CS880: Algorithmic Mechanism Design

Lecture 1: Introduction **Scribe:** Rojin Rezvan

1.1 Mechanism Design

Mechanism design is **optimization** together with consideration of **strategic behavior**. (Algorithmic Design).

- **Optimization:** Given some reported values and an objective function f, we try to optimize over all solutions that would be feasible according to the environment of the problem:

$$\begin{array}{ccc} \text{Agent 1} & \xrightarrow{x_1} & \\ \text{Agent 2} & \xrightarrow{x_2} & \text{f[Maximizing objective over feasible } y] \quad \to f(x_1, x_2, x_3) \\ \text{Agent3} & \xrightarrow{x_3} & \end{array}$$

- **Strategic Behavior:** Agents give us input values. Each agent gets a value from each outcome:

$$\text{val}_1(\text{outcome})$$
$$\text{val}_2(\text{outcome})$$
$$\text{val}_3(\text{outcome})$$

  According to what the value functions are, agents might signal some other value $\tilde{x_i}$ to better benefit. So the mechanism would compute $f(\tilde{x_1}, \tilde{x_2}, \tilde{x_3})$ instead of $f(x_1, x_2, x_3)$.

From this expression of problem, two questions come to mind:

1. How to incentivize agents to report input truthfully? Using money/payments.
2. How to predict agents behavior? Does strategic behavior hurt overall performance?

In presence of payments, agents would be maximizing their utilities:

$$\text{util(outcome, payment)} = \text{val(outcome)} - \text{amount they pay}$$

We call such utilities **Quasi-linear**. It is called so since the utility function is linear according to the payment. We begin with quasi-linear utility functions but will later see non-linear models for utility as well.

1.2 Literature

- Some other areas in Algorithmic Game Theory are as follows:
  1. **Mechanism Design without Money:** Trying to compute an algorithm not too far from "f" that without getting money from agents would incentivize them to be truthful.
  2. **Social Choice Theory**: Theoretic framework for analyzing a combination of individual's preferences to reach a collective decision.
- **Difference between mechanisms and algorithmic design:** In algorithm design, in order to find algorithm f, we just look at input x, but in mechanism design we should also consider relationship of $f(x)$ to $f(\tilde{x})$. (We should consider entire space of all inputs and size of input space.)
- Mechanism design is an old field within Economics. However, game theory started in early 1900s with Von Neuman and Nash. Mechanism design started as a sub-field of game theory mid-century.
- Start of this field: Algorithmic Mechanism Design by Nisan & Ronen 1999 [1]. Much of the work we will see is about past 20 years.
- **Why Algorithmic Mechanism Design?**
  - Internal: algorithmic questions.
  - External:
    1. Modern markets are increasingly large scale and algorithmic in nature.
    2. Distributed use of computational resources requires economic approaches.
- There are also traditional markets which are combinatorial in nature, such as FCC spectrum auctions.
- A lot of new systems have come out of how new computational technologies have evolved, such as cloud computing with combinatorial nature and distributed features which need economic thinking about how to allocate resources.
- **Objectives:**
  1. Social Welfare or Economic Efficiency: Resources go to those who value/benefit from them the most.
  2. Revenue: Maximizing received payment for seller, very useful in practice.
  3. Other objectives: Fairness, make-span.

1.3 Social Welfare Maximization

Let's start with an example.

Example: Assume we have two items 1 and 2, and two agents A and B. Assume that each agents reports a value for each object and The agents' values are $v_1(1) = 2, v_1(2) = 1, v_2(1) = 3, v_2(2) = 1$. From this we can build a graph as below:

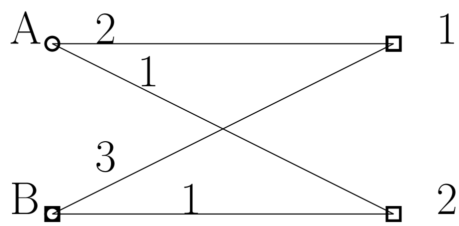

Figure 1.3.1: Maximum matching yields social welfare maximization.

Note that in such graph, finding the maximum bipartite graph and allocating items based on that would yield to welfare maximization. In this case A will get item 2 and B will get item 1, and social welfare would be 4. But will the agents report truthfully? In this allocation A is unhappy because she prefers item 1, and so she would report a very high value for item 1 to make sure she gets it. ■

So what should we do to ensure truthfulness? Consider an example in a simpler case: **Example:**

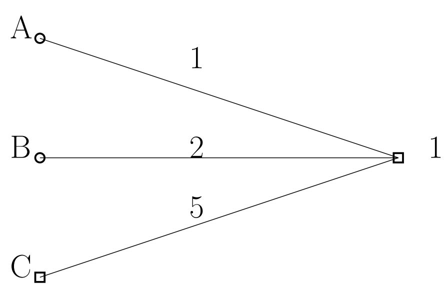

Figure 1.3.2: Single item without monetary charges is not truthful.

