# Economics and Computation

ECON 425/563 and CPSC 455/555

Professor Dirk Bergemann and Professor Joan Feigenbaum

## Combinatorial Auctions

In case of any questions and/or remarks on these lecture notes, please contact Oliver Bunn at oliver.bunn(at)yale.edu.

# 1 Combinatorial Auctions

## 1.1 Overview

Our treatment of combinatorial auctions will be organized as follows:

1. Basic Structure
2. Vickrey-Clarke-Groves (VCG) mechanism;
3. Detour into Query-Models
4. Efficiency and Linear Programming
5. Efficient Allocations and Walrasian Equilibria
6. Detour into Ascending Bundle-Price Auctions

The analysis will rely strongly on chapter 11 of the textbook, [NRTV08].

## 1.2 Basic Structure

### 1.2.1 Basic Elements and Notation

- The **set of objects/items** is given by $\mathcal{K} = \{1, ..., K\}$. The **set of possible bundles** that can be formed from $K$ is denoted by the power-set[1] $2^{\mathcal{K}}$. An arbitrary bundle is denoted by $S \in 2^{\mathcal{K}}$. Observe that the cardinality of $2^{\mathcal{K}}$, i.e. the number of elements in the set $2^{\mathcal{K}}$, is given by $2^K$.
- The **set of bidders/agents** is given by $\mathcal{N} = \{1, ..., N\}$.
- Agent $n \in \mathcal{N}$ has a **valuation function**

  $$v_n : 2^{\mathcal{K}} \to \mathbb{R}_+.$$

  That is, each agent $n \in \mathcal{N}$ assigns a non-negative number to every possible subset of the set of objects $\mathcal{K}$.

**Assumption 1** *For any bidder $n \in \mathcal{N}$, the valuation-function $v_n$ satisfies "free disposal", i.e.*

- *$v_n$ is normalized, i.e.*

  $$v_n(\emptyset) = 0.$$

[1] The power-set of $\mathcal{K}$ refers to the set of all subsets of $\mathcal{K}$. For example for the set $\{1, 2, 3\}$ the power-set is given by $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

- *$v_n$ is monotone, i.e. for any two sets $S, T \in \mathcal{K}$ satisfying $S \subseteq T$*

$$v_n(S) \leqslant v_n(T).$$

Assumption 1 imposes two very intuitive restrictions on every bidders' valuation function:

- If a bidder does not receive any object, then his valuation is zero.
- A bundle $T$ that is simply a superset of a bundle $S$ cannot yield strictly lower valuation for a bidder than the bundle $S$. That is, additional elements cannot make a bidder worse off.

Most importantly, the valuation-function $v_n$ is agent $n$'s private knowledge.[2] She can be asked to report it, but it is by no means clear that agent $n$ will actually tell the truth. So, it is the task of the mechanism-designer to set up a mechanism such that it is in the agent's self-interest to report her valuation. This will be a crucial feature of mechanisms that we will be talking about below.

In order to reflect different preferences for bundles of items, certain restrictions CAN[3] be imposed on the valuation-function $v_n$ for an agent $n \in \mathcal{N}$:

- $v_n$ can be assumed to be **additive**, i.e.

  $$v_n(S) = \sum_{k \in S} v_n(k).$$

  Under the linearity-assumption, one obtains the valuation for a bundle by simply adding up the valuations for the objects in $\mathcal{K}$ that constitute the subset $S$. In consequence, one does not need to worry about specifying the valuation of agent $n$ for any possible subset $S \subseteq K$, but all valuations can be reverse-engineered from the valuations of the objects $k \in \mathcal{K}$.
  Another important aspect is related to the point of view of an auctioneer. She does not have to worry about offering bundles of the objects to be auctioned off, but it is sufficient for her to assign each object $k \in \mathcal{K}$ in a separate auction.

- Alternatively, the valuation function might be defined in a non-additive manner for two arbitrary subsets $S, T \in 2^{\mathcal{K}}$ satisfying $S \cap T = \emptyset$:

[2] So, the agent's valuation function will be regarded as his type. This has previously been denoted by $t_n$.

[3] In comparison to the property of "free disposal", the following properties will not be assumed a priori. They will be mentioned explicitly every time that each of them is imposed.

– $S$ and $T$ are said to be **complements** iff

$$v_n(S \cup T) \geqslant v_n(S) + v_n(T).$$

That is, agent $n$ values having both bundles more than getting either one of them.

– $S$ and $T$ are said to be **substitutes** iff

$$v_n(S \cup T) \leqslant v_n(S) + v_n(T).$$

That is, agent $n$ values having both bundles less than getting either one of them.

Making use of the valuation-function[4] $v_n$ of any agent $n \in \mathcal{N}$, it will be assumed that **agent $n$'s utility function** has the following functional form:

$$\begin{aligned} u_n : 2^{\mathcal{K}} \times \mathbb{R} &\rightarrow \mathbb{R} \\ (S, t) &\mapsto v_n(S) - t \end{aligned}$$

So, agent $n$'s utility has an arbitrary subset $S \subseteq \mathcal{K}$ and a **monetary transfer** $t$ as an input. Then, her utility is given by her valuation of the bundle $S$ minus the transfer that she has to make. This particular specification of the utility-function is called **quasi-linear utility**.

In order to complete the model, one also needs to think about the assignment of a utility-function to the auctioneer, which we will also refer to as the government or the residual recipient.[5] This utility function, denoted by $u_0$, is given by the following mapping:

$$\begin{aligned} u_0 : 2^{\mathcal{M}} \times \mathbb{R}^N &\rightarrow \mathbb{R} \\ (S; t_1, ..., t_N) &\mapsto \sum_{n=1}^{N} t_n \end{aligned}$$

The government's utility function takes a subset from $\mathcal{M}$ and the transfers from all agents as an input. Its utility value is simply the sum of the transfers. The fact that a subset of $\mathcal{M}$ appears as an input of $u_0$, but does not affect the government's utility at all is purely due to notational convenience. It is by no means clear that the auctioneer will necessarily distribute all objects to the bidders. Hence, the government will be considered as an

[4]None of the properties for $v_n$ need to hold for the following discussion - even "free disposal" is not necessary for the results in this section. The following statements are valid for any functional assumption of $v_n : 2^{\mathcal{K}} \rightarrow \mathbb{R}_+$.

[5]In the following analysis, we will always assume that the auctioneer is benevolent, i.e. he does not have a self-interest, but acts in the interest of the entity of the participants in the auction.

additional entity that will receive all unassigned objects. The subset of the set $\mathcal{K}$ that the government receives will be denoted by $S_0$. Furthermore, we will use the notation

$$\mathcal{N}_0 \triangleq \{0\} \cup \mathcal{N}$$

to describe the set of agents plus the government.

### 1.2.2 Efficiency

In this subsection, we take the point of view of the auctioneer and try to determine the "best" or efficient way (which will be made precise below) to distribute the $M$ objects among the $N$ bidders and the government, i.e. over the set $\mathcal{N}_0$.

In order to come to the concept of efficiency, we need to formally define the concept of a **(feasible) allocation**:[6]

**Definition 1** *A **feasible allocation** is a partition of the set $\mathcal{K}$ over the set $\mathcal{N}_0$, i.e. a collection of subsets*

$$S = (S_0, S_1 ..., S_N)$$

*such that:*

1. *For all $n, n' \in \mathcal{N}_0$*
$$S_n \cap S_{n'} = \emptyset;$$
2. $$\bigcup_{n=0}^{N} S_n = \mathcal{K}.^7$$

*The **set of all such partitions** (feasible allocations) will be denoted by $\mathcal{S}$.*

So, an allocation assigns a subset of objects from $\mathcal{K}$ to each player $n \in \mathcal{N}$. We require the intuitive conditions to be satisfied that no object is assigned to more than one player (property 1.) and that the union of all assignments stays within the bounds of the set of available objects (property 2.)

In order to obtain efficiency, it is necessary to aggregate all market-participants' utilities into a social (economy-wide) utility. An example of this aggregation is to take sum all utilities, to which we will restrict attention in the following:[8]

[6] We will not consider any unfeasible allocations. Therefore, the following definition of "allocation" already incorporates feasibility.

[7] It is exactly the inclusion of the government that yields equality in this condition.

[8] The sum of all participants' utilities is by far the most widespread criterion for efficiency.

Taking an arbitrary feasible allocation $(S_0, S_1, ..., S_N)$ and a tuple of transfer $(t_1, ..., t_N)$, the **social utility** is given by

$$\begin{aligned} & \sum_{n=1}^{N} u_n(S_n, t_n) + u_0(S_0; t_1, ..., t_N) \\ = & \sum_{n=1}^{N} [v_n(S_n) - t_n] + \sum_{n=1}^{N} t_n \\ = & \sum_{n=1}^{N} v_n(S_n) \end{aligned}$$

The last expression (the sum of all agent's valuations) is also referred to **gross utility**.

So, an **efficient allocation** is a feasible allocation that maximizes gross or social utility, i.e. $(S_1^*, ..., S_N^*)$ is called an efficient allocation iff

$$(S_1^*, ..., S_N^*) \quad \in \operatorname{argmax} \left\{ \sum_{n=1}^{N} v_n(S_n) \right\}$$
$$\text{s.t. feasibility-conditions 1. and 2.}$$

## 1.3 VCG-mechanism

Previously, we have already discussed the Second-Price auction as an example of an efficient allocation.[9] In the following, we aim to generalize the concept of the Second-Price auction by introducing the Vickrey-Clarke-Groves[10] (VCG) mechanism. But, before that, we have to clarify the notions of **direct mechanism** and **truthful revelation in dominant strategies**.

### 1.3.1 Valuation-Functions and Construction of a Direct Mechanism

A **direct mechanism** is given by the following pair of mappings:

- An allocation:

$$a : \mathbb{R}_+^{2^K \cdot N} \rightarrow \mathcal{S},$$

  $a = (a_1, ..., a_N)$, where $a_n : \mathbb{R}^{2^K \cdot N} \rightarrow 2^{\mathcal{K}}$ denotes the allocation that is assigned to player $n \in \mathcal{N}$.

