4 Combinatorial Auctions

In this section, we present a few examples from the area of mechanism design. The fundamental questions that one attempts to address in mechanism design is the following: Assuming that players act strategically, how should we design the rules of the game such that the players’ strategic behavior leads to a certain desirable outcome of the game? As a motivating example, we first consider one of the simplest auctions, known as Vickrey Auction. We then turn to more general combinatorial auctions.

4.1 Vickrey Auction

Suppose there is an auctioneer who wishes to auction off a single item. An instance of the single-item auction consists of

- a set of players $N = [n]$ that are interested in obtaining the item;
- every player $i \in N$ has a private valuation $v_i$ which specifies how much the item is worth to player $i$; $v_i$ is only known to player $i$.
- every player $i$ has a bid $b_i$ which represents the maximum amount player $i$ declares to be willing to pay for the item.

The auctioneer receives the bids and needs to determine who receives the item and at what price. A mechanism can be thought of as a protocol (or algorithm) that the auctioneer runs in order to make this decision. That is, based on the submitted bids $(b_i)_{i \in N}$, the mechanism determines

1. a player $i^*$ in $N$, called the winner, who receives the item, and
2. a price $p$ that this player has to pay for the item.

We define $x_i = 1$ if player $i \in N$ wins the auction and $x_i = 0$ otherwise. We model a player’s preferences over different outcomes of the game by means of a utility function. Let’s assume that the utility function of player $i$ represents the net gain, defined as $u_i = x_i(v_i - p)$. Note that the utility is zero if the player does not receive the item. Otherwise, it is his private valuation minus the price he has to pay. Such utility functions are also called quasi-linear.

There are several natural properties that we want to achieve:

(P1) Strategyproofness: Every player maximizes his utility by bidding truthfully, i.e., $b_i = v_i$.

(P2) Efficiency: Assuming that every player bids truthfully, the mechanism computes an outcome that maximizes the social welfare, i.e., among all possible outcomes $x$ it
Algorithmus 2 Vickrey Auction

1: Collect the bids \((b_i)_{i \in N}\) of all players.
2: Choose a player \(i^* \in N\) with highest bid (break ties arbitrarily).
3: Charge \(i^*\) the second highest bid \(p := \max_{i \neq i^*} b_i\).

chooses one that maximizes the total valuation \(\sum_{i \in N} x_i v_i\); here, this is equivalent to require that the mechanism chooses the player with maximum valuation as the winner.

(P3) Polynomial-time computability: The outcome should be computable in polynomial time.

As it turns out, there is a remarkable mechanism due to Vickrey that satisfies all these properties; this mechanism is also known as Vickrey auction or second-price auction (see Algorithm 2).

Lemma 4.1. In a Vickrey Auction, bidding truthfully \(b_i = v_i\) is a dominant strategy for every player \(i \in N\). More formally, for every player \(i \in N\) and every bidding profile \(b_{-i}\) of the other players, we have

\[
 u_i(b_{-i}, v_i) \geq u_i(b_{-i}, b_i) \quad \forall b_i.
\]

Proof. Consider player \(i\) and fix a bidding profile \(b_{-i}\) of the other players. Let \(B = \max_{j \neq i} b_j\) be the highest bid if player \(i\) does not participate in the game.

Assume \(v_i \leq B\). Then player \(i\) has zero utility if he bids truthfully: Note that player \(i\) loses if \(v_i < B\) and may win if \(v_i = B\) (depending on the tie breaking rule); however, in both cases his utility is zero. His utility remains zero for every bid \(b_i < B\) or if \(b_i = B\) and \(i\) loses (due to the tie breaking rule). Otherwise, \(b_j = B\) and \(i\) wins or \(b_j > B\). In both cases \(i\) wins and pays \(B\). However, his utility is then \(u_i = v_i - B \leq 0\), which is less than or equal to the utility he obtains if he bids truthfully.

Next assume that \(v_i > B\). If player \(i\) bids truthfully, he wins and receives a positive utility \(u_i = v_i - B > 0\). He is worse off by obtaining a utility of zero if he bids \(b_i < B\) or if he bids \(b_i = B\) and loses (due to the tie breaking rule). Otherwise \(b_i = B\) and \(i\) wins or \(b_i > B\). In both cases, \(i\) wins and receives a utility of \(u_i = v_i - B > 0\), which is the same as if he had bid \(b_i = v_i\). \(\square\)

It is easy to see that the Vickrey Auction satisfies (P2) and (P3) as well. More specifically, it satisfies (P2) since it selects the winner \(i^*\) to be a player whose valuation is maximum, assuming that every bidder bids truthfully. Moreover, its computation time is linear in the number of players \(n\). We can thus summarize:

Theorem 4.1. The Vickrey Auction is strategyproof, efficient and runs in polynomial time.

4.2 Combinatorial Auctions and the VCG Mechanism

We now turn to a more general model of auctions. Suppose there is a set \(M\) of \(m \geq 1\) items to be auctioned off to \(n\) players. A player may now be interested in a bundle \(S \subseteq M\) of items. Every player \(i \in N\) has a private valuation function \(v_i : 2^M \to \mathbb{R}^+\), where \(v_i(S)\) specifies player \(i\)'s value for receiving the items in \(S \subseteq M\). We say \(v_i(S)\) is the valuation of player
Algorithm 3 VCG mechanism

1: Collect the bids \( b_i(S) \) for every player \( i \in N \) and every set \( S \subseteq M \).

