# Lecture 10

*Lecturer: Ariel D. Procaccia* *Scribe: Yuval Vardi, Yosef Prat and Assaf Weiner*

## 1 Fair Division - An Introduction

In previous lectures we talked about envy-free allocations of divisible (continuous) goods (for example a cake). For divisible goods (a cake) Sue proved:

**Theorem 1 (Sue, 99)** *There always exists an allocation of a cake among players which is envy-free.*

In the last lecture (lecture 9) we saw that it is not the case for indivisible goods.

## 2 The Communication Requirements of Allocations of Indivisible Goods

**Definition 2** *We will denote the set of players (agents) by $N = \{1, ..., n\}$.*

**Definition 3** *We will denote the set of goods (items) for allocation by $G = \{g_1, ..., g_m\}$.*

**Definition 4 (Evaluation Function)** *For each player $i \in N$, we will define an Evaluation Function $V_i : 2^{|G|} \to \mathbb{R}$ that will assign a real value for every subset of items.*

**Definition 5 (Allocation)** *Let $A = (A_1, ..., A_n)$ be an allocation of the goods, such that $\forall i \in N, A_i \subseteq G$ and $\forall i, j \in N, i \neq j, A_i \cap A_j = \emptyset$. According to $A$, player $i$ gets the subset $A_i$.*

**Definition 6** *For players $i, j \in N$ we will define the envy of player $i$ of $j$ as*
$e_{ij} = max\{0, V_i(A_j) - V_i(A_i)\}$.

**Definition 7 (Total Envy)** *We will define the Total Envy as $e(A) = \max_{i,j} e_{ij}$.*

**Theorem 8 (Nisan-Segal, 02)** *Every protocol that finds allocation with minimal total envy must use exponential number of bits of communication in the worst case.[1]*

**Proof**

We assume $|G| = m = 2k$. Let $F$ be a set of evaluation functions such that:

$$\forall V \in F, S \subseteq G, \quad V(S) = \begin{cases} 1 & \text{if } |S| > k \\ 0 & \text{if } |S| < k \\ 1 - V(\bar{S}) & \text{if } |S| = k \end{cases}$$

Therefore, each evaluation function $V$ is defined by the subsets of $G$ of size $k$. In addition, for every subset $S \subseteq G$ it is clear that if $V(S) = 1$ then $V(\bar{S}) = 0$. From these two observations we can see that

$$|F| = 2^{\frac{\binom{m}{m/2}}{2}}$$

This is because each function $V$ is defined by all the subsets $S \subseteq G$ of size $k$ such that $V(S) = 1$. There are $\binom{m}{m/2}$ subsets of size $k$ and only for half of them $V(S) = 1$.

We focus on states with two players ($n = 2$) and we will denote a state (input) of two players with the same evaluation functions $V_1 = V_2 = V$ by $(V, V)$.

**Lemma 9** *Let $u \neq v$ be arbitrary evaluation functions in $F$. Then, the sequence of bits transmitted on inputs $(v, v)$, is not identical to the sequence of bits transmitted on inputs $(u, u)$.*

Before we prove the lemma let us see how the main theorem is implied. Since different input valuation pairs lead to different communication sequences, we see that the total possible number of communication sequences produced by the protocol is at least the number of valuation pairs $(v, v)$, which is exactly $|F|$. Since $|F|$ is double-exponential, so is the number of different communication sequences produced by the protocol. These are binary sequences, so at least one of the sequences must of length at least $\log |F| = \frac{\binom{m}{m/2}}{2}$, which is exponential.

**Proof** [of lemma]

Assume, by way of contradiction, that $u, v \in F$ and that the communication sequence on $(v, v)$ is the same as on $(u, u)$. We first show that the same communication sequence would also be produced for $(v, u)$ and for $(u, v)$. In particular, the protocol would produce the same allocation for all these cases. Consider the case of $(v, u)$, i.e. player 1 (Alice) has valuation $v$ and player 2 (Bob) has valuation $u$. Alice does not see $u$, so she behaves and communicates exactly as she would in the $(v, v)$ case. Similarly, Bob behaves as he would in the $(u, u)$ case. Since the communication sequences in the $(v, v)$ and the $(u, u)$ cases are the same, neither Alice nor Bob ever notice a deviation from this common sequence, and thus never deviate themselves. In particular, this common sequence is followed also on the $(u, v)$ case. Thus, the same allocation is produced by the protocol in all four cases: $(v, v)$, $(u, u)$, $(v, u)$, $(u, v)$. We will show that this is impossible, since a single allocation cannot be optimal for all four cases.

