Control and Cybernetics
vol. **41** (2012) No. 4

Bidding languages for auctions of divisible goods*

by

Mariusz Kaleta

Warsaw University of Technology
Institute of Control and Computation Engineering

Abstract: Bidding languages are well-defined for combinatorial auctions. However, the auctions of divisible goods are quite common in practice. In contrast to combinatorial auctions, the feasible volumes of the offers are continuous in the case of divisible commodities. Such auctions are called continuous auctions. In the paper we introduce three families of bidding languages for continuous auctions. They are based on the concepts derived from the combinatorial auctions. We generalize the language families based on goods, bids, and some mixture of both of them, to the continuous case. We also analyze fundamental properties of the new languages. Simple examples, reflecting the complementarity and substitutability, are provided with exemplary representations in different languages.

Keywords: bidding languages, auctions, market mechanism design

1. Introduction

Market mechanisms are entering into new areas of life. From the beginning of this century one can observe rapidly increasing number of organized markets, mainly auctions, both on the retail and wholesale markets. Electronic auction mechanisms are visible in the Internet at specialized web pages, including those most recognizable, like eBay. But they are also being introduced into other web services, e.g. social networks, where availability of additional information enables the new functionalities. A trend to real-time operation is one of the main drivers for electronic trade on the wholesale markets. The increasingly competitive conditions have led to several management concepts like real-time enterprises or dynamic supply chains. Besides more competitive conditions, the nature of traded commodities forces the need of trading in nearly real time, that is, electronic markets are entering into markets of real-time commodities, e.g. telecommunication bandwidth, electrical energy, and so on. As a result of these trends, the new requirements and needs for new functionalities are

*Submitted: October 2012; Accepted: November 2012

clear. The question how to built, integrate and manage the information systems for running the electronic auctions is addressed in a stream of studies, e.g. Benyoucef and Pringadi (2006), Kaleta and Traczyk (2012), Rolli and Eberhart (2005). Current research is focused on designing methodologies for auction systems. The reusability, extensibility, simplicity of deployment are the key aspects of the methodologies. Many works propose multi-agent approaches, e.g. Rolli and Eberhart (2005), which make incentives for strong automation of the trade. For example, in Benyoucef and Pringadi (2006) a BPEL based approach is proposed to achieve high flexibility in auction protocols.

One of the important problems in the context of electronic auction system designs is a bidding language (Nisan, 2000). A bidding language is a tool to represent valuations of a given agent to the market. From the point of view of particular agent, the overall evaluation of the market system strongly depends on the functionality that this tool gives to the agents. There are three criteria for evaluation of the bidding languages. The first is the *expressive power*, that is, what kind of valuations an agent can express using the particular bidding language. The second criterion is its *succinctness* – how verbose is a given language and as a result, how much memory it requires to store the bids. And the last one is the *complexity* and logical foundations of a given language from the agent point of view.

Expressive, but simple languages may involve exponential growth of data describing a bid when the number of commodities increases. A utility function of a given agent often has some logical structure. The question is, whether a given language enables to exploit this structure in the bids. If so, the bids should be more convenient for the agents. Moreover, it usually leads to more succinct language. The succinctness is important for communication protocols and data management.

There is a wide stream of studies devoted to bidding languages for combinatorial auctions (Boutilier and Hoos, 2001; Lehmann at al., 2006; Nisan, 2000). We discuss the achievements in this field further in the paper. However, in the current literature, relatively little attention has been paid to auctions of divisible goods, which are quite common in practice. We call such auctions the continuous auctions since the feasible volumes of bids are continuous in contrast to combinatorial auctions.

In the paper we focus on bidding languages for continuous auctions. We introduce the classes of bidding languages for continuous auctions. In Section 2 we introduce basic notions related to combinatorial and continuous auctions. Then, we describe bidding languages for combinatorial auctions in the next section. Bidding languages for continuous auctions are introduced in Section 4. After introducing the new classes of bidding languages we discuss their basic properties in Section 5. We close the paper with a summary and the directions of further research.

2. Auctions

On the grounds of mechanism theory, an auction can be perceived as a mechanism. In a mechanism there is a set of agents who participate in a certain game defined by the mechanism rules. The agents send some signals to the mechanism. Under certain conditions, the mechanism is triggered to compute a temporary market equilibrium and find the winners. After that, the results are sent back by the mechanism to the agents. Each agent obtains information about commodities allocated to him and related payments, usually in the form of unit price.

Let $\mathcal{C} = \{1, 2, \dots, C\}$ be a set of commodities being traded. We assume that for a given payment p_j and allocation $x_j = (x_1, \dots, x_C)_j$, the agent j is able to specify his willingness: accept or not the allocation x_j with payment p_j. Agent j-th *preferences* $w(p_j)$ for a given payment p_j is a set of all allocations that agent j is willing to accept. Preference is monotone with respect to price if $w(p_j) \in w(p'_j)$ for every $p'_j > p_j$ or $w(p'_j) \in w(p_j)$ for every $p'_j > p_j$. Fig. 1 illustrates monotone and non-monotone preferences.