[9] In this case, our argument for efficiency rested on the fact that the agent/bidder with the highest valuation received the object, which is completely in line with our efficiency-criterion above.

[10] The corresponding articles are Vickrey (1962), Clarke (1972) and Groves (1974).

- A vector of transfers:

$$t : \mathbb{R}_+^{2^K \cdot N} \to \mathbb{R}^N$$

.

Both mappings that constitute a direct mechanism have $\mathbb{R}_+^{2^K \cdot N}$ as their domain. That, is they take a report about all valuations (remember that an agent's valuations specifies a non-negative value for every subset, i.e. a valuation-vector for an agent has length $\mathbb{R}_+^{2^K}$) from all agents (N agents) as their input. Then, the allocation-mapping outputs a (feasible) allocation as described above and the transfer-mapping specifies a monetary amount that agent $n$, $n \in \mathcal{N}$, has to pay to the government (the benevolent planner).

### 1.3.2 Design of the Transfer

The transfer-vector $t = (t_1, ..., t_N) \in \mathbb{R}^N$ will be specified with a very particular goal in mind:

**Truth-telling shall be a dominant strategy for the agent**, i.e.

for all $n \in \mathcal{N}$ and all $v_n \in \mathbb{R}_+^{2^K}$ the following condition is satisfied

$$\begin{aligned} & v_n(a_n(v'_1, ..., v_n, ..., v'_N)) - t_n(v'_1, ..., v_n, ..., v'_N) \geqslant \\ & v_n(a_n(v'_1, ..., v'_n, ..., v'_N)) - t_n(v'_1, ..., v'_n, ..., v'_N) \qquad \forall \text{ tuples } (v'_1, ..., v'_n, ..., v'_N) \end{aligned}$$

The following remarks on this condition can be made:

- Player $n$ compares two different regimes. In the first regime, she truthfully reports the $\mathbb{R}_+^{2^K}$-vector to the mechanism-designer as $v_n$. In the second regime, she makes up a valuation-vector $v'_n$ to report to the mechanism-designer.
- Agent $n$ makes the comparison between the two scenarios for all possible reports that the other agents may submit, i.e. for $(v'_1, ..., v'_{n-1}, v'_{n+1}, ..., v'_N)$. This is the characteristic feature of the concept of "dominant strategies", because these strategies are optimal irrespective of the opponents' actions. This is in contrast to any notion of Nash-equilibrium that we have considered so far. Here, one presupposes a certain kind of action for the opponents, namely the equilibrium-actions.
- The vectors of valuations for all agents are plugged into the allocation-function and agent $n$'s allocations as represented by the $n$th row $a_n$ of the allocation-matrix.
- Finally, the allocation that agent $n$ is assigned is evaluated according to agent $n$ true valuation function $v_n$.

Abstractly, agent $n$'s strategy, $n \in \mathcal{N}$, in this setting can be defined as a mapping $r_n$ as follows:

$$m_n : \mathbb{R}_+^{2^K} \to \mathbb{R}_+^{2^K}.$$

So, agent $n$ takes her valuation-vector $v_n$ and transforms it into her report (which is often referred to as her message) $m_n(v_n)$. It will only be her report that she will announce toward the mechanism-designer. In contrast, $v_n$ will remain her private knowledge.
The concept of truthful revelation corresponds to $r_n$ being the identity-mapping.

### 1.3.3 VCG-mechanism

As a particular example of a transfer-scheme which induces truth-telling as a dominant strategy, we will specify the VCG-mechanism in the following. This mechanism is also referred to as **social externality pricing**. That is, any agent is supposed to make a payment according to the negative externality that she imposes on the remaining agents by her presence.

Compare the following two social programs:[11]

- Social program including agent j:

$$S^* = (S_1^*, ..., S_{j-1}^*, S_j^*, S_{j+1}^*, ..., S_N^*) \in \operatorname{argmax} \sum_{n=1}^{N} v_n(S_n).$$

- Social program excluding agent j:

$$S_{-j}^* = (S_1^*, ..., S_{j-1}^*, S_{j+1}^*, ..., S_N^*) \in \operatorname{argmax} \sum_{n \neq j} v_n(S_n).$$

In the social program including agent $j$, the objects from the set $\mathcal{K}$ are optimally and feasibly distributed taking all agents into account. In contrast, the second program simply excludes agent $j$ from the determination of an optimal and feasible distribution of the elements in $\mathcal{K}$.

Denote

$$\begin{aligned} V^* &\triangleq \sum_{n=1}^{N} v_n(S_n^*), \\ V_{-j}^* &\triangleq \sum_{n \neq k} v_n(S_{-j,n}^*). \end{aligned}$$

[11] Both programs are obviously computed under the feasibility-restriction.

So, $V^*$ and $V^*_{-j}$ describe the levels of social utility from the two social programs.

Observe the following general properties that arise from the comparison of the two programs:

1. Assume $S^*_j = \emptyset$. That is, in the program that includes agent $j$, this particular agent optimally receives no element from $\mathcal{M}$. In other words, agent $j$ does not impose any externality on the other agents. In this case, the difference between $V^*$ and $V^*_{-j}$ is equal to zero.

2. In general, the following inequality holds:
$$V^* \geqslant V^*_{-j}.$$
The optimization-problem for $V^*_{-j}$ can be seen as a special case of the optimization-problem for $V^*$ in which the additional restriction $S^*_j = \emptyset$ is imposed. So, the inequality above simply originates from the fact that the maximum for $V^*$ is taken over a superset of the set over which $V^*_{-j}$ is determined.

Now, we are in the position to ask ourselves what is the exact amount of the externality that agent $j$ should be charged. It is flawed to simply take the difference between $V^*$ and $V^*_{-j}$, because this ignores agent $j$'s contribution to social welfare in the first program. Part of the payment of agent $j$ would be his own valuation that he contributes to social welfare. This problem can be overcome by the following definition of the transfer in the VCG-mechanism:
$$t_j^{\text{VCG}} \triangleq \sum_{n \neq j} v_n(S^*_{-j,n}) - \sum_{n \neq j} v_n(S^*_n).$$
This exactly reflects the notion of social externality pricing. Agent $j$ is charged the difference in the sum of the utilities of all other agents (social utility without agent $j$), when she is not considered in the allocation (first sum) and when she is (second sum).

**Lemma 1**

*For all $j \in \mathcal{N}$, the inequality $t_j^{VCG} \geqslant 0$ holds.*

**Proof of Lemma 1**

The first summand $\sum_{n \neq j} v_n(S^*_{-j,n})$ in the definition of $t_j^{\text{VCG}}$ describes the optimal level of social utility that can be achieved of one takes agents $(1, ..., j-1, j+1, ..., N)$ into account. The second summand simply denotes another level of social utility in the situation that considers $(1, ..., j-1, j+1, ..., N)$. Therefore, the second summand is necessarily smaller than or equal to the first summand.

□

Making use of the definition of $t_j^{\text{VCG}}$, agent $j$'s utility from the VCG-mechanism is given by

$$v_j(S_j^*) - t_j^{\text{VCG}}.$$

**Lemma 2**

*The following two properties hold for any agent $k \in \mathbb{N}$:*

1. *Agent $j$ will participate in the mechanism.*[12] *Put differently, her utility is non-negative, i.e.*
$$v_j(S_j^*) - t_j^{VCG} \geqslant 0.$$
2. *Truth-telling is a dominant-strategy.*

**Proof of Lemma 2**

Applying the definition of $t_j^{\text{VCG}}$, one obtains

$$\begin{aligned} v_j(S_j^*) - t_j^{\text{VCG}} &= v_j(S_j^*) - \left[\sum_{n \neq j} v_n(S_{-j,n}^*) - \sum_{n \neq j} v_n(S_n^*)\right] \\ &= v_j(S_j^*) + \sum_{n \neq j} v_n(S_n^*) - \sum_{n \neq j} v_n(S_{-j,n}^*) \\ &= \sum_{n=1}^{N} v_n(S_n^*) - \sum_{n \neq j} v_n(S_{-j,n}^*) \qquad (1) \\ &= V^* - V_{-j}^*. \qquad (2) \end{aligned}$$

As it has been argued above in the general property 2., the difference $V^* - V_{-j}^*$ from (2) is non-negative, proving part 1. of the claim.
In order to verify claim 2., observe that the expression $V_{-j}^* = \sum_{n \neq j} v_n(S_{-j,n}^*)$ in (1) does not depend on the report of agent $j$, because he is simply not taken into account. So, agent $j$ tries to choose a report in order to maximize the social utility $V^* = \sum_{n=1}^{N} v_n(S_n^*)$ in (1). This will guarantee herself maximum-possible utility. But fixing the other agents' reported valuations, it is exactly the true valuation for agent $j$ which will yield the best possible level of social utility for agent $j$, so it is optimal for agent $j$ to tell the truth. Because the other agents' reports have been assumed to be arbitrary, it follows that truth-telling is a dominant strategy for agent $j$.

□

[12] Implicit in this statement is the assumption that the outside-option of any agent is zero, i.e. any agent who does not participate in the mechanism obtains zero utility.

### 1.3.4 Special Cases of the VCG-Mechanism

In this part we will look at the VCG-mechanism in the specific context of auctions with unit demand.[13] We will assume that all bidders are sorted by their valuation, i.e. we have

$$v_1 \geqslant v_2 \geqslant ... \geqslant v_N.$$

**Example - Single Object**

So, first consider a situation in which there is only one good to be auctioned off:

- Each agent/bidder simultaneously submits a bid for an object.
- The person with the highest bid wins the object.