First we show that in the $(u, v)$ case there is an envy-free allocation (and the same for $(v, u)$). Since $u \neq v$, we have that for some $T \subseteq G$ such that $|T| = k$, $v(T) \neq u(T)$. Without loss of generality, $u(T) = 1$ and $v(T) = 0$ and therefore $v(\bar{T}) = 1$. Thus, for the $(u, v)$ case the allocation $(T, \bar{T})$ is envy-free.

Now (still in the $(u, v)$ case), since the protocol minimizes the total envy, it will produce an envy-free allocation $(S, \bar{S})$ for some $S \subseteq G$. Therefore $u(S) = 1$ (otherwise $u(\bar{S}) = 1$ and the allocation is not envy-free), and also $v(S) = 0$ (same reason). As we showed, the same allocation would be produced also for the $(v, u)$ case. In this case, Alice gets $S$ (the same allocation as in the previous case) and has the evaluation function $v$. However, $v(S) = 0$ and $v(\bar{S}) = 1$, so the allocation is not envy-free, and we have contradiction. Therefore, the sequence of bits transmitted on inputs $(v, v)$, is not identical to the sequence of bits transmitted on inputs $(u, u)$.
■

**Definition 10 (Maximum Marginal Utility)** *Let us denote $\alpha$, the Maximum Marginal Utility, as the maximum additional utility a player can gain by adding one more good to his share. Hence, the Maximum Marginal Utility is given by:*

$$\alpha = \max_{i,g,S \subseteq G \setminus \{g\}} [V_i(S \cup \{g\}) - V_i(S)]$$

**Theorem 11 (Lipton, Markakis, Mossel, Saberi, 04)** *There exists an allocation $A$ such that the maximum envy of $A$ is bounded by the maximum marginal utility of the goods, i.e. $e(A) \leq \alpha$. Furthermore, there is an $O(mn^3)$ time algorithm for finding such an allocation.[2]*

We will define the Envy-Graph, $G_A$, for an allocation $A$, as follows: The nodes are the $N$ players, and there exists an edge $i \rightarrow j$ iff $e_{ij} > 0$. For the proof of Theorem 11 we will use the following lemma:

**Lemma 12** *For any partial allocation $A$, we can find another partial allocation $B$ with envy graph $G_B$ such that:*

1. $e(B) \leq e(A)$
2. $G_B$ *is acyclic*

**Proof** [Lemma 12]

If $G_A$ is acyclic, we are done. So let us suppose $G_A$ contains a cycle $C = i_1 \rightarrow i_2 \rightarrow ... \rightarrow i_r \rightarrow i_1$. We define a new allocation $B$ by re-allocating the goods as follows: if $j \notin C$ then $B_j = A_j$, else $\forall_{i_k \in C}\ B_{i_k} = A_{i_{k+1}}$ (while $B_{i_r} = A_{i_1}$). Note that under the allocation $B$ every player is at least satisfied as he was under $A$, this is because every player such that his share was changed under $B$ in respect to $A$ received the share of a player that he envied in $A$. Hence $e(B) \leq e(A)$. We can show now that this cycle removal process must decrease the number of edges in the envy graph:

- For the edges inside $N \setminus C$ nothing has been changed.
- As for the edges from $N \setminus C$ into $C$, their number must be the same since if a player in $N \setminus C$ envied a player $i_k \in C$ then it would now envy the player $i_{k+1}$ in $C$.
- The number of edges from $C$ into $N \setminus C$ can only decrease, since the players in $C$ are more satisfied now.
- The number of edges inside $C$ must decrease since edges between successive nodes were removed.