The preferences are not convenient for further computations. We assume, that preferences are rational and hence they can be represented by some utility functions.

Instead of the preferences or their utility representations, the agents need to present their *valuations* to the mechanism. The valuations may differ from the preferences. One reason for this is that the agents play a game and act strategically, thus they do not want to reveal their private preferences. Another reason is that the agents may even not know their preferences accurately or preference representation in the form of valuations can be inaccurate.

The valuations may be expressed by the utility function $v(x)$, where $x = (x_1, \dots, x_C)$ is a vector of commodity quantities and $v(x)$ is a marginal unit price. We implicitly assume here that the bidder's valuation depends only on the set of goods he wins. It is known as "no externalities" assumption.

The utility function expresses the willingness to accept a vector of commodity quantities x if price is $v(x)$ or better. The following proposition can be proved.

PROPOSITION 1 *Valuations defined by $v(x)$ may express only monotone preferences.*

In the rest of the paper we assume that each agent has its own utility function and the valuations are expressed by the utility functions.

The agents present their valuations to the mechanism via bids*. An agent uses a given language to show his valuation to the market via his bid. A bid encodes a valuation v that a given agent has. In general, it may be different from his utility function.

*Although there are some subtle semantic differences in the following notions: *commodities* and *goods*, *bids* and *offers*, we treat them as synonymous in the paper

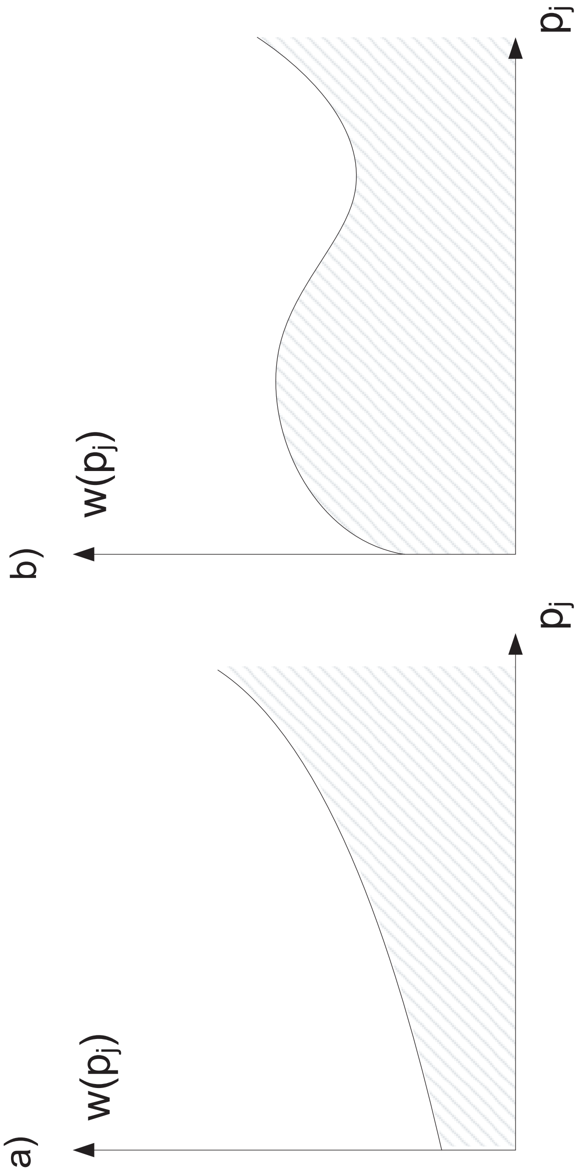

Figure 1. Monotone (a) and non-monotone (b) preferences

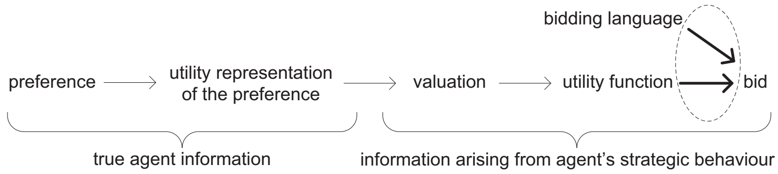

Figure 2. A sequence of transitions from preference to a bid; in the last transition marked with the dotted ellipse, a bidding language is used

Fig. 2 presents the main notions introduced and relations between them. Some inaccuracy may happen with each transition in a sequence shown in the figure. In the paper we focus on the last transition in which a valuation in the form of utility function, that an agent wants to present to the mechanism, is transformed into a bid using a given bidding language.