So, what will be the transfers that the VCG-mechanism prescribes? Remember that VCG implies that truth-telling is a dominant strategy, so we do not have to care about strategies, but can simply restrict attention to valuations of the agents:

- For agent 1, i.e. $j = 1$:
  - According to the rules of the auction, she will be the bidder who receives the object.
  - Social utility is $V^* = v_1$ in the program that includes her, since she is the only one who receives the object.
  - If agent 1's valuation is subtracted from $v^*$ in order to obtain the sum $\sum_{n \neq j} v_n(S_n^*)$, it follows that
  $$\sum_{n \neq j} v_n(S_n^*) = 0.$$
  - Now, assume that agent 1 is excluded from the social program. Then, it will be agent 2 who receives the only object to be auctioned off. In consequence, social utility equals agent 2's valuation, i.e.
  $$\sum_{n \neq j} v_n(S_{-j,n}^*) = v_2.$$
  - By the definition of $t_1^{\text{VCG}}$, it follows that
  $$t_1^{\text{VCG}} = v_2.$$
- For any agent $j \in \{2, ..., N\}$:

[13] Unit demand means that each bidder/agent only wants to obtain one object.

- According to the rules of the auction, she will not receive the object, so her valuation is zero.
- Social utility is $v^* = v_1$ in the program that includes her.
- Because agent $j$'s valuation is zero, $v^*$ remains unchanged if agent $j$'s valuation is subtracted. Therefore, it follows that

$$\sum_{n \neq j} v_n(S_n^*) = v_1.$$

- Now, assume that agent $j$ is excluded from the social program. Then, it will still be agent 1 who receives the only object to be auctioned off. In consequence, social utility equals agent 1's valuation, i.e.

$$\sum_{n \neq j} v_n(S_{-j,n}^*) = v_1.$$

- By the definition of $t_j^{\text{VCG}}$, it follows that

$$t_j^{\text{VCG}} = 0.$$

Therefore, it is only agent 1 who has to make a payment and this payment equals the second-highest bid/valuation. This is exactly the logic of the Second-Price auction.

**Example - $K$ Objects**

Now, consider a situation in which there are $k$ identical goods to be auctioned off:[14]

- Each agent/bidder simultaneously submits a bid for an object.
- The person with the m highest bids each win one object.

So, what will be the transfers that the VCG-mechanism prescribes? Again, remember that VCG implies that truth-telling is a dominant strategy, so we do not have to care strategies, but can simply restrict attention to valuations of the agents:

- For any agent $j \in \{1, ..., k\}$:
  - According to the rules of the auction, she will one of the bidders who receive one object.

[14] For ease of exposition, we will assume that the number of bidders is strictly bigger than the number of objects to be auctioned off.

- Social utility is $V^* = \sum_{n=1}^{k} v_n$ in the program that includes her.
- If agent $j$'s valuation is subtracted from $v^*$ in order to obtain the sum $\sum_{n\neq j} v_n(S_n^*)$, then it follows that
$$\sum_{n\neq j} v_n(S_n^*) = \sum_{n\in\{1,...,k\}\setminus\{j\}} v_n.$$
- Now, assume that agent $j$ is excluded from the social program. Then, it will be agents $1, ..., j-1, j+1, ..., k, k+1$ who receive one object to be auctioned off. In consequence, social utility equals
$$\sum_{n\neq j} v_n(S_{-j,n}^*) = \sum_{n\in\{1,...,k+1\}\setminus\{j\}} v_n.$$
- By the definition of $t_j^{\mathrm{VCG}}$, it follows that
$$t_j^{\mathrm{VCG}} = v_{k+1}.$$

- For any agent $j \in \{k+1, ..., N\}$:
  - According to the rules of the auction, she will not receive the object, so her valuation is zero.
  - Social utility is $v^* = \sum_{n=1}^{k} v_n$ in the program that includes her.
  - Because agent $j$'s valuation is zero, $v^*$ remains unchanged if agent $j$'s valuation is subtracted. Therefore, it follows that
$$\sum_{n\neq j} v_n(S_n^*) = \sum_{n=1}^{k} v_n.$$
  - Now, assume that agent $j$ is excluded from the social program. Then, it will still be agents $1, ..., k$ who receive one unit of the object to be auctioned off. In consequence, social utility equals the sum of agent 1's to $k$'s valuation, i.e.
$$\sum_{n\neq j} v_n(S_{-j,n}^*) = \sum_{n=1}^{k} v_n.$$
  - By the definition of $t_j^{\mathrm{VCG}}$, it follows that
$$t_j^{\mathrm{VCG}} = 0.$$

Therefore, it is agents $1, ..., k$ who have to make a payment and these payments equal the (k+1)-highest bid/valuation. This is a generalization of the notion of a Second-Price auction, which is called $(k+1)$**th Price auction**.

**Example - 2 bidders and 2 Objects**

Now, we will move away from the general framework of the previous two examples. We will consider the following explicit valuation-profile in a setting of two bidders competing for two objects:

| | $v(a,b)$ | $v(a)$ | $v(b)$ |
|---|---|---|---|
| Alice | 2 | $\alpha$ | $\beta$ |
| Bob | 2 | 2 | 2 |

Hereby, the condition $0 < \alpha < \beta < 1$ is satisfied.

The beginning of the analysis of this example will be the determination of the efficient allocation, i.e. the particular allocation of the two objects $a, b$ such that the sum of the utilities of Alice and Bob is maximized. Obviously, this efficient allocation is given by the following allocation:

- Assign item $b$ to Alice.
- Assign item $b$ to Bob.

Social utility in this case is given by $2 + \beta$.

In a next step of the analysis, the transfers that are implied by the VCG-mechanism will be determined. This is a generalization of the VCG-mechanism insofar as this mechanism can also be used to determine bundle-prices. The previous description hinged on the VCG-prices implementing the efficient allocation.

But, taking a specific bundle to be part of the allocation to be implemented allows to continue to apply the logic of the VCG-mechanism and obtain bundle-specific VCG-transfers for each agent.

The resulting VCG-transer-scheme is given by:

| | $v(a,b)$ | $v(a)$ | $v(b)$ |
|---|---|---|---|
| $t_A^{\text{VCG}}$ | 2 | 0 | 0 |
| $t_B^{\text{VCG}}$ | 2 | $2-\beta$ | $2-\alpha$ |

This transfer-scheme is obtained as follows:

- For $t_A^{\text{VCG}}(\{a,b\})$:

  If Alice is not involved in the allocation, then it is the efficient allocation to assign Bob the bundle $\{a,b\}$, which gives him valuation 2. If Alice is involved and is assigned the bundle $\{a,b\}$, giving her valuation 2, then Bob's valuation is zero. Hence, total valuation minus Alice's valuation is 0. In consequence,

$$t_A^{\text{VCG}}(\{a,b\}) = 2 - 0 = 2.$$

- For $t_A^{\text{VCG}}(\{a\})$:

  If Alice is not involved in the allocation, then it is the efficient allocation to assign Bob the bundle $\{a, b\}$, which gives him valuation 2. If Alice is involved and is assigned $a$, giving her valuation $\alpha$, then Bob's valuation from $b$ is 2. Hence, total valuation minus Alice's valuation is 2. In consequence,

  $$t_A^{\text{VCG}}(\{a\}) = 2 - 2 = 0.$$

- For $t_A^{\text{VCG}}(\{b\})$:

  If Alice is not involved in the allocation, then it is the efficient allocation to assign Bob the bundle $\{a, b\}$, which gives him valuation 2. If Alice is involved and is assigned $b$, giving her valuation $\beta$, then Bob's valuation from $a$ is 2. Hence, total valuation minus Alice's valuation is 2. In consequence,

  $$t_A^{\text{VCG}}(\{b\}) = 2 - 2 = 0.$$

- For $t_B^{\text{VCG}}(\{a, b\})$:

  If Bob is not involved in the allocation, then it is the efficient allocation to assign Alice the bundle $\{a, b\}$, which gives her valuation 2. If Bob is involved and is assigned the bundle $\{a, b\}$, giving him valuation 2, then Alice's valuation is zero. Hence, total valuation minus Bob's valuation is 0. In consequence,

  $$t_B^{\text{VCG}}(\{a, b\}) = 2 - 0 = 2.$$

- For $t_B^{\text{VCG}}(\{a\})$:

  If Bob is not involved in the allocation, then it is the efficient allocation to assign Alice the bundle $\{a, b\}$, which gives her valuation 2. If Bob is involved and is assigned $a$, giving him valuation 2, then Alice's valuation from $b$ is $\beta$. Hence, total valuation minus Bob's valuation is $\beta$. In consequence,

  $$t_B^{\text{VCG}}(\{a\}) = 2 - \beta.$$

- For $t_B^{\text{VCG}}(\{b\})$:

  If Bob is not involved in the allocation, then it is the efficient allocation to assign Alice the bundle $\{a, b\}$, which gives her valuation 2. If Bob is involved and is assigned $b$, giving him valuation $\beta$, then Bob's valuation from $a$ is $\alpha$. Hence, total valuation minus Alice's valuation is $\alpha$. In consequence,

  $$t_B^{\text{VCG}}(\{b\}) = 2 - \alpha.$$

So, the net-utility (valuation minus transfer) for Alice and Bob is given by:

- Alice's net-utility is given by $\beta$.
  She receives item $b$, giving her valuation $\beta$, and the transfer that is prescribed by the VCG-mechanism for this "bundle" is 0.

- Bob's net-utility is also given by $\beta$.
  He receives item $a$, giving him valuation 2, and the transfer that is prescribed by the VCG-mechanism for this "bundle" is $2 - \beta$.

It can easily be determined that the efficient allocation in combination with the prescribed transfer-scheme is incentive-compatible - even without referring to the general property of the VCG-mechanism outlined above. For neither of the two players, there is another bundle which offers higher net-utility than the efficient allocation with the VCG-pricing-scheme, i.e. neither Alice nor Bob have an incentive to deviate from the efficient allocation.

**General Properties of the VCG-mechanism**

Two more properties are apparent for the above stated VCG-transfer-scheme:[15]

- The price for a bundle is dependent on the identity of Alice and Bob, i.e. Alice pays different transfers for the same bundles than Bob.

- The pricing-scheme is non-linear.