In this case everyone would like to report as high a number as possible to get the item.

■

So how can we ensure truthfulness? The answer is to charge the agents some money. Now how much would we charge so that the agents would report truthfully?

Fact 1.3.1 *This is a classic "House Auction" problem. We treat these values as "maximum willingness" to pay.*

Auctions take advantage of such pricing mechanisms. In the next section, we will see different auctions.

1.4 Auctions

1.4.1 First Price sealed-bid auction

Each agent bids a number in a sealed envelope to an auctioneer, and the auctioneer picks the highest option and sells the item to them with the bid value's price.

Question: Is this auction truthful? In the previous example, what each bidder would do depends on what they think about value of others. If they think others have much lower values than them, they will bid lower than their true value, because they will win and have to pay less. So sealed-bid auction is not truthful because bid amounts depends on agents' beliefs about others.

1.4.2 English Auction(Ascending Auction)

Starts from a reserved price and keeps increasing by the bidders until only one person is left in the auction. Some attributes of this auction are:

- The person winning the auction is the one with maximum value. (A good attribute we might seek for, for example in social welfare maximization.)
- Sealed-bid auction is not necessarily like this.
- English auction is efficient. We don't call it truthful but each agent's strategy is clear.

1.4.3 Dutch Auction(Descending Auction; Clock Auction

It works as opposite of English auction. Auctioneer starts at a high reserved price and keeps decreasing it until an agent announces that they would buy it.

Question: What will happen?

Dutch auction, in sense of incentives and strategic behavior, is identical to first price sealed bid auction. Highest bidder might think he has higher value, so they wouldn't stop at their price but possibly lower. This means that in equilibrium, each agent in first price and Dutch auction would act the same way.

Fact 1.4.1 *Note that it does not matter whether the price changes continuously or not. Even if the values are continuous, we can set a decrementing value that all values are divisible by that, and decrease the price using that value, so the whole process would be discrete.*

1.4.4 Second Price Sealed-bid Auction(Veckrey Auction)

Each agent bids, highest bid wins and winner pays second highest bid.

Definition 1.4.2 *We call a mechanism Dominant Strategy Incentive Compatible or DSICwhen it is truthful. The Incentive Compatible part means agents report true values and Dominant Strategy means no matter what others do or believe, each agent would still follow the same strategy.*

Definition 1.4.3 *We call a set of strategies Dominant Strategy Equilibrium or DSE when strategy of each agent might not be truthful, but they have a different dominant strategy.*

Claim 1.4.4 *Second Price Auction is DSIC.*

Proof: Fix agent i with value v_i. Denote by t the maximum bid value amongst other agents except agent i:

$$t = \max_{j \neq i} b_j, \quad b_j \text{is the value bid by agent } j$$

We call this value t the critical value of agent i. Note that this critical value is independent of agent's bid. Now the true value of agent i, or v_i, is either less than t or more. We plot these two cases as follows:

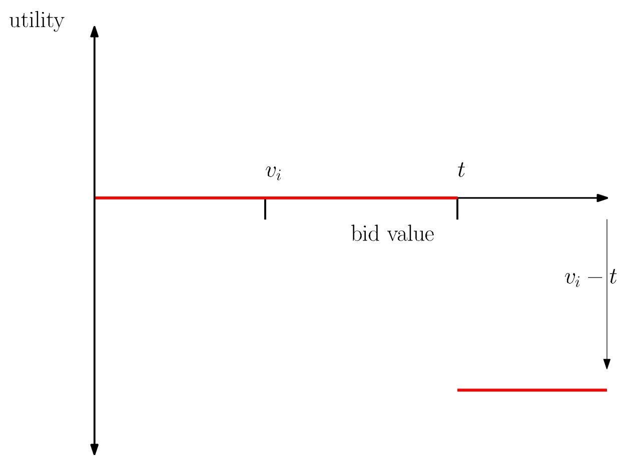

Figure 1.4.3: In this plot, the x axis represents the true value and bid value, and y-axis represents agent's utility. This case represents the situation that the critical price is less than agent's value.

Note that in this case, if the agent bids anything less than t, she will not win and hence not pay anything. If she bids an amount more than t, she will win and pay $v_i - t$. So in this case, biding v_i would give the agent the maximum utility.

Now assume $v_i < t$. (Figure 1.4.4) In this case, if the agent bids less than t, she will not win and hence not pay, and gets utility of 0. If she bids more than t, she will win and pay t and gets utility of $v_i - t$, but since this value is negative, in this case the best strategy for the agent would be to bid less than t.

Note that in both these cases, bidding v_i yields the maximum utility. So this auction is DSIC.