Computation of temporary market equilibrium requires determining the volumes of winning bids and payments. Usually, it is done in two stages. First, the volumes of winning bids, and next, the market prices and related cash flows, are calculated. The problem of finding the winning bids is called the Winner Determination Problem (WDP) (for Network Winner Determination Problem in case of continuous and combinatorial auctions see Kaleta, 2012).

In combinatorial auctions the agents may submit the bids on combinations of commodities. Assume, that $\mathcal{C} = \{1, 2, \ldots, C\}$ is a set of commodities being traded. We also assume the following form of the valuation functions in combinatorial auctions: $v : 2^{\mathcal{C}} \to \mathbb{R}$. Later in this paper we assume that v is normalized, which means that $v(\{\}) = 0$, and v is monotonic. Monotonicity means that if $X \subseteq Y$ then $v(X) \leq v(Y)$. We also assume the free disposal property which means that there is no cost of excess allocation. The Winner Determination Problem is defined as follows.

Definition 1 *(Winner Determination Problem, WDP) The seller has a set of commodities, $\mathcal{C} = \{1, 2, \ldots, C\}$, to sell. The buyers submit set of offers (bids) $m \in \mathcal{B} = \{1, 2, \ldots, B\}$. An offer m encodes valuation v_m. An allocation of commodities is denoted by $X_m(S) \in \{0, 1\}$, where $X_m(S)$ is equal to one if bundle $S \subseteq \mathcal{C}$ is allocated to the bid m. The Winner Determination Problem (WDP) is to find an allocation of commodities to buying offers which is revenue-maximizing under the constraints that no commodity is allocated more than once.*

In the above, classical formulation of the WDP, it is assumed that each commodity can be allocated to at most one buying offer (Lehmann at al., 2006).

In continuous version of WDP, each bid can be accepted partially, and commodities are perfectly divisible. The seller has volume p_c^{max} of commodity c, $c \in \mathcal{C}$. A valuation function in continuous case is a function $v_m : \mathbb{R}^C \to \mathbb{R}$. It is defined over an allocations space, where an allocation is a vector of commodity levels, $(X)_m \in \mathbb{R}^C, c \in \mathcal{C}$, where $(X_c)_m$ is a volume of commodity c allocated to the offer m. We assume normalization and monotonicity of a valuation function. A valuation is normalized if $v([0,\dots,0]) = 0$. A valuation is monotonic if for two allocation vectors $X = (x)$ and $Y = (y)$ if $x_c \le y_c \ \forall_c$ then $v(X) \le v(Y)$. The following continuous WDP is to be solved in a continuous auction case.

Definition 2 *(Continuous Winner Determination Problem, cWDP) The seller has a set of given volumes of commodities, $p_c^{max}, c \in \mathcal{C}$, to sell, where $\mathcal{C} = \{1, 2, \dots, C\}$ is a a set of commodities, and p_c^{max} is a volume of commodity c. The buyers submit set of offers (bids) $m \in \mathcal{B} = \{1, 2, \dots, B\}$. Maximum volumes are explicitly given in the buying offers. An offer m encodes valuation v_m. An allocation is a vector of commodity levels denoted by $(X_c)_m \in \mathbb{R}^C, c \in \mathcal{C}, X_c \in \mathcal{R}$, where $(X_c)_m$ is the level of commodity c allocated to offer m. The Continuous Winner Determination Problem (cWDP) is to find an allocation of commodities to buying offers which is revenue-maximizing under the constraints that no maximal volume in the offers nor p_c^{max} are exceeded.*

In Definitions 1 and 2 there is nothing on how the valuation is encoded in a bid. Offer encoding is a task of a bidding language. Without specifying a bidding language the description of the market mechanism is not full and the market cannot be run.

3. Bidding languages for combinatorial auctions

An atomic bid is a tuple $\langle \mathcal{G}, p \rangle$, $\mathcal{G} \subseteq \mathcal{C}$, where $\mathcal{G}$ is a bundle of goods, and $p \in \mathbb{R}^+$ is a bid price. The atomic bid represents the following valuation:

$$v(X) = \begin{cases} p & \text{if } X \subseteq G \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

Further, we will identify a bid with the associated valuation.

The simplest way to express an agent's valuation is to assign a value to each combination of the commodities and submit to the mechanism a set of atomic bids. Obviously, every valuation can be expressed in that way, but it may require exponential number of atomic bids. It is not simple for an agent since it does not reflect and exploit the structure of the valuation. In combinatorial auctions a bidding language is used to represent valuations over the bundles of goods more compactly and somehow with respect to structure of the valuation. A language gives a semantic meaning to the well-formed syntactic elements.

Two notions – *complementarity* and *substitutability* – are related to the structure of the valuation. Two goods, a and b, are complementary if the valuation of both of them together is equal or greater than the sum of valuations of individual goods:

$$v(\{a\}) + v(\{b\}) \le v(\{a, b\}). \tag{2}$$

Example 1 *The following valuation function features complementarity*

x	$v(x)$
$\{a\}$	2
$\{b\}$	3
$\{a,b\}$	6

Complementarity is perfect if $v(\{a\}) + v(\{b\}) = 0$, that is, both goods are needed to create any value.