Subsequent to this example, it is reasonable to pursue the following general questions:

- Is there a price-scheme such that prices are independent of the identity of the bidders?

- Is there a price-scheme such that prices are linear?

- If the answer to any of the above two questions is "yes", can we determine such an allocation?

- Once we have a procedure to determine such an allocation, can one make any statement about the efficiency of this allocation about incentive-compatibility?

These questions constitute the transition from the examples on VCG-pricing, and VCG-pricing in general, to a specific mechanism that yields identity-independent and linear prices, the so-called **Ascending Auctions**. But before this auction-format is discussed, there will be a small detour into the area of **queries**.

[15]The described properties actually hold in much more general settings that go far beyond the content of this example.

## 1.4 The Query Model

By designing a mechanism, an auctioneer aims at eliciting the bidders' private information, i.e. their valuation-functions. She can do this in the following two distinct ways. Hereby, an important role is played by the notion of a bidder's **demand set** which is defined as follows:[16]

**Definition 2 (Demand Set)** *Given a price-vector $p = (p_1, ..., p_K)$, a set $S \in 2^{\mathcal{K}}$ is said to be the* ***demand set*** *$D_n(p)$* ***of agent*** *$n$, $n \in \mathcal{N}$, iff*

$$v_n(S) - \sum_{k \in S} p_k \geqslant v_n(T) - \sum_{k \in T} p_k \qquad \forall T \in 2^{\mathcal{K}}.$$

Now, the two approaches to elicit the bidders' private information are the following:

- **Value Query**
  The presents a bundle $S \in \mathcal{K}$ to the bidder. Subsequently, the bidder reports his valuation for this bundle.

- **Demand Query**
  The presents a bundle $S \in \mathcal{K}$ to the bidder. Subsequently, the bidder reports his valuation for this bundle.

Subsequently, we will compare these two definitions, i.e. in particular we will investigate the interchangeability of the two approaches. This first proposition makes a statement about the use of demand queries to imitate value queries:

**Proposition 1** *A value query may be simulated by at most $K \cdot t$ demand queries, where $t$ is the number of bits of precision in the representation of a bundle's value.*

**Proof of Proposition 1**

So, consider a set $S = \mathcal{K}$. Wlog, $S$ can be represented as

$$S = \{1, ..., m\},$$

where, trivially, $m \leqslant K$. The proof will be conducted in two steps:

1. A bidder's valuation for the set $S$ satisfies the relation

$$v(S) = \sum_{j=1}^{m} v(\{i \in S | i \leqslant j\}) - v(\{i \in S | i < j\})$$

[16] The following definition does not exclude multiple sets in the demand of a bidder. In most of the following analysis, we will abstract from this issue for reasons of simplicity. Otherwise, one would have to make sure every time that a demand shows up and is used in the subsequent analysis that the used demand coincides with the "correct" (out of the potentially multiple possible) set chosen by the bidder.

In words, the valuation for the set $S$ can be decomposed into the sum (over $S$'s elements) of so-called marginal value queries, i.e. queries that investigate the additional valuation that a bidder obtains from one particular element.

2. For any $j \in \{1, ..., m\}$, the auctioneer can obtain the quantity

$$v(\{i \in S | i \leqslant j\}) - v(\{i \in S | i < j\})$$

by at most $t$ demand queries.

**Ad 1.**

From the assumption that $v$ satisfies "free disposal" it follows that

$$\begin{aligned} v(S) &= v(S) - v(\emptyset) \\ &= v(\{i \in S | i \leqslant m\}) - v(\{i \in S | i < 0\}) \end{aligned}$$

Hence, the desired property is a telescopic sum that adds and subtracts the elements

$$v(\{i \in S | i \leqslant 1\}), v(\{i \in S | i \leqslant 2\}), ..., v(\{i \in S | i \leqslant m-1\}),$$

making use of the fact that

$$v(\{i \in S | i \leqslant j\}) = v(\{i \in S | i < j+1\}) \qquad \text{for } j = 1, ..., m-1.$$

**Ad 2.**

In order to elicit the valuation

$$v(\{i \in S | i \leqslant j\}) - v(\{i \in S | i < j\}),$$

by a demand-query, the auctioneer can offer the following set of item-prices:

- For $i = 1, ..., j-1$: $p_i = 0$.
- For item $j$: $p_j > 0$ to be varied by the auctioneer.
- For $i = j+1, ..., m$: $p_i = \infty$.[17]

[17] From the logic of the following argument, it becomes apparent that a very high price (exceeding the highest possible valuation for any bundle from $2^{\mathcal{M}}$) will work - $\infty$ should be understood as a synonym for such a quantity.

In this case, a bidder will never demand a bundle that contains any of the items $j+1, ..., m$. Furthermore, any bundle that is demanded by a bidder will contain all of the items $1, ..., j-1$ because these items have non-negative value for the bidder but a zero-price at the same time. So, the only question is whether the bidder will demand the set $\{1, ..., j-1\}$ or the set $\{1, ..., j\}$. According to the previous two properties, these will be the sets that yield the two highest net-utilities for a bidder. The bidder will demand the set $\{1, ..., j\}$ if

$$v(\{1, .., j\}) - \sum_{i=1}^{j} p_i \geqslant v(\{1, .., j-1\}) - \sum_{i=1}^{j-1} p_i.$$

Taking the above specified price-set into account, one obtains

$$\begin{aligned} & v(\{1, .., j\}) - p_j \geqslant v(\{1, .., j-1\}) \\ \Leftrightarrow\quad & v(\{i \in S | i \leqslant j\}) - v(\{i \in S | i < j\}) \geqslant p_j \end{aligned}$$

Hence, one can start with the smallest possible price for $p_j$. Gradually increasing $p_j$, the desired valuation-difference is obtained if the bidder stops demanding the bundle $\{1, ..., j\}$. This requires exactly $t$ many demand queries. Making use of the fact that this procedure has to be repeated $m$ times to determine all marginal valuation queries, one obtains the upper bound $m \cdot t$ for the number of demand queries to obtain the result for one value-query.

□

After having investigated the exchangeability of a value query by demand queries, we will now turn to the reverse question:

**Proposition 2** *An exponential numbers of value queries may be required for simulating a single demand query.*

### Example for Proposition 2

This has been the content of Problem 1 on Assignment 3. The following example - taken from [BN05] - contains a valuation-profile satisfying appropriate properties:

- A single demand query reveals enough information for determining the optimal allocation.
- The elicitation of the optimal allocation may require an exponential number (in the number of items $K$) of value queries.

Consider an auction with two bidders that are competing for bundles from $\mathcal{K}$. Let $B$ denote a specific subset of $\mathcal{K}$ such that $|B| = \frac{K}{2}$.

Bidder 1's valuation is given by

$$v_1(S) = \begin{cases} 2|S| & \text{for every } S \in \mathcal{P}(\mathcal{K})\backslash\{B\} \\ 2|S|+2 & \text{for } S = B \end{cases}$$

That is, bidder 1 values every bundle from the set $\mathcal{K}$ according to twice the number of its elements, but she gets an extra jolt of utility from the bundle B.

Bidder 2's valuation is given by[18]

$$v_2(S) = 2|S| + 1 \text{ for every } S \in \mathcal{P}(\mathcal{K}).$$

Because it is the only allocation that achieves the maximum possible total valuation of $2|K|+3$, the following allocation to assign the bundle $B$ to agent 1 and the bundle $\mathcal{K}\backslash B$ to bidder 2 is optimal.

Now, consider a single demand query in which the price for each item in $\mathcal{K}$ will be set to $2+\epsilon$, where $\epsilon \in (0, \frac{4}{K})$. In this case, bidder 1 will demand the bundle $B$:

- $B$ is the only bundle that guarantees bidder 1 a positive net-utility because

  $$2\frac{K}{2} + 2 - 2\frac{K}{2} - \epsilon\frac{K}{2} > 0$$

  by the choice of $\epsilon$.

- All other bundle generate negative net-utility because

  $$2\frac{K}{2} - 2\frac{K}{2} - \epsilon\frac{K}{2} < 0$$

  by the choice of $\epsilon$.

So, the demand query reveals enough information (coming from bidder 1's demand-response) to determine the optimal allocation.

Determining the optimal allocation, the auctioneer can make use of the available information that there exists a set of cardinality $\frac{K}{2}$ for which bidder 1 receives an extra jolt of utility, but the auctioneer does not know the bundle $B$. Hence, it is the auctioneer's task to elicit the information which set of cardinality $\frac{K}{2}$ gives one of the bidders this extra jolt. So, the auctioneers will ask bidder 1 for her valuation of different sets of cardinality $\frac{K}{2}$, until the auctioneer has found the set $B$. Therefore, it can be concluded that the elicitation of the particular bundle $B$ requires at most $\binom{K}{\frac{K}{2}}$ value queries, the number of sets of cardinality $\frac{K}{2}$ within the set $\mathcal{K}$, having cardinality $K$. But, the number $\binom{K}{\frac{K}{2}}$ is exponential in $K$.

[18]Both valuations are designed to satisfy the assumption of "free disposal" imposed above. This property is the whole reason why the cardinality of all involved sets is multiplied by 2.

□

The results in Propositions 1 and 2 suggest that it is more appropriate to use demand queries to determine a solution to the combinatorial-auction-problem. Each of the two forms of queries may be more suitable for certain valuation-profiles of the participating bidders. But whereas it is possible to imitate a value query with a demand query in polynomial time (Proposition 1), the reverse imitation is not polynomial (Proposition 2).

Another complication from the use of value-queries arises from incentive-compatibility. Whereas a bidder in demand-query will always report the bundle that gives him the highest net-utility (which is, by definition, her demand), the bidder might not find it optimal to report her true valuation in a value-query.

For both of these reasons, attention will be restricted to special cases of demand queries in the following.

### 1.4.1 Ascending Item-Price Auction Algorithm

According to the analysis of the detour on query-models, it is more appropriate for the auctioneer - in the sense of being more robust to different valuation-profiles of the bidders - to make use of demand queries to elicit the bidders' private information, i.e. their valuation-function.