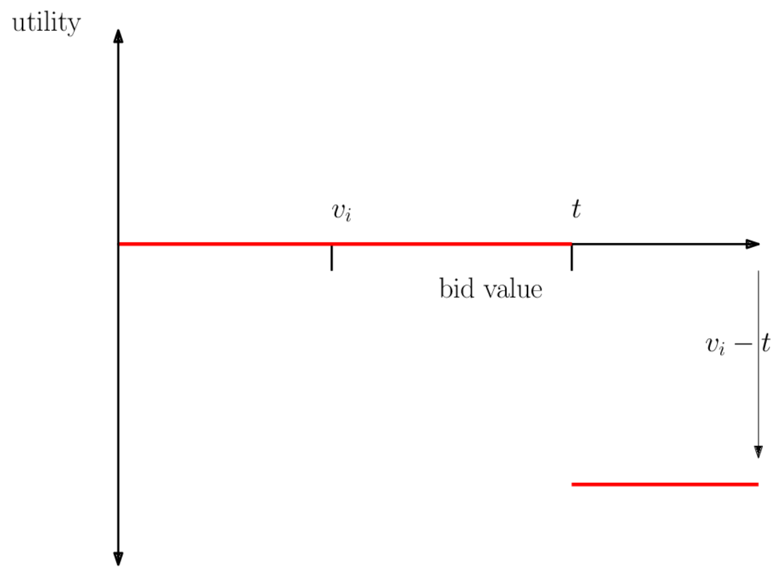

Figure 1.4.4: In this plot, the x axis represents the true value and bid value, and y-axis represents agent's utility.This case represents the situation where the critical price is more than the agent's value.

■

1.4.5 Comparison of Auctions

As for communication between bidders and auctioneer, there can be two types of auctions:

1. **Direct:** The auctioneer asks agents to report their values.
2. **Indirect:** Protocols with many steps and communications, usually with extra complications.

Example: Sealed bid auctions are direct auctions, where as English/Dutch auctions are indirect, because there is a series of communications between the auctioneer and bidders rather than a one time reporting of values. ■

Theorem 1.4.5 ***Revelation Principle:*** *If a social choice function can be implemented by an arbitrary mechanism in a certain equilibrium, then it can be implemented by a direct incentive compatible mechanism in the same equilibrium concept.*

Below we can see the visual sketch of an indirect mechanism vs. a direct one:

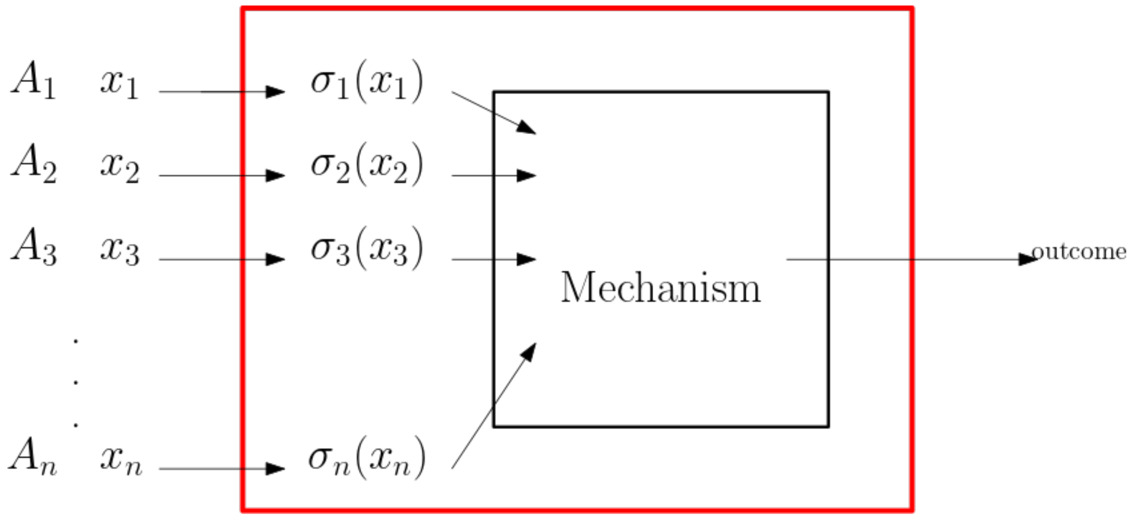

Figure 1.4.5: In this picture we can see an sketch of a direct mechanism vs. an indirect one. The red box represents a direct mechanism, in which the agents only report a value to the auctioneer and do not need to strategize. The direct mechanism gets truthful information and computes agents' strategies for them, so that there is a truthful dominant strategy for everyone. On the other hand, in an indirect mechanism there might be many steps of communications between agents and auctioneer and they need to come up with the strategy function σ on their own.