Two goods are perfectly substitutable if the valuation of both of them is equal to sum of valuations of individual goods:

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

Example 2 *The following valuation is a case of perfect substitutability*

x	$v(x)$
$\{a\}$	2
$\{b\}$	3
$\{a,b\}$	5

Imperfect substitution (substitution in short) is defined by the following relation:

$$v(\{a\}) + v(\{b\}) > v(\{a, b\}). \tag{4}$$

Example 3 *The following valuation function features (imperfect) substitutability*

x	$v(x)$
$\{a\}$	2
$\{b\}$	3
$\{a,b\}$	4

Notice that the free disposal assumption means that $v(a) + v(b) \geq v(a)$ and $v(a) + v(b) \geq v(b)$.

Three families of bidding languages for combinatorial auctions are considered in the literature (Boutilier and Hoos, 2001). The first family, denoted $\mathcal{L}_G$, assumes that price and logical formula of commodities are provided in a bid. For a given allocation the formula can be evaluated as true or false. If it is evaluated as true, then the bid is accepted and paid at least the price given in the bid. Some level of conciseness is achieved because the logical expressions can be used instead of enumeration of all desired combinations of goods. If desired valuation for each combination is the same, then the expression can be used to build one bid. Thus, $\mathcal{L}_G$ allows to express some agent preferences in a natural way. In the case of perfect substitutability it exploits the logical structure of the preferences and leads to quite concise bid. An exemplary language from the $\mathcal{L}_G$ family is proposed in Hoos and Boutilier (2000). The authors have introduced the $\mathcal{L}_G^{pos}$ language and its variants. $\mathcal{L}_G^{pos}$ assumes that no negation can be used in the formulas. It allows to express some typical valuations in a natural way. In an example formulated in Hoos and Boutilier (2000), an agent desires either a_1 or b_1, and a_2 or b_2, and a_3 or b_3. The case is captured in a straightforward way as follows: $\langle (a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge (a_3 \vee b_3), p\rangle$, where p is a price for any of the three feasible bundles. However, $\mathcal{L}_G$ does not cover value increase for adding more commodities to a bid with already satisfied logical formula. Thus, the perfectly complementary goods a and b are captured by the following bid: $\langle a \wedge b, p\rangle$. But general (imperfect) complementarity and substitutability cannot be captured with this language. Notice that even perfect substitutability cannot be expressed with the following bid $\langle a \vee b, p\rangle$, since if a and b are allocated to the bid, then the payment is 4 instead of 8 or more (see equation (2)). Extending the $\mathcal{L}_G^{pos}$ with so called *dummy goods* increases its expressiveness (Fujishima et al., 1999), but it becomes more verbose and closer to full enumeration of all combinations. Another disadvantage of $\mathcal{L}_G$ is revealed when some combinations of goods have different valuations. Assume that value of obtaining a and b is 4, and value of a and c is 5. In this case individual bids must be prepared: $\langle a \wedge b, 4\rangle$, $\langle a \wedge c, 5\rangle$.

In the second family of bidding languages, $\mathcal{L}_B$, the logical formulas of atomic bids are provided. Any language from family $\mathcal{L}_B$ combines the atomic bids in a logical formula and assigns a price. But in contrast to $\mathcal{L}_G$, not the whole formula is evaluated as true or false, but just individual atomic bids are checked to be satisfied or not. The price to be paid is a result of some function of atomic bid prices with respect to the logical formula of the bids.

$\mathcal{L}_B^{or}$ language is an example from the $\mathcal{L}_B$ family. In $\mathcal{L}_B^{or}$ language the logical operator OR is used to bind the atomic bids. If two bids with valuations v_1 and v_2 are combined with OR operator, then the resulting valuation is as follows (Nisan, 2000):

$$(v_1 \ OR \ v_2)(X) = \max_{X_1, X_2 \subseteq X, X_1 \cap X_2 = \emptyset} (v_1(X_1) + v_2(X_2)). \tag{5}$$

The valuation in the Example 1 can be presented with the following set of bids: $v_1 = \langle \{a\}, 2\rangle$, $v_2 = \langle \{b\}, 3\rangle$, $v_3 = \langle \{a, b\}, 6\rangle$, and OR bid $\langle v_1 \ OR \ v_2 \ OR \ v_3\rangle$. If both commodities are allocated to the OR bid, then the maximum defined in (5) is obtained when v_3 is satisfied and is equal to 6. Perfect substitutability can be directly modeled by joining two atomic bids with OR operator. However, any imperfect substitutability cannot be expressed in $\mathcal{L}_B^{or}$ since maximization in (5) would not let choosing a "worse" combination of $a \wedge b$. $\mathcal{L}_B^{or}$ is not fully expressive, since it cannot represent any valuation with substitutability.