2: Choose an allocation \( a^* \in O \) such that

\[
a^* = \arg \max_{a \in O} \sum_{i \in N} b_i(a).
\]

3: Compute the price \( p_i \) of player \( i \) as

\[
p_i := b_i(a^*) - \left( \max_{a \in O} \sum_{j \in N} b_j(a) - \max_{a \in O, j \neq i} \sum_{j \in N, j \neq i} b_j(a) \right).
\]

4: return \( a^* \)

\( i \) for bundle \( S \). We assume that \( v_i(\emptyset) = 0 \). (Although this and the assumption that \( v_i(\cdot) \) is non-negative is not essential here).

If every player has a separate value for each item and the value of a subset \( S \subseteq M \) is equal to the sum of all values of the items in \( S \), then we can simply run a separate Vickrey Auction for every item. However, this assumption ignores the possibility that different bundles may have different values. More precisely, for a player \( i \), items in \( S \subseteq M \) might be

- **substitutes**: the player’s valuation to obtain the entire bundle \( S \) might be less than or equal to the individual valuations of the items in \( S \), i.e., \( v_i(S) \leq \sum_{k \in S} v_i(k) \); for example, if the items in \( S \) are (partially) redundant.

- **complements**: the player’s valuation to obtain the entire bundle \( S \) might be greater than or equal to the individual valuations of the items in \( S \), i.e., \( v_i(S) \geq \sum_{k \in S} v_i(k) \); for example, if the items in \( S \) are (partially) dependent.

Here, we consider the most general setting, where we do not make any assumption on the valuation functions \( v_i \) of the players.

Let \( O \) denote the set of all possible allocations of the items in \( M \) to the players. An allocation \( a \in O \) is a function \( a : M \to N \cup \{ \perp \} \) that maps every item to one of the players in \( N \) or to \( \perp \), which means that the item remains unassigned. Let \( a^{-1}(i) \) be the subset of items that player \( i \in N \) receives. Every player declares a bid \( b_i(S) \) for every bundle \( S \subseteq M \). (Let’s not care about polynomial-time computability for a moment.) For the sake of conciseness, we slightly abuse notation: Given an allocation \( a \in O \), we write \( v_i(a) \) and \( b_i(a) \) to refer to \( v_i(a^{-1}(i)) \) and \( b_i(a^{-1}(i)) \), respectively. The auctioneer needs to decide how to distribute the items among the players in \( N \) and at what price. That is, he determines an allocation \( a \in O \) and a pricing vector \( p = (p_i)_{i \in N} \), where player \( i \) obtains the bundle \( a^{-1}(i) \) at a price of \( p_i \). As before, we consider quasi-linear utility functions: The utility of player \( i \), given the outcome \( (a, p) \), is \( u_i = v_i(a) - p_i \).

A mechanism is strategyproof in this setting if a dominant strategy for every player is to bid \( b_i(S) = v_i(S) \) for every \( S \subseteq M \). Moreover, a mechanism is efficient, if it outputs an allocation \( a^* \) that maximizes the total social welfare, i.e., \( a^* = \arg \max_{a \in O} \sum_{i \in N} v_i(a) \), assuming that every player truthfully reports his valuation.

A powerful mechanism for this quite general class of combinatorial auctions is known
as VCG mechanism due to Vickrey, Clarke and Groves (see Algorithm 3). In particular, as we will see, the VCG mechanism is strategyproof and efficient.

**Theorem 4.2.** The VCG mechanism is strategyproof and efficient.

**Proof.** Clearly, if every player bids truthfully the allocation \( a^* \) output by the VCG mechanism maximizes total social welfare. Thus, the VCG mechanism is efficient.

We next prove that the VCG mechanism is strategyproof. Consider an arbitrary player \( i \in N \). Let \( b = (b_{-i}, b_i) \) be the bid vector of some arbitrary bids and let \( \bar{b} = (b_{-i}, v_i) \) be the same bid vector, except that player \( i \) reports his private valuations truthfully. Moreover, let \((a^*, p)\) and \((\bar{a}^*, \bar{p})\) be the outcome computed by the VCG mechanism for input \( b \) and \( \bar{b} \), respectively. Observe that we have

\[
\bar{b}_i(a) = v_i(a) \quad \forall a \in O \quad \text{and} \quad \bar{b}_j(a) = b_j(a) \quad \forall j \neq i, \forall a \in O. \tag{13}
\]

Moreover, \( \bar{a}^* \) has been chosen such that

\[
\sum_{j \in N} \bar{b}_j(\bar{a}^*) \geq \sum_{j \in N} \bar{b}_j(a) \quad \forall a \in O. \tag{14}
\]

Using these two observations, we can infer:

\[
v_i(\bar{a}^*) - \bar{p}_i = v_i(\bar{a}^*) - \left[ \bar{b}_i(\bar{a}^*) - \left( \max_{a \in O} \sum_{j \in N} \bar{b}_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \right) \right]
\]

\[
= \sum_{j \in N} \bar{b}_j(\bar{a}^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a)
\]

\[
\geq \sum_{j \in N} \bar{b}_j(\bar{a}^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a)
\]

\[
= \sum_{j \in N, j \neq i} b_j(a^*) + v_i(a^*) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a)
\]

\[
= v_i(a^*) - \left[ b_i(a^*) - \left( \max_{a \in O} \sum_{j \in N} b_j(a) - \max_{a \in O} \sum_{j \in N, j \neq i} b_j(a) \right) \right]
\]

Thus, \( b_i = v_i \) is a dominant strategy for player \( i \).

Although the VCG mechanism satisfies strategyproofness and efficiency, it is highly computationally intractable. In particular, the mechanism relies crucially on the fact that one can compute an optimal allocation \( a^* \in O \). This problem is typically also called the \textit{allocation problem}.

34