1.4.6 Combinatorial Auctions

In these auctions we deal with more than one item. Agents have value functions over sets of items. Consider the example below:

Example: Assume we have two items 1 and 2, and two agents A and B. Moreover, assume we are given the below value functions:

$$v_A(\{1\}) = 3, \quad v_A(\{2\}) = 3, \quad v_A(\{1,2\}) = 3$$

$$v_B(\{1\}) = 2, \quad v_B(\{2\}) = 2, \quad v_B(\{1,2\}) = 4$$

We want to maximize the total value we get out of allocation of items. In this example, the optimal allocation is to give item 1 to agent A, and give item 2 to agent B. ■

Question: How much should we charge agents to incentivize them, to report truthful values?

In a more general view of the problem, we will have:

- n agents
- m possible outcomes $O_1, \ldots, O_m$. These outcomes could be feasible allocation of items to agents, or could be a more general space of outcomes.
- utility functions $v_i(g)$ for $i \in [n], j \in [m]$

- Goal: Find $\max_{j \in [m]} \Sigma_{i \in [n]} v_i(j)$ [Social Welfare Objective]

Pictorially, we are given a table as below, and we want to pick an outcome with maximum sum of column:

	O_1	O_2	...	O_j	...	O_m
1						
2						
. .						
i				$v_i(j)$		
. . n						

Figure 1.4.6: In this table, the value of each buyer for each scenario is given. The aim is to find a column with maximum sum. Such outcome will be the one that gives maximum social welfare.

Question: How do we achieve maximum social welfare? Can this be achieved with a direct mechanism where agents report their values (rows) truthfully? The answer to this question is in extension of Vickrey auction which we will see in next section.

1.5 Vickrey-Clark-Groves(VCG) Mechanism

1.5.1 Mechanism Description

The mechanism is as follows:

- Ask agents to report value functions. Assumes these functions are truthful.
- Compute $j = \arg\max \sum_i v_i(j)$
- Charge each agent their **critical value.**

Claim 1.5.1 *VCG is a DSIC mechanism.*

Proof: Here we will see an example to justify what the critical prices should be so that the mechanism is truthful. Fix an agent i. From the perspective of this agent, if we define $v_{-i}(j) = \sum_{i' \neq i} v_{i'}(j)$, table in 1.4.6 will look like this:

	O_1	O_2	...	O_m
i	$v_i(1)$	$v_i(2)$	...	$v_i(m)$
$-i$	$v_{-i}(1)$	$v_{-i}(2)$	...	$v_{-i}(m)$

Figure 1.5.7: Input of mechanism from perspective of agent i. Note that the values of $v_{-i}(j)$ are determined independent of what agent i reports.

We will define one critical price for each outcome. Suppose agent i wanted outcome 1. How much would they have to pay? Let's take a look at the example below:

	O_1	O_2
i	4	1
$-i$	4	6

Figure 1.5.8: A small example representing the reasoning behind critical prices.

If agent i wants outcome 1 to happen, they have to bid at least 2 on outcome 1, but they don't have to bid anything for outcome 2, because it's already the optimal outcome in sense of social welfare. With this in mind, we will define a critical price for each agent and each outcome:

$$p_j = \max_{j'} v_{-i}(j') - v_{-i}(j)$$

Now note that agent i aims to maximize her utility, or $v_i(j) - p(j)$. To maximize this quantity we have:

$$\begin{aligned} \arg\max_j \{v_i(j) - p_j(i)\} &= \arg\max_j \{v_i(j) - \{\max_{j'} v_{-i}(j') - v_{-i}(j)\}\} \\ &= \arg\max_j \{SW(j) - \max_{j'} v_{-i}(j')\} \\ &= \arg\max_j SW(j) \end{aligned}$$

Where $SW(j) = \sum_i v_i(j)$. Note that in order for i to maximize her utility, it is best for her to maximize the social welfare, and that the price she pays is independent of what she report. This is one way of reasoning that this mechanism is DSIC.

Let's see another approach for this proof. Consider agent i and that she reports a value other than her true value, say $b_i(j)$. We want to show that in this case, this agent cannot achieve a utility better than what she would earn when she reports her true value. Note that the critical prices are $p_{j^*}(i)$, when $j^* = \arg\max_j SW(j)$. Now that agent i is reporting some other value, let $j' = \arg\max_t \sum_{k \neq i} v_k(t) + b_i(t)$. In this case the utility agent i gains would be:

$$v_i(j') - p_{j'}(i) = v_i(j') - \max_t v_{-i}(t) + v_{-i}(j')$$

Had she reported her value truthfully, agent i's utility would be:

$$v_i(j^*) - p_{j^*}(i) = v_i(j^*) - \max_t v_{-i}(t) + v_{-i}(j^*)$$

In order for her utility in the non-truthful case to be more, we should have:

$$v_i(j') - \max_t v_{-i}(t) + v_{-i}(j') > v_i(j^*) - p_{j^*}(i) = v_i(j^*) - \max_t v_{-i}(t) + v_{-i}(j*)$$

Or equivalently:

$$SW(j') > SW(j^*)$$

However, this is in contradiction with the way we chose j^*. Therefore, it is not possible that agent i's utility be more when she does not report truthfully. So the mechanism is DSIC. ■