Another language from $\mathcal{L}_B$ family, the XOR language $\mathcal{L}_B^{xor}$, is fully expressive. XOR combination of two valuations v_1 and v_2 defines the following valuation (Nisan, 2000):

$$(v_1 \ XOR \ v_2)(X) = \max\{v_1(X), v_2(X)\}. \tag{6}$$

The following bids can be formulated in case of Example 1: $v_1 = \langle\{a\}, 2\rangle$, $v_2 = \langle\{b\}, 3\rangle$, $v_3 = \langle\{a, b\}, 6\rangle$, and XOR bid $\langle v_1\ XOR\ v_2\ XOR\ v_3\rangle$. In case of allocation of $\{a, b\}$, the value 6 will be chosen due to maximization in (6). In Example 3 of imperfect substitution, also three valuations $v_1 = \langle\{a\}, 2\rangle$, $v_2 = \langle\{b\}, 3\rangle$, $v_3 = \langle\{a, b\}, 4\rangle$ must be submitted. For the allocation $\{a, b\}$ the maximal value is obtained when the goods are allocated to v_3. Notice that due to the free disposal assumption, value of v_3 will be always greater than of v_1 and v_2.

Although XOR-bids can represent any valuations, the exponential number of atomic bids may be needed in case of additive valuations, while in the same case the OR language requires much less number of atomic bids (Nisan, 2000). Consider the following valuation with perfect substitutability, that an agent wants to express:

x	$v(x)$
$\{a\}$	4
$\{b\}$	4
$\{a, b\}$	8

An agent must define the following atomic bids: $\langle\{a\}, 4\rangle$, $\langle\{b\}, 4\rangle$. They are sufficient to express the valuation in $\mathcal{L}_B^{or}$ language: $\langle v_1\ OR\ v_2\rangle$. But in $\mathcal{L}_B^{xor}$ language the agent must define atomic bids for every subset of commodities, so $\langle\{a, b\}, 4\rangle$ must be introduced and the final offer should be as follows: $\langle v_1\ XOR\ v_2\ XOR\ v_3\rangle$.

Next three languages in $\mathcal{L}_B$ family arise from the attempt to combine advantages of the two previously mentioned languages. In OR-of-XOR language a bid comprises OR combinations of XOR combinations of atomic bids. On the contrary, in XOR-of-OR language there are XOR combinations of OR combinations of atomic bids. OR/XOR language is the most general of those, since it allows for any combination of ORs and XORs. Each of these languages is fully expressive, but no one dominates in terms of conciseness and simplicity.

Nisan has also proposed a variant of OR language with phantom items – OR* language. An agent is allowed to include in his bids the phantom items which enable to simulate XOR language. OR* is fully expressive and it has also good properties in terms of conciseness, but it may require quadratic number of phantom items (Nisan, 2000).

Boutilier and Hoos (2001) proposed another family of bidding languages denoted by $\mathcal{L}_{GB}$. It allows for logical combination of both goods and bids and thus it inherits the advantages of both $\mathcal{L}_G$ and $\mathcal{L}_B$ families.

4. Bidding languages for continuous auctions

Bidding languages for combinatorial auctions are well established in the literature. Thus, it is natural to derive the languages for continuous auctions from combinatorial ones. We will formulate languages for continuous auctions in relation to the languages defined in the previous section.

A valuation function for a continuous auction is a function $v : \mathbb{R}^C \to \mathbb{R}$. It is defined over the allocations space, where an allocation is a vector of commodity levels, $X = (X_c) \in \mathbb{R}^C, c \in \mathcal{C}, X_c \in \mathbb{R}$.

In combinatorial auction a bidding language can be used to present one's valuation in an approximate (strategic) or accurate way. In the continuous case the accurate representation of the valuations would make the Winner Determination Problem too complex for efficient computation. In the rest of the paper we assume that the utility function, and so the valuations, are the Lipschitz functions.

We will focus only on bidding languages superfamily under the assumption that the valuations can be approximated by piecewise linear functions. As in combinatorial case in which one may enumerate all combinations and thus may present his utility accurately, also in continuous case an agent may achieve required error level of approximation with sufficiently large number of linear pieces. If the number of pieces goes to infinity, then the approximation error converges to zero.

Definition 3 *A bidding language is asymptotically fully expressive if the error of approximation tends to 0 when the number of offers used to represent the utility function goes to infinity.*

Asymptotically fully expressive bidding language can express any utility function using infinite (or less) number of offers.

Computational complexity of a given language is a complexity related to determining the valuation of a bid in the language for a given allocation. The complexity of bidding language is important, because to make a decision about the allocation in the Winner Determination Problem, the value of the bids must be computed.