A particularly important example for this procedure is the following algorithm of an **Ascending Item-Price Auction**: [19]

**Algorithm**

- Initialization

  Set $S_n = \emptyset \ \forall n \in \mathcal{N}$, $p_k = 0 \ \forall k \in \mathcal{K}$.

- Loop

  1. For each $n \in \mathcal{N}$, let $D_n$ be the demand of agent $n$ at the following prices:

$$p_k \quad \text{for} \quad k \in S_n, \tag{3}$$

$$p_{j+\epsilon} \quad \text{for} \quad j \notin S_n. \tag{4}$$

[19] It is important to note that only item-prices will play a role in the subsequent analysis. The price of a bundle in this algorithm will be the sum of the prices of the items that are contained in the bundle.

If $S_n = D_n \ \forall n \in \mathcal{N}$, exit the loop.

2. Otherwise, pick $n \in \mathcal{N}$ such that $D_n \neq S_n$ and update:
   For every item $k \in D_n \backslash S_n$: $p_k \leftarrow p_k + \epsilon$.
   $S_n \leftarrow D_n$.
   For every bidder $j \neq n$ $S_j \leftarrow S_j \backslash D_i$

- Output

  Allocation $S_1, ..., S_K$.

Observe that this algorithm exhibits a slight asymmetry, as it picks a particular bidder in each step who is assigned his demand. But this asymmetry is ameliorated by the fact that the price-increase $\epsilon$ is assumed to be very small and by the fact that the specific bidder in every step of the algorithm is chosen at random.

**Example for an Ascending Item-Price Auction**

In order to get an idea about the ascending-auction algorithm consider the following valuation-profile:

| | $v(a,b)$ | $v(a)$ | $v(b)$ |
|---|---|---|---|
| Alice | 4 | 4 | 4 |
| Bob | 10 | 5 | 5 |

1. The algorithm starts out with the initialization $p_a = p_b = 0$ as well as $S_A = S_B = \emptyset$. At the start of the loop, prices are set to $\epsilon > 0$. At these prices, Alice is indifferent between the bundles $\{a\}$ and $\{b\}$ and, hence, will demand either $a$ or $b$. Bob will demand $\{a, b\}$. Obviously, $D_A \neq S_A$ as well as $D_B \neq S_B$.

2. Assume that the algorithm picks Bob and assigns him the items $a$ and $b$, i.e. $S_B = \{a, b\}$. Prices are set to $p_a = \epsilon$ as well as $p_B = \epsilon$. Furthermore, the allocation for Bob means that $S_A = \emptyset$ remains as in the initialization.

3. Alice's demand is determined at price $2\epsilon$ for both items. She is indifferent between the bundles $\{a\}$ and $\{b\}$ and, hence, will demand either $a$ or $b$. Bob's demand is determined at price $\epsilon$ and he will again demand the bundle $\{a, b\}$. For Bob, one observes $S_B = D_B$, but for Alice $S_A \subsetneq D_A$.

4. At this stage, assume that the algorithm picks Alice and assigns her demand to her, say the bundle $\{a\}$, i.e. $S_A = \{a\}$. Moreover, $p_A$ is set to $2\epsilon$, but $p_B$ remains at $\epsilon$. Additionally, the allocation for Alice implies that $S_B = \{b\}$.

5. Alice's demand is determined at prices $2\epsilon$ for both items. She is indifferent between the bundles $\{a\}$ and $\{b\}$ and, hence, will demand either $a$ or $b$. Bob's demand is determined at prices $p_a = 3\epsilon$ and $p_b = \epsilon$ and he will again demand the bundle $\{a, b\}$. For Bob, one observes that $S_A \subsetneq D_A$ and $S_B \subsetneq D_B$.

6. At this stage, assume that the algorithm picks Alice again and assigns her demand to her, say the bundle $\{a\}$, i.e. $S_A = \{a\}$. Moreover, $p_A$ is set to $3\epsilon$, but $p_B$ remains at $\epsilon$. Additionally, the allocation for Alice implies that $S_B = \{b\}$.

7. Alice's demand is determined at prices $p_a = 3\epsilon$ and $p_b = 2\epsilon$. In consequence, she will demand $\{b\}$. Bob's demand is determined at prices $p_a = 4\epsilon$ and $p_b = \epsilon$ and he will again demand the bundle $\{a, b\}$. For Bob, one observes that $S_A \subsetneq D_A$ and $S_B \subsetneq D_B$.

8. ...

The algorithm will terminate at prices $p_a$ and $p_b$ that satisfy

$$4 \leqslant p_a < 4 + \epsilon, 4 \leqslant p_b < 4 + \epsilon.$$

At these prices, Bob will be assigned the bundle $\{a, b\}$ and Alice will not receive a bundle. The net-utility for the two bidders is given by

$$\begin{aligned} 0 - 0 &\quad \text{for Alice} \\ 10 - p_a - p_b &\quad \text{for Bob} \end{aligned}$$

The described payments and net-utilities can be compared to those that are implied by the VCG-mechanism:

- For the bundle $\{a, b\}$, the VCG-mechanism prescribes a transfer of 4:
  - Without Bob being present, Alice would receive the bundle $\{a, b\}$ and obtain valuation 4.
  - With Bob being present, Alice receives the empty set, implying valuation 0.
  - The VCG-transfer is therefore given by $4 - 0 = 4$.
- Bob's net-utility is therefore given by $10 - 4 = 6$ which is strictly higher than the resulting net-utility from the ascending-price auction. Alice's net-utility remains the same at 0.

This example already hinges on a very essential problem of ascending item-price auctions: From the analysis of the VCG-mechanism, it becomes apparent that Bob is actually not obtaining his maximum possible net-utility from the ascending item-price auction (the net-utility from the VCG-mechanism is higher). Hence, it appears as if Bob might have an incentive to manipulate the ascending item-price auction (by not acting according to his valuation-function in every step of the algorithm) in order to make himself better off. Hence, we can conclude that the ascending item-price auction is not as neat as possible when it comes to incentive-compatibility.

This concludes the elaboration on the example of an Ascending Item-Price Auction. Now, we will turn back to the general setting.

The following property plays a very important role in the convergence of the above described general auction-mechanism:

**Definition 3 (Gross Substitutes)** *The items in the combinatorial-auction-problem are called* ***gross substitutes*** *for agent* $n \in \mathcal{N}$ *iff*
*for prices*[20] $p \leqslant q$ *and sets*

$$A \in argmax_{S \in 2^{\mathcal{K}}}\{v_n(S) - \sum_{k \in S} p_k\}$$
$$Q \in argmax_{S \in 2^{\mathcal{K}}}\{v_n(S) - \sum_{k \in S} q_k\}$$

*the following inclusion holds:*

$$\{k \in A | p_k = q_k\} \subseteq Q.$$

In words, the Gross-Substitutes-property states that items can only drop from a bidder's demand if their price is changed. This property rules out any form of complementarity.

**Example for the Gross-Substitute-property**

Consider the case of two items $a, b$. Suppose that valuations are as follows:

$$v(a) = v(b) = 3, v(\{a, b\}) = 10.$$

Furthermore, prices are given by

$$p_a = p_b = 4.$$

[20] The relation "$\geqslant$" with respect to vectors means greater than or equal to for every component of the respective vectors.

In this case, a bidder's demand will be given by $\{a, b\}$ because this is the only bundle that guarantees her strictly positive net-utility. Increasing $p_b$ to 7 and leaving $p_a$ unchanged at 3, the Gross-Substitute-property implies that the demand at the new prices has to include the item $a$. But the new demand is given by the empty-set, yielding zero net-utility, because any other bundle will give the bidder strictly negative net-utility. Contrary to the predictions of the Gross-Substitute-property, the new demand does not include item $a$ anymore although its price has not changed. Hence, the Gross-Substitute-property is violated in this example.

After having clarified the notion of Gross Substitutes in the example, we will now turn to the property of convergence of the auction-algorithm that heavily builds upon the Gross-Substitutes-property:

**Proposition 3** *If all bidders have substitute valuations, the ascending price auction will converge.*

**Lemma 3** *Suppose that all bidders have substitute valuations. Then, at any step of the algorithm, the following inclusion holds for any player* $nin\mathcal{N}$*:*

$$S_n \subset D_n$$

**Proof of Lemma 3**

At the initial step, $S_n = \emptyset$ for all bidders $n \in \mathcal{N}$. Hence, it will trivially be true that at this step

$$S_n \subseteq D_n, \qquad \forall n \in \mathcal{N}.$$

So, now consider the step of updating the sets $\{S_n\}_{n\in\mathcal{N}}$ within the loop of the algorithm. For one of the bidders, say bidder $j \in \mathcal{N}$, the sets $S_j$ and $D_j$ will coincide by the construction of the algorithm. For all other bidders $k \in \mathcal{N}\backslash\{j\}$two changes might occur, potentially affecting their $S$-sets as well as their demands:

- Items might be taken away from bidder $k$. But, from the previous step of the algorithm, the property $S_k \subseteq D_k$ has been known. Therefore, reducing the set $S_k$ will not affect the validity of the inclusion $S_k \subseteq D_k$.

- The price of items outside of $S_k$ might increase. This change does not affect the set $S_k$. Concerning the set $D_k$, it is the Gross-Substitute-property that guarantees that only those items can be removed from $D_k$ (due to the price-increase) which are outside of the set $S_k$. All items in $S_k$ have unchanged prices, so the Gross-Substitute-property requires those elements of $D_k$ that are in $S_k$ [from the previous step of the loop] to remain in $D_k$ after the price-increase, i.e. $D_k$ will still be a super-set of $S_k$.

This concludes the proof of the property that $S_n \subseteq D_n$ for all $n \in \mathcal{N}$ at all steps of the ascending-price-auction algorithm.