1.5.2 View point of an agent in VCG

- Ask other agents to report values.
- Compute $p_j(i) = \max_{j'} v_{-i}(j') - v_i(j)$
- Allow agent i to select outcome j^*.
- Agent i will choose $j^* = \arg\max_{j} v_i(j) - p_j(i)$

1.5.3 Properties of VCG

- Dominant Strategy Incentive Compatible(DSIC)
- Social Welfare Maximizing

1.5.4 Problems with VCG

- **Needs exact computation of optimum- often computationally hard:** Number of outcomes can be exponentially large. Also we cannot consider a less computationally hard approximation of optimum, because then agents might have incentive to lie to alter their outcome, ie. DSIC property would not necessarily hold.
- **Needs too much communication:** The mechanism asks all agents their value for each outcome. If number of outcomes is exponentially large, so will be the number of communications and that might be unreasonable.

  Example: Spectrum Auctions: A real life auction which maximizes social welfare. FCC auctions on band frequencies of radio waves. Large or small amount of bidders might engage in this auction. Number of packages they auction off is very large, for example across the whole country, and each area could have some constraints. If every bidder was to bid on any possible outcome, it would be impossible to compute. ■

- Revenue non-monotonicity: How the outcome might change when poeple go in and out of auction. This is important from an economic point of view.

  Example: Consider two items, and red and green agents with the valuations in figure 1.5.9. Optimal mechanism cannot allocate to both agents simultaneously, but could allocate to either. Let's say it allocated item 1 to agent red.

  Now let's say we add agent blue as in figure 1.5.10. In this case, the optimal allocation would be to allocate to red and blue. Now note that in previous case, the total payment would be 1 from agent red, and now since the critical price from both agents is 0, the total payment would be 0. ■

  So we added an agent to the system, the social welfare increased, but the agents' payment decreased. So if FCC is concerned about revenue from this spectrum auction, it's not welcome that revenue could go down from presence of more people.

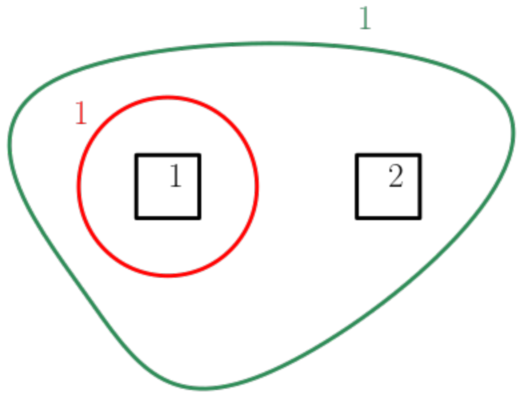

Figure 1.5.9: In this example there are two agents red and green. Agent red values item 1 by 1, and doesn't value item 2. Agent green values the bundle of both items by 1.

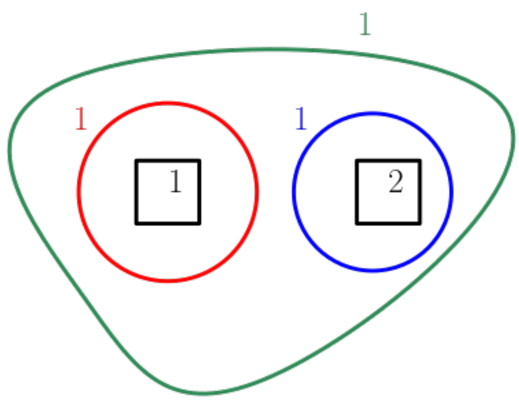

Figure 1.5.10: Previous example when we add an agent blue. This agent values item 2 by 1.

- **Collusion, false name bids:** When agents have incentive to bid under fake agents to decrease their payment, the case we saw in the previous example. This is an unacceptable aspect for companies concerned about revenue. There are other mechanisms that avoid such properties which we will see later.
- **Credibility on part of the auctioneer:** The auctioneer has incentive to enter fake bidders to improve revenue, so VCG is not used in real life.

Question: Can better hardness and DSIC be done simultaneously? Not trivially!

1.6 Special Simpler Case: Single-minded Agents

The computational issue with VCG was to find the optimal case in order to maximize social welfare. In the following special case, the problem would still be computationally hard, but we can come up with some approximation. The model is as follows:

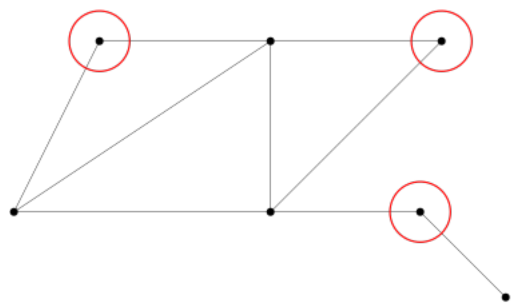

Figure 1.6.11: Representation of independent set problem. In the given graph, the vertices in the maximum independent set are shown with red circles.