Definition 4 *(Nisan, 2000) A bidding language is polynomially interpretable if there exists a polynomial time algorithm that for any bid in the language and given allocation X it computes the value $v(X)$.*

With a polynomially interpretable bidding language there is a hope to achieve efficient algorithm for the WDP. However, in the field of combinatorial auctions the polynomially interpretable languages are not expressive enough and instead that, it is desired that having a proof (argument of the valuation function) it is possible to verify its optimality in polynomial time (Nisan, 2000).

Now, we will reformulate the notions of *complementarity* and *substitutability* for the continuous auction case. The underlying ideas remain unchanged, only formal definitions need to be rewritten. Let $x^i = [0, \ldots, 0, x_i, 0, \ldots, 0]$ denote a vector of commodities in which only i-th element is not equal to 0 and is equal to x_i. Then complementarity is defined by the following condition:

$$v(x^a) + v(x^b) \leq v(x^a + x^b). \tag{7}$$

Complementarity is perfect if $v(x^a) = v(x^b) = 0$.

Substitutability is defined as follows:

$$v(x^a) + v(x^b) \geq v(x^a + x^b) \tag{8}$$

and in case of perfect substitution $\geq$ is replaced with equality.

Offers on combinatorial auctions can be accepted or rejected. In case of continuous auctions the offers are accepted at a certain level. Let us consider an agent that would like to obtain the bundle of two units of a and three units of b and is going to pay 6 monetary units for that bundle. Assuming that the mechanism allocates to him one item a and one item b, what is the level of acceptance? Obviously, the demand of this agent is not fully covered. Half of demand for a is covered, but only $\frac{1}{3}$ of demand for b is satisfied. Thus, only $\frac{1}{3}$ of the whole bundle is covered and the payment should be $\frac{1}{3} * 6 = 2$. We can say that the accepted volume of the bundle is $\frac{1}{3}$, and 6 is the unit price of the so called *normalized volume* of the bundle.

Definition 5 *(Normalizing function) Assume that an agent specifies the bundle of commodities that he would like to obtain, $y \in \mathbb{R}^C$, where y_c is desired volume of commodity c. Let p be a price for the bundle y given by the agent. Normalizing function of a given bid is a function $f : \mathbb{R}^C \to \mathbb{R}$, that for a given commodity allocation gives the level of bid acceptance and satisfies the following conditions*

$$f(0) = 0 \tag{9}$$

$$f(y) = p \tag{10}$$

$$f(\epsilon y) = \epsilon p, \forall \epsilon \in \mathbb{R}^+. \tag{11}$$

Normalization is an immanent part of a bid which must be provided explicitly or implicitly. For instance, in a bid for a single item c, the normalizing function is obvious: $f(x) = x_c$, and price in the offer is price for a unit of c. If normalizing function is not obvious, then it must be provided by an agent in a bid.

4.1. Family of good-based languages

Analogously to the family $\mathcal{L}_G$ we introduce $\mathcal{L}_{CG}$, a family of languages for continuous auctions. Instead of logical formulas on commodities, the domain $\mathcal{D} \subseteq \mathbb{R}^C$ of feasible allocations is provided in a bid in any language that belongs to the family $\mathcal{L}_{CG}$.

Definition 6 *An offer in family $\mathcal{L}_{CG}$ is a tuple $(f, p, \mathcal{D})$, where*

- *$f : \mathbb{R}^C \to \mathbb{R}$ is a function to compute the normalized, unit volume,*
- *p is a price for a unit of normalized volume defined by function f,*
- *$\mathcal{D} \subseteq \mathbb{R}^C$ is domain of feasible commodity allocation to the offer.*

Then, the valuation is defined as follows:

$$v(X = (X_1, \dots, X_C)) = p\bar{x} \tag{12}$$

$$\bar{x} = \begin{cases} \max_{x \in \mathcal{D}, 0 \le x_c \le X_c \forall c: X_c \ge 0, X_c \le x_c \le 0 \ \forall c: X_c < 0} f(x) & \textit{if such } x \textit{ exists} \\ 0 & \textit{otherwise} \end{cases} \tag{13}$$

Various languages in the family $\mathcal{L}_{CG}$ differ in the way of defining $\mathcal{D}$ in a bid. Let us introduce a language $\mathcal{L}_{CG}^{simplex}$ from the family $\mathcal{L}_{CG}$ with restriction that $\mathcal{D}$ is defined by a simplex and $f(x)$ is linear. In $\mathcal{L}_{CG}^{simplex}$ a definition of $f(x)$ can be replaced with a vector $\alpha = (\alpha_1, \dots, \alpha_C) \in \mathbb{R}^C$ of commodity shares. Then, the normalization function is defined as $f(x = (x_1, \dots, x_C)) = \sum_c \alpha_c x_c$.

