the trimmed piece, if C did not pick the trimmed piece we make B pick it (which is fine as B values trimmed piece at least as much as all other pieces)

Note that at this stage (ignoring the trimmed part) all agents got their favourite piece. C as they had the first pick, B as they had 2 equally favourite pieces after trimming so after C's pick they can still choose their favourite, and A valued all pieces equally before trimming, and doesn't get the trimmed piece.

Phase 2: dividing the trimmed part

We know that either B or C has picked the trimmed piece. Let $T \in \{B, C\}$ be the agent who picked the trimmed piece and $T^c = \{B, C\} \setminus T$ be the agent among B, C who didn't pick the trimmed piece.

- T^c cuts the trimmed part into 3 pieces such that they value all equally
- All agents pick their favourite piece, in the order T, A, T^c

Note that T, T^c got their favourite piece in phase 2, T as they got the first pick, and T^c as they valued all the pieces equally.

Claim 8.2 *The above algorithm is envy free.*

Proof: T, T^c don't envy anyone, as they got their favourite piece in both phases. A doesn't envy T^c as they got to pick their favourite piece in phase 1, and picked before T^c in phase 2. A doesn't envy T, as T got the trimmed piece plus only a portion of the trimmed part, whereas A values their piece as much as the trimmed piece plus the whole trimmed part, as A valued all pieces equally before trimming. ■

8.2.2 Envy Free Divsion for n Agents

Procaccia (2009) proved that for n agents, we will need at least $\Omega(n^2)$ queries. *Aziz and Mckenzie (FOCS 2016)* discovered an algorithm for n agents, however it's complexity was extremely large, $O\left(n^{n^{n^{n^{n^n}}}}\right)$. Not much is known between these 2 bounds of complexity.

8.3 Indivisible Item Allocation

Examples: allocation of hostel rooms, inheritances, vaccines

	A	B	C	D	E
1	1	2	5	1	1
2	1	0	5	1	1
3	1	1	5	1	1

Table 8.1: Example 1

Here's the setup. We have a set of agents $N = \{1, 2, \ldots, n\}$, and a set of items $M = \{1, 2, \ldots, m\}$. We seek to distribute the items among the agents such that each item goes to exactly one agent. The agents each have their own valuation functions, $v_i : 2^M \to \mathbb{R}$ which corresponds to how much they value a set of items. We make a simplifying assumption that all valuations are additive i.e. the valuation of a set is the sum of valuations of the individual items. For example in 8.1, $v_i(\{A, B, D\}) = v_i(A) + v_i(B) + v_i(D)$.

An allocation $A = (A_1, A_2, \ldots, A_n)$ must satisfy

- $A_i \cap A_j = \phi \ \forall i \neq j$ (no item can be allocated to 2 agents)
- $\bigcup\limits_{i \in N} A_i = M$ (no wastage assumption, all items are allocated to an agent)

An example allocation for Table 8.1 is $A = (\{A, B\}, \{C\}, \{D, E\})$.

Definition 8.3 *An allocation is envy free (EF) when each agent prefers their bundle over any other agent's bundle.* $v_i(A_i) \geq v_i(A_j) \ \forall i, j$

	A	B	C
1	2	1	2
2	1	1	3

Table 8.2: Example 2

In Table 8.2, an example of an envy free allocation is $A = (\{A, B\}, \{C\})$. $v_1(\{A, B\}) = 3 > v_1(\{C\}) = 2$, $v_2(\{C\}) = 3 > v_2(\{A, B\}) = 2$. But note that there's no envy free allocation in Table 8.1, and envy free indvisible item allocation is not guaranteed to exist.

8.3.1 Relaxation to Envy Freeness (Budish 2011)

A new criteria of allocation was made, where an allocation is envy free upto 1 item (EF1)if envy can be removed by removing an item from the envied bundle (note that the item isn't literally removed from the allocation, just for the sake of calculating valuations). Or mathematically, $v_i(A_i) < v_i(A_j)$ is possible but $v_i(A_i) \geq v_i(A_j \setminus \{x_j\}), x_j \in A_j$.

Definition 8.4 *An allocation* $A = (A_1, A_2, \ldots, A_n)$ *is envy free upto 1 item (EF1) if for every pair of agents* $i, j \in N$, $\exists x_j \in A_j$ *s.t* $v_i(A_i) \geq v_i(A_j \setminus \{x_j\})$

In Table 8.1, $A = (\{A, B\}, \{C\}, \{D, E\})$ is EF1.

8.3.2 Round Robin Algorithm

The following is an algorithm to find an EF1 allocation, assuming all allocations are additive.

- Assign a fixed priority order to all the agents
- In this order, each agent picks their favourite unallocated item, and the item is allocated to that agent

- The algorithm cycles back to the highest priority agent if no agents are remaining as many times as it required and a new round is started, until all items are allocated

Theorem 8.5 *The round robin algorithm is EF1*

Proof: We shall prove for all pairs of agents i, j, agent i doesn't envy agent j upto 1 item.

Case 1: i picks before j in the priority order. In this case for every round, as agent i picks the most valued remaining item, the item picked by i is more valued than the item picked by j, so overall the bundle of i is more valued than the bundle of j. Hence i doesn't envy j, and we don't even need to remove an item.

Case 2: i picks after j in the priority order. In this case, j could pick something more valued than what i picks in every round, as j picks first. But note that what i picks in round k will be more valued than what j picks in round $k+1$, as the latter happens later than the former. Thus we can pair every item j picked in round $k+1$ with every item picked in round k by i, for each pairing, i's pick is valued more. But note that j's first item will not be paired. But we can choose this specific item from j's bundle to remove in order to show EF1.

■

One disadvantage of the round robin algorithm is that even though it always outputs an EF1 allocation, it might not output an EF allocation even if it exists. For example in Table 8.1, if agent 2 has higher priority order, it will output an allocation which is EF1 but not EF.

8.4 Next Topic: EF1 for monotone valuation

We no longer assume additive valuation, instead we assume monotone valuation which means that $v_i(S) \leq v_i(T)\ \forall S \subseteq T \subseteq M$. Essentially, adding new elements to your bundle, cannot decrease your valuation.

Exercise 8.6 *Find an example of monotone valuation where the round robin algorithm doesn't give EF1 allocation.*

In the next class we shall develop an algorithm to give an EF1 allocation for this case, known as the *envy cycle elimination algorithm.* The items are ordered, and we form a directed graph with the agents. This is an *envy graph*, where each agent points to whoever they envy. We allocate the items one by one, and always give the items to a source in the graph (a source always exists if the graph has no cycles). If a cycle forms in the graph, we shift the whole bundles in the direction of the cycle.

Claim 8.7 *The envy cycle elimination algorithm is EF1*

The intuition for proving the claim is inductive, assume that at a stage in the algorithm it is EF1. If after allocating an item, someone envies the agent, we know that without that item no one would envy the agent as the agent was a source (no one envied them).