- n agents; m items.
- Agent i's value is given by (S_i, v_i), where S_i is the only set of items that agent i is interested in, and v_i is the value she has for that bundle. So agents are only interested in a particular set of items:

  For $T \subseteq [m]$:

$$v_i(T) = \begin{cases} 0 & \text{if } T \not\supseteq S_i \\ v_i & \text{if } S_i \subseteq T \end{cases}$$

Fact 1.6.1 *In the described problem We want a subset of agents with disjoint desired sets, and we want sum of such values to be maximized. (A packing problem)*

Fact 1.6.2 *In this case, we don't have a communication issue because information from each agent is small. But this problem is still NP-hard.*

Theorem 1.6.3 *Social welfare maximization is NP-hard for single-minded agents.*

Proof: Independent set problem can be reduced to this problem. The independent set problem is as follows: Given a graph, find the maximum size independent set:

Fact 1.6.4 *Maximum independent set problem is NP-hard. It is also NP-hard to approximate it better than $n^{1-\epsilon}$ for some $\epsilon > 0$ where n is the number of vertices.*

Now consider the reduction as below:

- Agents are vertices.
- Items are edges.
- Each vertex will want all edges incident on it.
- Each vertex has value 1.

In this reduction, choosing an agent and giving her the bundle of items she desires is equivalent to selecting that vertex to be in the maximum independent set, because once an agent is chosen, this means all the items which are edges incident on it would be allocated to her, so no other agent who desires them, or equivalently is a neighbor of that vertex, can be chosen again. So a special case of social welfare maximization for single minded agents is hard to approximate, and so is the general case. ∎

Corollary 1.6.5 *Note that hard cases of independent set problem occur when the graph is dense. So in this case, $m = O(n^2)$. So **social welfare maximization is hard to approximate with a factor of better than** $m^{\frac{1}{2}-\epsilon}$.*

We can see that even without incentive concerns, the problem is hard to approximate. There are algorithms for approximation of $m = O(n^2)$ factor. We want to extract mechanisms from them. First idea we will pursue is as follows: Instead of computing j^* exactly in VCG, what if we compute an approximation of it and run the same mechanism? Would it ensure DSIC? This exact idea does not work, but some alteration to it does.

1.6.1 Greedy Mechanism

Consider the below greedy mechanism:

1. Ask agents to report (S_i, v_i).
2. Order agents in decreasing order of value:

$$v_1 \geq v_2 \geq \ldots \geq v_m$$

3. Set $S = \varnothing$(Set of items allocated so far)
4. For $i = 1$ to n:

   If $S_i \cap S = \varnothing$

   then allocate S_i to i amd set $S = S \cup S_i$.

5. Charge i their critical price: Θ_i = minimum value at which $i$wins S_i. (Basically we would take out i from the list, find the latest place where S_i has not been covered yet by S, and set value there.)

Claim 1.6.6 *Greedy algorithm is DSIC.*

Proof: Assuming true S_i's are reported, by the same argument of VCG we will have that $v_i's$ should be truthful:

Assume agent i reports $\tilde{v_i}$. Note that the critical price for agent i is determined regardless of what she reports. Now two cases can happen:

1. Agent i wins if $\tilde{v_i} \geq \Theta_i$. Her utility in this case will be $v_i - \Theta_i$. If her actual value is less than Θ_i, her utility would be negative. If her actual value is more, her utility will not change from said utility and she will nevertheless win the auction.

2. Agent i loses if $\tilde{v}_i < \Theta_i$. Her utility in this case will be 0. If her actual value is more than Θ_i, she could have won the auction by reporting truthfully and gained positive utility. If her true value is less than Θ_i she would nevertheless lose and gain the same 0 utility.

So we can see that reporting a non-truthful value would not increase the utility of an agent.

Now note that it doesn't help agent i to report any set other than S_i or a super-set of it, because then they will decrease their chance of winning the auction. This could happen in the case that when it is agent i's turn to be checked, her true set is not covered by S, but her false reported set is, because of some items she does not desire. So chance of getting blocked is minimized when they report S_i and nothing more than that. It is also clear that they would at least report S_i, because other than that the value they have for such set is 0.

This concludes that greedy mechanism is DSIC. ■

Question: How far from optimal is this algorithm? Consider the bad example below:

Example: Assume we have n agents and $n-1$ items. For i from 1 to n, agent i desires item i and values it by 1. Agent n values the whole bundle of items for $1+\epsilon$. In this greedy algorithm, we order agents by decreasing order of values, so agent n will be considered first and he gets all the items. So in this case social welfare would be $1+\epsilon$. But if for i from 1 to $n-1$, agent i was allocated item i, the social welfare would be $n-1$. So the approximation factor of this algorithm is at least $O(n)$. ■

The problem with this algorithm is that it only looks at the values and completely ignores how large of a set they want. So how can we fix the algorithm underlying this mechanism?