$\square$

**Proof of Proposition 3**

From the updating-procedure that is described by the algorithm, it becomes apparent that no item from the union of sets $S_1, ..., S_N$, $\bigcup_{n=1}^{N} S_n$, can be eliminated from this union via updating:

- One set $S_j$, $j \in \mathcal{N}$, will be set to $D_j$, where it follows from Lemma 3 that $S_j \subseteq D_j$.
- For all other $S_k$, $k \neq j$, it is only the potentially additional items that have been assigned to $S_j$ which are deleted from $S_k$.

Furthermore, the increasing prices ultimately lead to a deletions in the demanded bundles. Hence, the demand-set will ultimately be driven down to the respective $S$-sets, terminating the algorithm.

$\square$

The analysis of the ascending item-price auction will be concluded by a statement about an upper bound for the number of iterations of the algorithm. Thereby, a central role is played by the **maximum possible valuation**, which is defined as

$$v_{\max} \triangleq \max_n \max S \in 2^{\mathcal{K}}.$$

**Proposition 4** *For a fixed $\epsilon > 0$, all the objects are assigned to the bidders in at most $\frac{K \cdot v_{max}}{\epsilon}$ steps. The resulting allocation is feasible.*

It is important to observe what is NOT contained in the previous proposition:

Proposition 4 is NOT concerned with

- a statement on efficiency of the prescribed allocation;
- a statement on prices that support the prescribed allocation;
- a state of the relation between the prescribed allocation and the VCG-allocation.

It would be highly desirable to have statements like the following:

- The prescribed allocation is socially efficient.
- The prices supporting the allocation are equal to the equilibrium prices that support the $\epsilon$-Walrasian equilibrium.

These issues will be addressed in the following two sections.

## 1.5 Efficiency and Linear Programming

In this part, the focus will, again, be put on efficient solutions of the general combinatorial auction problem. In particular, the question will be investigated how the efficient solution is related to the concept of a Walrasian equilibrium for the general problem.

The determination of the efficient solution to the combinatorial auction can be written as the following **linear programming problem**:

$$\max \quad \sum_{n\in\mathcal{N}} \sum_{S\in 2^{\mathcal{K}}} x_{n,S} v_n(S) \tag{5}$$

$$\text{s.t.} \quad \sum_{n\in\mathcal{N}} \sum_{\{S|k\in S\}} x_{n,S} \leqslant 1 \qquad \forall k \in M \tag{6}$$

$$\sum_{S\in 2^{\mathcal{K}}} x_{n,S} \leqslant 1 \qquad \forall n \in \mathcal{N} \tag{7}$$

$$x_{n,S} \in \{0,1\} \qquad \forall n \in \mathcal{N}, \forall S \in 2^{\mathcal{K}}. \tag{8}$$

In words, this linear programming problem exhibits the following properties:

- The maximand is the social utility from an allocation in the combinatorial auction problem. Only those valuations play a role for which the algorithm prescribes the bundle to be assigned, i.e. $x_{n,S} = 1$.
- This set of constraints implies that every item can at most be included in one bundle that is contained in the allocation.
- This set of constraints implies that every bidder can at most be allocated one bundle.

Instead of the binary constraint on $\{x_{n,S}\}_{n\in\mathcal{N},S\in 2^{\mathcal{K}}}$, the relaxed linear programming problem simply imposes a non-negativity-constraint on the variables $\{x_{n,S}\}_{n\in\mathcal{N},S\in 2^{\mathcal{K}}}$. This relaxation allows to apply the powerful machinery of linear programming and duality (to be displayed in full generality in the next part) to the combinatorial-auction problem:[21]

The **primal program** of the **linear programming relaxation (LPR)** of the combinatorial-

[21] As it will become apparent below, the general methodology only applies to inequality-constraints, rendering the binary constraints impossible.

auction-problem is given by

$$\max \quad \sum_{n\in\mathcal{N}} \sum_{S\in 2^{\mathcal{K}}} x_{n,S} v_n(S) \tag{9}$$

$$\text{s.t.} \quad \sum_{n\in\mathcal{N}} \sum_{\{S|k\in S\}} x_{n,S} \leqslant 1 \qquad \forall k \in M \tag{10}$$

$$\sum_{S\in 2^{\mathcal{K}}} x_{n,S} \leqslant 1 \qquad \forall n \in \mathcal{N} \tag{11}$$

$$x_{n,S} \geqslant 0 \qquad \forall n \in \mathcal{N}, \forall S \in 2^{\mathcal{K}}. \tag{12}$$

The **dual program** of the **linear programming relaxation (LPR)** of the combinatorial-auction-problem is given by

$$\max_{\{u_n\}_{n\in\mathcal{N}},\{p_k\}_{k\in\mathcal{K}}} \quad \sum_{n\in\mathcal{N}} u_n \sum_{k\in\mathcal{K}} p_k \tag{13}$$

$$\text{s.t.} \quad u_i + \sum_{k\in S} \geqslant v_n(S) \qquad \forall n \in \mathcal{N}, \forall S \in \mathcal{K} \tag{14}$$

$$u_n(S) \geqslant 0 \qquad \forall n \in \mathcal{N} \tag{15}$$

$$p_k(S) \geqslant 0 \qquad \forall k \in \mathcal{K} \tag{16}$$

Importantly, the primal and the dual problem are interrelated as follows:

- The set $\{p_k\}_{k\in\mathcal{K}}$ represents the Lagrange-multipliers to the constraints in (10).
- The set $\{u_n\}_{n\in\mathcal{K}}$ represents the Lagrange-multipliers to the constraints in (11).

### 1.5.1 Canonical Linear Programming

This section deals with the three most important results that arise from the study of linear programming:

- Weak Duality
- Duality Theorem
- Complementary Slackness

All of these results are presented for very general linear programming problems. In a subsequent step, the general framework will be translated into the special case of the combinatorial-auction-problem.

The **primal linear programming program** is given by

$$Z_p \triangleq \begin{array}{ll} \max_x & c \cdot x \\ \text{s.t.} & Ax = b \\ & x \geqslant 0 \end{array} \tag{17}$$

Here, $c$ and $x$ are vectors in $\mathbb{R}^n$, $A$ is a matrix in $\mathbb{R}^{m \times n}$ and $b$ is a vector in $\mathbb{R}^m$. Hence, (17) represents $m$ constraints that are imposed on the optimization problem.

**Remark 1** *Instead of the inequality-constraint $Ax \leqslant b$, it is without loss of generality to impose the constraint $Ax = b$. This is due to the fact that the case $Ax < b$ can be circumvent by the introduction of a non-negative slack-variable $s \triangleq b - Ax$, which allows to express the constraint as*

$$Ax + s = b.$$

**Remark 2** *A solution to the primal linear programming problem will fall into one of the following three categories:*

- *The solution is infeasible, i.e. a solution to the linear programming problem does not exist.*
- *The cardinality of the set of solutions is infinite.*
- *The cardinality of the set of solutions is finite.*

The **dual program** to the above primal linear programming problem is given by

$$Z_D \triangleq \begin{array}{ll} \min_y & y \cdot b \\ \text{s.t.} & y^T A \geqslant c \\ & y \text{ is unconstrained} \end{array}$$

Here, $y$ is a vector in $\mathbb{R}^m$ and represents the Lagrange-multipliers to the $m$ constraints of the primal program in (17).

**Proposition 5 (Weak Duality)** *The optimum-values of the primal and the dual linear programming problem are related via*

$$Z_D \geqslant Z_P.$$

This proposition states that the solution from the dual problem (the minimization-problem) will always be bigger than or equal to the solution of the primal problem (the maximization problem).

**Proof of Proposition 5** Consider the vector $c$ and the set of constraints represented by $A$. Surely, one can find a linear combination $yA$ of the constraints such that

$$c \leqslant yA. \tag{18}$$

Making use of the property $x \geqslant 0$, (18) can be equivalently transformed into

$$c \cdot x \leqslant yAx. \tag{19}$$

But, according to the constraint $Ax = b$, the right hand side of (19) can be rewritten as

$$c \cdot x \leqslant y \cdot b,$$

implying the desired property $Z_D \geqslant Z_P$.

**Proposition 6 (Duality Theorem)** *If a finite solution to either the primal or the dual problem exists, then*

$$Z_D = Z_P.$$

This proposition strengthens the result from the previous proof in that it ensures - under the additional of a finite solution - that the solution to the primal and the dual problem coincide.

This proposition is a consequence of Farkas' lemma and its proof will not be stated.

**Proposition 7 (Complementary Slackness)** *If a feasible pair $(x^*, y^*)$ is optimal for the primal and the dual linear programming problem, then the following two implications hold:*

- $$x_k^* > 0 \Rightarrow \sum_n a_{nk} y_n^* = c_k.$$

- $$\sum_n a_{nk} y_n^* > c_k \Rightarrow x_k^* = 0.$$

This is a very important statement about the constraints of the two linear programming problem. It specifies when these conditions are binding and/or are slack.

**Proof of Proposition 7**

It will be shown that for all $k$ the following condition holds:

$$\left[\sum_n a_{nk} y_n^* - c_k\right] x_j^* = 0.$$

In matrix-form, the desired set of equalities read

$$y^* A x^* - c x^* = 0 \tag{20}$$

From this condition, the two implications of the proposition follow.

From Proposition 6, one obtains that

$$y^* b - c x^* = 0.$$

Now, (20) follows from the constraint $b = Ax^*$ of the primal linear programming problem.

**Example on Duality**

Consider the following problem to optimize over the variables $x_1, x_2, x_3$ subject to the stated constraints. The slack-variables are denoted by $s_1, s_2, s_3$ according to the three contraints (besides the positivity-constraints on $x_1, x_2, x_3$) that are imposed on the optimization-problem. Finally, the Lagrange-multipliers on the three constraints (again, besides the positivity-constraints) are denoted by $y_1, y_2, y_3$.