$\mathcal{L}_{CG}^{simplex}$ is polynomially interpretable since the valuation can be formulated as a linear programme:

$$\max_x p \sum_{c \in \mathcal{C}} \alpha_c x_c \tag{14}$$

subject to

$$x_c \le \alpha_c X_c \tag{15}$$

$$x_c \ge 0 \tag{16}$$

$$x_c \in \mathcal{D} \tag{17}$$

where $\mathcal{D}$ is a simplex.

$\mathcal{L}_{CG}^{simplex}$ allows to express substitutability. Assume that c_1 and c_2 are perfectly substitutable. An agent needs one unit of these goods and is willing to pay p. He can formulate the following offer: $\langle \alpha = (1,1), p, \mathcal{D} = \{x : x_{c_1} + x_{c_2} \le 1\} \rangle$. The normalizing function is $f(x) = x_{c_1} + x_{c_2}$. The value of an allocation is defined by $p(x_{c_1} + x_{c_2})$. It is easy to see that like $\mathcal{L}_G^{pos}$, $\mathcal{L}_{CG}^{simplex}$ is not sufficient to express imperfect substitutability.

Let us consider the following example: an agent would like to receive either c_1 or c_2 with total maximal volume equal to 4. For each unit of c_1 he is willing to pay 10, and for each unit of c_2 he is willing to pay 11. Notice, that this case cannot be represented in language $\mathcal{L}_{CG}$. But it can be represented in extended one, $\mathcal{L}_{CG}^{simplex*}$. Let $\mathcal{L}_{CG}^{simplex*}$ be the language $\mathcal{L}_{CG}^{simplex}$ with additional dummy commodities. These commodities are binary which means that each commodity can be accepted fully or not at all. Then, they can be used to model disjunctions like in the language $\mathcal{L}_G^{OR*}$ (Nisan, 2000). In the above case, an agent must introduce dummy commodity c_3. Then the offers may look like these: $\langle((1,0,1),10,\{x : x_{c_3} \geq x_{c_1}/M\})\rangle$, $\langle((0,1,1),1,\{x : x_{c_3} \geq x_{c_2}/M\})\rangle$, where M is a number big enough.

As in $\mathcal{L}_G^{pos}$, perfect complementarity can be modeled in $\mathcal{L}_{CG}^{simplex}$, but imperfect complementarity not. If c_1 and c_2 are perfectly complementary, then α should be $(1,0)$ or $(0,1)$ and $\mathcal{D}$ should be defined by equation $x_{c_1} = x_{c_2}$.

$\mathcal{L}_{CG}^{simplex}$ is not fully expressive, but in case of perfect substitutability or complementarity it can be a convenient tool.

4.2. Family of bid-based languages

Now, we will introduce the family $\mathcal{L}_{CB}$ of languages which, similarly to $\mathcal{L}_B$, is based on functions of atomic bids. There are two types of atomic bids: simple and bundle offers.

Definition 7 *(Simple offer in a continuous auction) A simple offer for a commodity $c \in \mathcal{C}$ is a pair $(p, \mathcal{D}_c)$, where p is an offer price, $\mathcal{D}_c \subseteq \mathbb{R}$ is feasible domain, that is, if the offer is winning, then the allocation X_c to this offer must satisfy the condition $X_c \in \mathcal{D}_c$.*

The valuation of simple offer is defined as follows:

$$v(X = (0, \ldots, X_c, \ldots, 0)) = \begin{cases} pX_c & \text{if } X_c \in \mathcal{D}_c \\ 0 & \text{otherwise} \end{cases} . \tag{18}$$

Definition 8 *(Bundle offer in a continuous auction) A bundle offer is a tuple $(\alpha, p, \mathcal{D})$, where*

- *$\alpha = (\alpha_1, \ldots, \alpha_C)$ is a vector of commodity shares, $\alpha_c \in \mathbb{R}$ is the share of commodity c in the bundle,*
- *p is an offer price of whole bundle,*
- *$\mathcal{D} \subseteq \mathbb{R}^C$ is feasible domain of commodities allocated to this offer.*

The valuation of the bundle offer is defined as follows:

$$v(X = (X_1, \ldots, X_C)) = p\bar{x} \tag{19}$$

where $\bar{x}$ is an accepted volume of the bundle α:

$$\bar{x} = \begin{cases} \arg\max_{x \in \mathcal{D}, 0 \leq x_c \leq X_c} \min_c \{\frac{x_c}{\alpha_c}\} & \text{if there exists such } x \\ 0 & \text{otherwise} \end{cases} . \tag{20}$$

Simple offer is a special case of the bundle offer, but sometimes it is convenient to refer to the simple offers, especially if they are the only type of atomic bids that is allowed on a given auction. Notice that in contrast to the combinatorial auctions, where limiting the atomic bids would make no sense, it could be quite natural in the simple version of continuous auction.