1.7 Alternate Greedy Mechanism

1. Ask agents to report (S_i, v_i).

2. Order agents in decreasing order of ratio of value to desired set size::

$$\frac{v_1}{|S_1|} \geq \frac{v_2}{|S_2|} \geq \ldots \geq \frac{v_n}{|S_n|}$$

   [So we are ordering based on value per item]

3. Set $S = \varnothing$(Set of items allocated so far)

4. For $i = 1$ to n:

   If $S_i \cap S = \varnothing$

   then allocate S_i to i amd set $S = S \cup S_i$.

5. Charge i their critical price: $\Theta_i =$ minimum value at which $i$wins S_i.

The same kind of algorithm works in knapsack problem. [Ordering the items based on value per volume.]

Claim 1.7.1 *This new algorithm is DSIC as well.*

Proof: The proof is quite similar to the proof for previous greedy algorithm being DSIC. ■
Question: How far from optimal is this algorithm? Consider the example below: **Example:** Consider 2 agents and m items. Agent 1 desires item 1 by value 1, and agent 2 desires the whole bundle of items by $m-1$. The algorithm will pick agent 1 first, because $\frac{m-1}{m} < 1$. The social welfare of this allocation would then be 1, but in optimal allocation it would be $m-1$ and all items will be given to agent 2. So the approximation factor of this algorithm cannot be better than m. ■

Note that there can be many different functions as to how to order the agents. This allows us to put aside the DSIC issue and focus on what function to pick to get a good approximation factor.

One of the first results in this area of mechanism design that showed that you could bring in interesting ideas from approximation algorithms into mechanism design is by *Lehman, O'callaghan* and *Shoham* at 2002 [2].

1.8 The LOS Algorithm

1. Ask agents to report their values: $(\tilde{S}_i, \tilde{v}_i)$.
2. Order agents in decreasing order of $\frac{\tilde{v}_i}{\sqrt{|\tilde{S}_i|}}$
3. Greedily choose agents to serve in this order.
4. Charge critical payments.

Claim 1.8.1 *LOS is DSIC.*

Proof: The proof is similar to the proof for previous greedy algorithms. ■

Claim 1.8.2 *LOS achieves an $O(\sqrt{m}$ approximation.*

Proof: Now that we know mechanism is DSIC, we will assume that the reported values are truthful.

General approach: We will look at the agents that the algorithm allocated to, and compare what they got with the ones that optimal allocation allocates to.

Assume $O_1, O_2, \ldots$ are the sets of items allocated by optimal mechanism, and $G_1, G_2, \ldots$ is the sets allocated by LOS greedy algorithm. When we look at agents in optimal allocation and their sets, it must be the case that an agent allocated in greedy algorithm has blocked them, and such agent could be the same agent, meaning it is possible that some agents in optimal and greedy allocation are common.

Observation: Every optimal set is blocked by some greedy set.

Now we will look at one greedy set and consider all optimal sets it blocks, and then we will account for all values it blocks:

Consider some agent i that is allocated by greedy algorithm. Let T_i = agents allocated in optimal allocated who are blocked by i, ie. have at least one common desired item.

Let $j \in T_i$. So it must be the case that i appeared earlier in our ordering than j. So we should have:

$$\frac{v_i}{\sqrt{|S_i|}} \geq \frac{v_j}{\sqrt{|S_j|}} \rightarrow v_j \leq v_i \sqrt{\frac{|S_j|}{|S_i|}}$$

Now in order to get the total optimal value blocked by i, we need to sum these values j for all j in T_i:

$$\text{Total optimal value blocked by } i = \sum_{j \in T_i} v_j \leq \frac{v_i}{\sqrt{|S_i|}} \sum_{j \in T_i} \sqrt{|S_j|}$$

This sum can be bounded by Cauchy-Schwarz inequality:

$$\text{Total optimal value blocked by } i \leq \frac{v_i}{\sqrt{|S_i|}} \sqrt{|T_i|} \sqrt{\sum_{j \in T_i} |S_j|}$$

Now let's try to determine how large these quantities can be.

First, note that O_i sets are disjoint. So the number of sets from O_i's that S_i can block is at most the size of S_i. ($|T_i| \leq |S_i|$).

Second, note that j's are agents in optimal allocation, so their desired sets are disjoint, so $\sum_{j \in T_i} |S_j|$ can be at most the total number of items. So $\sum_{j \in T_i} \leq m$.

So eventually we can conclude:

$$\text{Total optimal value blocked by } i \leq v_i \frac{\sqrt{|S_i|}}{\sqrt{|S_i|}} \sqrt{m} \leq \sqrt{m} v_i$$

Now if we sum this maximum blocked value for all agents in greedy, we will have:

$$\text{Total blocked optimal value by all agents in greedy allocation } \leq \sqrt{m} \sum_{i \in Greedy} v_i$$

Now note that the above value is an upper bound for OPT, because it can be the case where two greedy agents cover the same OPT agent, so we account for their blocked value more than once)and at least once). So we get:

$$SW(OPT) \leq \sqrt{m} SW(Greedy)$$

■

Fact 1.8.3 *The LOS mechanism gives us a more general class of mechanisms where we order agents according to some function* $f_i(\tilde{v}_i, \tilde{S}_i)$.