Because of the decrease in the number of edges the cycles removal process must end, and we will get an acyclic graph $B$ with $e(B) \leq e(A)$ as required. ■

**Proof** [Theorem 11]

In the first step the algorithm will allocate only the good $g_1$ to a random player. Note that in this stage $e(A) \leq \alpha$, because every player has at most one good less than any other player. In the $(k+1)^{th}$ step we assume there is an allocation $A$ of the first $k$ goods with $e(A) \leq \alpha$. We can now use Lemma 12 to create a new allocation $B$ with $e(B) \leq e(A) \leq \alpha$ and acyclic $G_B$. Because $G_B$ is acyclic it contains a player $i^*$ with in-degree 0, which means no one envies him. Now let the algorithm allocate the good $g_{k+1}$ to the player $i^*$. We now have $\forall_{i,j \neq i^*}\ e_{ij} \leq \alpha$ as before, and:

$$e_{ji^*} = \max\{0, V_j(B_{i^*} \cup \{g_{k+1}\}) - V_j(B_j)\} \leq \alpha + V_j(B_{i^*}) - V_j(B_j) \leq \alpha$$

Where the first inequality holds because if $j$ would get $g_{k+1}$ it will increase his share value by at most $\alpha$. The second inequality holds because $V_j(B_{i^*}) - V_j(B_j) \leq 0$, this is due to the fact that $j$ does not envy $i^*$ in $B$. ■

**Running time analysis**: We'll start with the *naive analysis*, in Lemma 12 we keep removing cycles until the envy graph is acyclic, finding a cycle and removing it takes at most $O(n^2)$, the algorithm has $m$ rounds and in each step the number of edge is bounded by $n^2$, thus the running time is $O(mn^4)$. Now lets use a simple amortized analysis for the running time, allocating of good at each round add at most $n$ edges to the new envy graph, so we add at most $O(m \cdot n)$ edges in the whole process, this is an upper bound on the number of times we have to remove a cycle and total the running time in $O(mn^3)$

## 3 Back to the Divisible Goods Problem

We will now use Theorem 11 to solve the Divisible Goods (Cake Division) Problem. The idea is to partition the cake into indivisible goods of value at most $\alpha$ and then use the algorithm presented above (in Theorem 11) to divide the goods.

we will limit the division protocol to use two queries:

1. $x_2 = cut_i(x_1, \alpha)$ such that $\alpha = V_i([x1, x2])$ this query return $x_2$ such that the interval $[x_2, x_2]$ is worth exactly $\alpha$ to player $i$.

2. $eval_i([x1, x2])$ returns $V_i([x_1, x_2])$.

**Theorem 13** *There exists a protocol that given $\epsilon > 0$ finds a partition $A = (A_1, .., A_n)$ of the cake such that $e(A) \leq \alpha$ ($\epsilon$-envy-free partition) which uses $O(n)$ cut queries and $O(n^2)$ eval queries.*

**Proof** [of theorem] We will define the following protocol:

1. We will ask each player to divide the cake to equally sized parts $x_{i_1}, x_{i_2}, ..., x_{i_j}$ such that $\forall_j\ V_i([x_{i_j}, x_{i_{j+1}}]) \leq \epsilon$ (each part value is at most $\epsilon$). We need $\sim \frac{1}{\epsilon}$ *cut* queries for each player and total of $\sim \frac{n}{\epsilon}$ *cut* queries.

2. We will refer to the intervals between adjacent cut points as an indivisible goods, the marginal value of each good is at most $\alpha$.

3. Now we'll use the previous algorithm: from Theorem 11 we can get a partition $A$ such that $e(A) \leq \alpha \leq \epsilon$. We can gather the required information of each $V_i$ using $\frac{n}{\epsilon}$ *eval* queries and total of $\frac{n^2}{\epsilon}$ *eval* queries. ■

## References

[1] Noam Nisan, Ilya Segal, *The Communication Requirements of Efficient Allocations and Supporting Lindahl Prices*. Technical report - A self-contained CS-friendly executive summary, The Hebrew University in Jerusalem and Stanford University, 2003.

[2] Richard Lipton, Evangelos Markakis, Elchanan Mossel, Amin Saberi, *On Approximately Fair Allocations of Indivisible Goods*. In Proceedings of the 5th ACM Conference on Electronic Commerce (New York, NY, USA, May 17 - 20, 2004). EC '04. ACM, New York, NY, 125-131. DOI=http://doi.acm.org/10.1145/988772.988792