The primal linear programming problem is given by

$$\begin{array}{lccccccccccc}
\max_{x_1,x_2,x_3} & x_1 & + & 2x_2 & & & & & & & & \\
\text{s.t.} & x_1 & + & \frac{8}{3}x_2 & + & s_1 & & & & & = & 4, \\
 & x_1 & + & x_2 & & & + & s_2 & & & = & 2, \\
 & 2x_1 & & & & & & & + & s_3 & = & 3, \\
 & x_1 & , & x_2 & , & s_1 & , & s_2 & , & s_3 & \geqslant & 0.
\end{array}$$

The dual linear programming problem to this primal problem is given by

$$\begin{array}{lcccccccc}
\min & 4y_1 & + & 2y_2 & + & 3y_3 & & \\
\text{s.t.} & y_1 & + & y_2 & + & 2y_3 & \geqslant & 1, \\
 & \frac{8}{3}y_1 & + & y_2 & & & \geqslant & 2, \\
 & y_1 & , & y_2 & , & y_3 & \geqslant & 0.
\end{array}$$

The duality-theorem implies that solving the primal linear programming problem is equivalent to solving the dual linear programming problem. In particular, the solutions to both problems coincide.

### 1.5.2 Translation into the Combinatorial Auction Problem

The general setting of linear programming problems is translated as follows into the setup of the combinatorial auction problem in which we are ultimately interested in:

- vector $c$:
  - $c$ is a $(N \times 2^K) \times 1$-vector.
  - It consists of a $2^K \times 1$-part for each bidder $n = (1, ..., N)$.
  - Each $2^K \times 1$-part contains the valuations for the respective bidder for all possible bundles (there are $2^K$-many bundles).
- vector $b$:
  - $b$ is a $(N + K) \times 1$-vector.
  - It consists of $(N + K)$-many 1-entries.
  - Each 1-entry corresponds to the right hand side of one of the $(N + K)$ restrictions that are imposed on the linear programming problem.
- vector $x$:
  - $x$ is a $(N \times 2^K) \times 1$-vector.
  - It consists of a $2^K \times 1$-part for each bidder $n = (1, ..., N)$.
  - Each $2^K \times 1$-part contains the bundles that are assigned to the respective bidder (there are $2^K$-many bundles).
  - If a bundle is assigned to bidder $n \in \mathcal{N}$, the respective part of the vector $x$ contains a 1-entry.
- matrix $A$:
  - $A$ is a $(K + N) \times (N \times 2^K)$-vector.
  - The first $K$ rows correspond to the constraints that each object $k \in \{1, ..., K\}$ can only be assigned once. Now, fix $k \in \mathcal{K}$. Each element in one of the $N$-many blocks of size $2^K$ corresponds to one bundle $S \in 2^{\mathcal{K}}$. Row $k$ will have an entry 1 if the element of the row-vector corresponds to a bundle in which item $k$ is contained. All other entries in row $k$ will be zero.
  - The constraint that each item $k \in \mathcal{K}$ can only be assigned once is incorporated in the preceding structure as follows:
    Element $k$ of the product $Ax$ corresponds to the number of bundles in which

$k$ is contained.[22] The constraint that is imposed by the vector $b$ is that there is at most one such bundle. In other words, item $k$ is at most assigned in one bundle, i.e. at most to one bidder.

- The bottom $N$ rows correspond to the constraints that each bidder $n = (1, ..., N)$ can only be assigned one bundle. Now, fix $n \in \mathcal{N}$. Row $n$ consists of $N$-many blocks of size $2^K$, each block corresponding to a bidder. The $n$th block of length $2^K$ will have all entries being 1.
- The constraint that each bidder $n \in \mathcal{N}$ can only be assigned one bundle: Element $n$ of the product $Ax$ corresponds to the number of bundles that are assigned to bidder $n$.[23] The constraint that is imposed by the vector $b$ is that each bidder obtains at most one such bundle.

## 1.6 Efficient Allocations and Walrasian Equilibria

After the theory of linear programming problems has been laid out in the previous section, we will now turn to the question how the solution to the linear programming problem is related to the optimality of the choice of a bundle for each agent. Such an allocation combined with item-prices that are prescribed by the auctioneer, in which each agent obtains the bundle which is optimal (in the sense of maximizing his net-utility) is called a **Walrasian equilibrium**:

**Definition 4 (Walrasian Equilibrium)** *A pair $(S^*, p^*)$, consisting of an allocation $S^* = (S_1^*, ..., S_N^*)$ and a set of nonnegative prices $p^* = (p_1^*, ..., p_K^*)$, is called a* ***Walrasian equilibrium*** *if at prices $p^*$ the following two conditions hold:*

1. *$S^*$ is a feasible allocation.*
2. *$S_n^*$ is agent $n$'s demand at $p^*$, $n \in \mathcal{N}$. Moreover, any $j \notin \cup_{n=1}^N S_n$ satisfies $p_j^* = 0$.*[24]

In a first step, it will be shown that, at a Walrasian equilibrium, we have already arrived at the socially optimal solution.

[22] The matrix-row in $A$ as well as the vector $x$ only contain 0 or 1 as elements. That is, the $k$th element of the product "counts" the number of times in which a 1-entry of the row-vector in $A$ "meets" a 1-entry of the vector $x$, i.e. the number of times that a bundle is assigned that contains item $k$.

[23] The matrix-row in $A$ as well as the vector $x$ only contain 0 or 1 as elements. That is, the $n$th element of the product "counts" the number of times in which a 1-entry of the row-vector in $A$ "meets" a 1-entry of the vector $x$, i.e. the number of bundles that are assigned to bidder $n$.

[24] In words, every object that is not assigned to any of the bidders ($j \notin \cup_{n=1}^N S_n$) has zero price. This corresponds to the most extreme situation for object 0. Although the object is as cheap as possible (it has zero price), it is in no bidders' interest to obtain the good.

**Theorem 1 (First Welfare Theorem)** *If the pair $(S^*, p^*)$ is a Walrasian equilibrium, then the allocation $S^*$ maximizes social welfare over all fractional allocations, i.e. all solutions to the relaxed linear programming problem.*[25] *That is, for all feasible solutions $\{x_{n,S}\}_{n\in\mathcal{N},S\in 2^{\mathcal{K}}}$ to the LPR*

$$\sum_{n\mathcal{N}} v_n(S_n^*) \geqslant \sum_{n\mathcal{N},S\in 2^{\mathcal{K}}} x_{i,S} v_i(S).$$

**Proof of Theorem 1**

The statement of the theorem compares the allocation $S^*$ from the Walrasian equilibrium-pair $(S^*, p^*)$ to any allocation $S$ that arises as the solution to the linear programming problem stated above.

By the definition of a Walrasian equilibrium, $S_n^*$ is the demand of agent $n$ at prices $p^*$, where $n \in \mathcal{N}$. That is, for all $n \in \mathcal{N}$, one has

$$v_n(S_n^*) - \sum_{k\in S_n^*} p_k^* \geqslant v_n(S) - \sum_{k\in S} p_k^* \qquad \forall S \in 2^{\mathcal{K}}. \tag{21}$$

Feasibility of any solution to the linear programming problem implies that

$$\sum_{S\in 2^{\mathcal{K}}} x_{n,S} \leqslant 1. \tag{22}$$

Making use of (22) allows, for any $n \in \mathcal{N}$, to rewrite (21) as follows

$$v_n(S_n^*) - \sum_{k\in S_n^*} p_k^* \geqslant \sum_{S\in 2^{\mathcal{K}}} x_{n,S} \left( v_n(S) - \sum_{k\in S} p_k^* \right). \tag{23}$$

Because (23) holds for all $n \in \mathcal{N}$, it follows that

$$\sum_{n\in\mathcal{N}} v_n(S_n^*) - \sum_{k\in S_n^*} p_k^* \geqslant \sum_{n\in\mathcal{N}} \sum_{S\in 2^{\mathcal{K}}} x_{n,S} \left( v_n(S) - \sum_{k\in S} p_k^* \right). \tag{24}$$

The property

$$\sum_{n\in\mathcal{N}} v_n(S_n^*) \geqslant \sum_{n\in\mathcal{N}} \sum_{S\in 2^{\mathcal{K}}} x_{n,S} v_n(S),$$

[25] This is a version of the First Welfare Theorem that is adapted to linear programming problems in the context of combinatorial auction problems. Its proof is especially tailored to fit the needs of the combinatorial-auction-environment. What economists usually refer to as the First Welfare Theorem is the following statement:
*Any Walrasian equilibrium is Pareto-efficient.*
An allocation is Pareto-efficient if no agent can be made better off without making any other agent worse off (both in terms of the utility that an agent receives from an allocation).

which concludes the proof of Theorem 1, can be deduced from (24) if the following property can be established:

$$\sum_{n\in\mathcal{N}}\sum_{k\in S_n^*} p_k^* \geqslant \sum_{n\in\mathcal{N}}\sum_{S\in 2^{\mathcal{K}}} x_{n,S}\sum_{k\in S} p_k^*. \tag{25}$$

One obtains (25) as follows:

- Because $S^*$ is a feasible allocation of the set $K$, the left-hand-side of (25) is equal to $\sum_{k=1}^{K} p_k^*$.
- Feasibility of the LPR involves the condition

  $$\sum_{n\in\mathcal{N}}\sum_{\{S|k\in S\}} x_{n,S} \leqslant 1 \qquad \forall k \in K.$$

  Hence, the right-hand side of (25) satisfies

  $$\sum_{n\in\mathcal{N}}\sum_{S\in 2^{\mathcal{K}}} x_{n,S}\sum_{k\in S} p_k^* \leqslant \sum_{k=1}^{K} p_k^*.$$

□

After having seen that Walrasian equilibria actually attain the socially optimal net-utility-level among the class of solutions to the LPR, we will now prove the reverse implication, i.e. we will characterize a solution of the linear programming relaxation as a Walrasian equilibrium. That is, at each of the solutions to the LPR, each player maximizes his own self-interest, which is a very important stability-property of the solutions to the LPR. Hence, no player has a unilateral interest to move away from the solution to the LPR.