1.8.1 A class of Mechanisms based on LOS

1. Ask agents to report their values $(\tilde{S}_i, \tilde{v}_i)$
2. Order agents in decreasing order of $f_i(v_i, S_i)$.

3. Greedily choose agents to serve in their order.

4. Charge critical payments.

Claim 1.8.4 *If for all i, $f_i(v_i, S_i)$ is non-decreasing in v_i and non-increasing in S_i, then the above mechanism is DSIC.*

What about other kinds of value functions?

1.9 A greedy algorithm by Borodin and Lucier, 2011[3]

This mechanism is an extension of LOS mechanism. Essentially shows that you can use greedy approach for most social welfare maximization settings where greedy works well. The setting considered in this paper is where for each agent we are given value functions that assign value to different subsets, not necessarily single-minded setting. There can be exponentially large vectors that assign values to different subsets. The mechanism is as follows:

1. Ask agents to report $\tilde{v_i}(.)$

2. While items remain un-allocated:
   - Let $(i, X_i) = \arg \max_{(i,X):X \text{ is un-allocated}} f_i(\tilde{v}_i, X)$
   - Allocate S_i to i.

3. Charge each agent their critical price:

$$\Theta_i = \min \text{ reported value } \tilde{v}_i(X) \text{ at which } i \text{ would win } S_i$$

What the mechanism does is that while we have un-allocated items, we select an agent who has not been allocated yet and pick a set of un-allocated items, and we will pick this tuple such that it maximizes $f_i(v_i, X_i)$ over all possible tuples, and then allocate X_i to i.

Claim 1.9.1 *This mechanism is not DSIC.*

So from now on, our assumption would be that underlying greedy algorithm is $c-$ approximation (For a particular choice of f_i's). Also f_i is non-decreasing in $\tilde{v}_i$ and non-increasing in S.

Now that we know agents might lie, we need to come up with some notion to analyze how they might lie. We will use notion of **Nash Equilibrium**:

Definition 1.9.2 ***Nash Equilibrium:*** *A tuple of strategies $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ forms a Nash Equilibrium if for all agents $i \in [n]$, fixing σ_i, agent i's best response is to play σ_i.*

Nash Equilibrium is a fixed point, because every agent is best responding to best response of others.

Claim 1.9.3 *Suppose that the functions f_i are monotone non-decreasing in v_i and the above greedy algorithm is a $c-$approximation. Then, in any Nash Equilibrium of the greedy mechanism, the social welfare obtained is a $(c + 1)-$ approximation to the optimal social welfare.*

For example,if we have an algorithm that gives a $\sqrt{m}$ approximation, we will get a $\sqrt{m}+1-$approximation here.

Definition 1.9.4 ***Price of Anarchy [With a maximization objective]:*** *A way to measure how well a non-truthful mechanism, or a game in general, performs at an equilibrium.*

$$\textit{Price of Anarchy} = \max_{\textit{Nash Equilibrium}} \frac{\textit{Optimal objective}}{\textit{Objective at Nash Equilibrium}}$$

Note that the above value is always greater than 1 in a maximization problem.

Price of anarchy is like an approximation ratio, and is a worse case over all equilibria. It measures how badly a system can perform because of incentives.

Fact 1.9.5 *Note that the previous claim is equivalent to saying that the price of anarchy is at most $c+1$.*

Proof: Here we will see the general steps we will take in proving the claim.

Observation 1. Intuitively, it can't be the case where all agents bid more than their value, because they might end up paying more than they will earn:

$$\sum_{(i,X_i)\text{allocated by greedy}} v_i(X_i) \geq \sum_{(i,X_i)\text{ allocated by greedy}} \tilde{v_i}(X_i)$$

Observation 2.

$$\sum_{(i,X_i)} \tilde{v_i}(X_i) \geq \frac{1}{c} \sum_{(i,S_i)\in OPT} \Theta_i(S_i)$$

Observation 3. Applying Nash Equilibrium property: this allows us to relate $\sum_{(i,S_i)\in OPT} \Theta_i(S_i)$ to $\sum_{(i,X_i)} \tilde{v_i}(X_i)$. Note that v_i cannot be much bigger than Θ_i, because then agents would have incentive to bid higher to win the auction. ■

References

[1] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1):166 – 196, 2001.

[2] Daniel Lehmann, Liadan Ita Oćallaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. ACM*, 49(5):577–602, September 2002.

[3] B. Lucier and A. Borodin. Price of anarchy for greedy auctions.