**Theorem 2 (Second Welfare Theorem)** *If an integral solution exists for the LPR, then a Walrasian equilibrium $(S^*, p^*)$ exists, where $S^*$ is the given solution and $p^*$ arises from the solution to the dual problem.*[26]

[26] This is a version of the Second Welfare Theorem that is adapted to linear programming problems in the context of combinatorial auction problems. Its proof is especially tailored to fit the needs of the combinatorial-auction-environment. What economists usually refer to as the Second Welfare Theorem is the following statement:
*For any Pareto-efficient allocation there exist prices $p^*$ such that the pair $(S^*, p^*)$ is a Walrasian equilibrium.*

**Proof of Theorem 2**

The solution to the linear programming relaxation defines a feasible allocation $S^* = (S_1^*, ..., S_N^*)$. By the duality theorem, one obtains the existence of a solution $(p^*, q^*) = (p_1^*, ..., p_N^*, q_1^*, ..., q_K^*)$ to the dual linear programming relaxation. Now, it will be shown that the pair $(S^*, p^*)$ constitutes a Walrasian equilibrium. This property will be derived from complementary slackness which is necessary and sufficient for the optimality of solutions to the primal as well as the dual linear programming problem.

- Given that $x_{n,S_n^*} > 0$ for some $n \in \mathcal{N}$,[27] complementary slackness implies that

$$q_n^* = v_n(S_n^*) - \sum_{j \in S_n^*} p_j^*$$

- Due to the fact that the condition

$$q_n^* = v_n(S_n^*) - \sum_{j \in S_n^*} p_j^*$$

  holds with equality, it follows that for any other bundle $S$

$$v_n(S_n^*) - \sum_{j \in S_n^*} p_j^* \geqslant v_n(S) - \sum_{j \in S} p_j^*.$$

- Finally, it follows from another application of complementary slackness that in the case

$$\sum_{n \in \mathcal{N}, S | j \in S} x_{n,S} < 1,$$

  i.e. in the situation in which item $j$ is not allocated, one obtains $p_j^* = 0$.

In sum, complementary slackness implies all properties to conclude that the pair $(S^*, p^*)$ is a Walrasian equilibrium.

□

The following corollary summarizes the content of the previous two theorem for the case of our linear-programming-analysis of the combinatorial-auction-problem. It states a one-to-one-relation between Walrasian equilibria and integral solutions to the linear programming relaxation:

**Corollary 1** *A Walrasian equilibrium exists in a combinatorial-auction-environment if and only if the corresponding linear programming relaxation admits an integral optimal solution.*

[27] By construction, this means that $x_{n,S_n^*} = 1$, i.e. bundle $S_n^*$ is assigned to bidder $n$.

## 1.7 Ascending Bundle-Price Auctions

In order to complete the treatment of combinatorial auction problems, we will cover Bundle-Price Auctions as an alternative mechanism to Item-Price Auctions that have been covered before. This section is mainly designed as a cursory overview of the results that arise in the context of this auction-format. Hence, only the relevant definitions, two algorithms and the most important results are stated and all proofs are omitted.[28]

Observe the following important difference in the approach of the analysis:

- In the previous analysis, we have always considered a set of prices for each item. Bundle-prices have been obtained as the sum of the prices of the items that are contained in the bundle.
- Now, we are assuming a set of prices of cardinality $2^K$, i.e. a separate price for each bundle.

Interestingly, the results concerning the comparison between value and demand queries as well as the efficiency analysis of solutions to the relaxed linear programming problems continue to hold under this change.

### 1.7.1 Personalized Bundle-Prices

**Algorithm**

- Initialization
  $\forall n \in \mathcal{N}$, set $p_n(S) = 0 \ \forall S \in \mathcal{K}$.
- Loop
  1. Determine the feasible allocation $T = (T_1, ..., T_N)$ that maximizes revenue at current prices, i.e. $T$ satisfies
  $$\sum_{n=1}^{N} p_n(T_n) \geqslant \sum_{n=1}^{N} p_n(Y_n)$$
  for any other feasible allocation $Y = (Y_1, ..., Y_N)$.
  (For bundles whose prices are zero, the auctioneer is indifferent between allocating the bundle to a bidder or not. In this case, we will assume that the auctioneer will always decide not to allocate the bundle. In consequence, we can assume that the prices for all allocated bundles are strictly positive.)

[28] The proofs can be found in chapter 11 of the textbook, [NRTV08]. Moreover, one property is taken from Assignment 3 whose solution is available on the class-webpage.

2. Determine the set of bidders $L$ satisfying

$$L \triangleq \{n | T_n = \emptyset\}.$$

These are the bidders which are not allocated any bundle by the auctioneer who aims at revenue-maximization, the so-called losing bidders.

3. For every $n \in L$, determine $D_n \triangleq D_n(p_n)$, bidder $n$'s demand at his current price-function $p_n$.
   - If $D_n = \emptyset$ for all $n \in L$, then terminate and output $S^* = T$.
   - For all $n \in L$ for which $D_n \neq \emptyset$, set

$$p_n(D_n) \leftarrow p_n(D_n) + \epsilon.$$

   That is, the price for those objects that are in bidder $n$'s demand is raised by $\epsilon$.[29]

**Definition 5** *A set personalized bundle-price function $(p_1(\cdot), p_2(\cdot), ..., p_N(\cdot))$ in combination to an allocation $S = (S_1, ..., S_N)$ is called a* ***competitive equilibrium in a bundle-price setting*** *iff:*

1. *For every bidder $n \in \mathcal{N}$, $S_n$ is bidder $n$'s demand at prices $p_n(\cdot)$, i.e. for any other bundle $T \subseteq \mathcal{K}$*

$$v_n(S_n) - p_n(S_n) \geqslant v_n(T) - p_n(T).$$

2. *The allocation $S$ maximizes seller's revenue given price-functions $(p_1(\cdot), p_2(\cdot), ..., p_N(\cdot))$, i.e. for any other feasible allocation $T_1, ..., T_n$ one obtains*

$$\sum_{i=1}^{n} p_n(S_n) \geqslant \sum_{i=1}^{n} p_n(T_n)$$

**Proposition 8** *In any competitive equilibrium $((p_1(\cdot), p_2(\cdot), ..., p_N(\cdot)), S)$ in a bundle-price setting, the allocation $S$ maximizes social welfare.*

**Definition 6** *A bundle $S \in 2^{\mathcal{K}}$ is an* ***$\epsilon$-demand under the price-function $p_n$ for bidder*** *$n$ iff*
*for any other bundle $T \in 2^{\mathcal{K}}$*

$$v_n(S) - p_n(S) \geqslant v_n(T) - p_n(T) - \epsilon.$$

[29] Of course, this gives the auctioneer the possibility to increase its revenue and, at the same time, makes it possible that bidder $n$ will no longer demand the particular bundle.

**Definition 7** *A set personalized bundle-price function* $(p_1(\cdot), p_2(\cdot), ..., p_N(\cdot))$ *in combination to an allocation* $S = (S_1, ..., S_N)$ *is called an* ***$\epsilon$-competitive equilibrium in a bundle-price setting*** *iff:*

1. *For every bidder* $n \in \mathcal{N}$*,* $S_n$ *is bidder* $n$*'s* $\epsilon$*-demand at prices* $p_n(\cdot)$*.*

2. *The allocation* $S$ *maximizes seller's revenue given price-functions* $(p_1(\cdot), p_2(\cdot), ..., p_N(\cdot))$*, i.e. for any other feasible allocation* $T_1, ..., T_n$ *one obtains*

$$\sum_{i=1}^{n} p_n(S_n) \geqslant \sum_{i=1}^{n} p_n(T_n)$$

**Proposition 9** *An ascending bundle-price auction with bidder-specific prices terminates with an* $\epsilon$*-competitive equilibrium. The welfare obtained from this auction is within* $n\epsilon$ *from the optimal social welfare.*

### 1.7.2 Anonymous Bundle-Prices

**Algorithm**

- Initialization
  $\forall n \in \mathcal{N}$, set $p(S) = 0 \ \forall S \in \mathcal{K}$.

- Loop

  1. Determine the feasible allocation $T = (T_1, ..., T_N)$ that satisfies the following two conditions:
     It maximizes revenue at current prices, i.e. $T$ satisfies

     $$\sum_{n=1}^{N} p(T_n) \geqslant \sum_{n=1}^{N} p(Y_n)$$

     among the set of allocations $Y = (Y_1, ..., Y_N)$ satisfying

     $$v_n(Y_n) - p(Y_n) \geqslant 0$$

     The last condition guarantees that bidders who are assigned a bundle by the auctioneer actually prefer possessing this bundle over not obtaining any bundle at the current price schedule $p(\cdot)$.
     (For bundles whose prices are zero, the auctioneer is indifferent between allocating the bundle to a bidder or not. In this case, we will assume that the auctioneer will always decide not to allocate the bundle. In consequence, we can assume that the prices for all allocated bundles are strictly positive.)

2. For all $n \in \mathcal{N}$, the

- Determine the set of bidders $L$ satisfying

$$L \triangleq \{n | T_n = \emptyset\}.$$

  These are the bidders which are not allocated any bundle by the auctioneer who aims at revenue-maximization, the so-called losing bidders.

- For every $n \in L$, determine $D_n \triangleq D_n(p)$, bidder $n$'s demand at the current price-schedule $p(\cdot)$.
  - If $D_n = \emptyset$ for all $n \in L$, then terminate and output $S^* = T$.
  - For all $n \in L$ for which $D_n \neq \emptyset$, set

$$p(D_n) \leftarrow p(D_n) + \epsilon.$$

    That is, the price for those objects that are in bidder $n$'s demand is raised by $\epsilon$.[30]

**Proposition 10** *When the valuations of all participating bidders are super-additive, the anonymous price-variant of the bundle-price ascending auction terminates with the socially efficient allocation.*

# References

[BN05] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions i: Demand queries. *Discusiion paper no. 381, The Center for the Study of Rationality, The Hebrew University*, 2005.

[NRTV08] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, 2008.

[30] Of course, this gives the auctioneer the possibility to increase its revenue and, at the same time, makes it possible that bidder $n$ will no longer demand the particular bundle.