In $\mathcal{L}_{CB}^{or}$ language several atomic bids can be combined with the operator OR. A combination of two bids, b_1 and b_2, defines the following valuation:

$$(v_1 \ OR \ v_2)(X) = \max_{(x)_1,(x)_2} (v_1((x)_1) + v_2((x)_2)) \tag{21}$$

subject to constraints

$$(x)_1 + (x)_2 \le X, (x_i)_1, (x_i)_2 \ge 0 \tag{22}$$

where $(x)_1, (x)_2 \in \mathbb{R}^C$ and $(x_i)_n$ is i-th element of vector $(x)_n$.

Consider Example 1 assuming that fractional allocations are allowed. Then, complementarity can be addressed in a similar way as in combinatorial case. An agent needs to define two simple offers and one bundle offer with shares $\alpha = (1, 1)$. Then, the offers must be combined with the OR operator. Perfect substitutability can be addressed directly by combining two simple offers with OR. $\mathcal{L}_{CB}^{or}$ does not allow to model an imperfect substitutability, since two simple bids will be treated separately instead of the "worse" bundle.

In $\mathcal{L}_{CB}^{xor}$ language two (or more) atomic bids can be combined with the operator XOR, which defines the following valuation:

$$(v_1 \ XOR \ v_2)(X) = \max_{(x)_1,(x)_2} \{v_1((x)_1), v_2((x)_2)\} \tag{23}$$

subject to constraints

$$(x)_1 + (x)_2 \le X, (x_i)_1, (x_i)_2 \ge 0. \tag{24}$$

Both Examples 1 and 3 can be modeled in the same way like in combinatorial case – by combining with XOR two simple offers and one bundle bid.

Analogously to the other languages defined in Section 4, the languages $\mathcal{L}_{CB}^{or-of-xor}$, $\mathcal{L}_{CB}^{xor-of-or}$ and $\mathcal{L}_{CB}^{or/xor}$ can be also defined. Introduction of a language equivalent to OR* requires that the dummy commodities be binary – they can be accepted with volume 1 or not accepted and this integrality must be taken into account in equation (22).

4.3. Family of good- and bid-based languages

The last family of languages is based on the concept of Boutilier and Hoos, which is a kind of mixture of the previously defined families (Boutilier and Hoos, 2001). The language $\mathcal{L}_{CGB}$ is defined as follows:

- bundle offer is in $\mathcal{L}_{CGB}$,
- if $b_1, b_2 \in \mathcal{L}_{CGB}$ then $(b_1 \wedge b_2, p), (b_1 \vee b_2, p), (b_1 \oplus b_2, p)$ are all in $\mathcal{L}_{CGB}$.

Let $\Phi(b)$ be the formula associated with a bid b, taking one of the following shapes: $b, (b_1 \wedge b_2), (b_1 \vee b_2), (b_1 \oplus b_2)$. The function $\sigma(\Phi(b), X)$ gives the volume allocated to the bid b with formula Φ, when the allocation computed by WDP is X.

- If $\Phi(b)$ is a bundle offer, then $\sigma(\Phi(b), X) = \bar{x}$, $\bar{x}$ is defined as in (13);
- If $\Phi(b) = b_1 \vee b_2$ or $\Phi(b) = b_1 \oplus b_2$, then $\sigma(\Phi(b), X) = \max(\sigma(\Phi(b_1), X), \sigma(\Phi(b_2), X))$;
- If $\Phi(b) = b_1 \wedge b_2$, then $\sigma(\Phi(b), X) = \min(\sigma(\Phi(b_1), X), \sigma(\Phi(b_2), X))$.

The valuation is defined as follows:

- If the bid is a bundle offer, then the valuation is equal to the one of bundle offer;
- If $\Phi(b) = b_1 \vee b_2$ then the valuation is the sum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \vee \Phi(b_2), X)$;
- If $\Phi(b) = b_1 \wedge b_2$ then the valuation is the sum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \wedge \Phi(b_2), X)$;
- If $\Phi(b) = b_1 \oplus b_2$ then the valuation is the sum of maximum of valuations for $\Phi(b_1)$ and $\Phi(b_2)$ and $p\sigma(\Phi(b_1) \vee \Phi(b_2), X)$.

Notice that in contrast to $\mathcal{L}_{CB}$ there are no constraints like (22) or (24). So, if the allocation is satisfying many formulas, then each formula is taken in the valuation. More justifications for the definition of the language can be derived from its combinatorial version presented in Boutilier and Hoos (2001).

Let us consider Example 1 under the assumption that fractional allocation is feasible. Two bundle offers should be stated: $v_1 = \langle (1,0), 2, D = \{x : x_1 \leq 1, x_2 = 0\}\rangle$ and $v_2 = \langle (0,1), 3, D = \{x : x_1 = 0, x_1 \leq 1\}\rangle$. Then the offer $\langle v_1 \wedge v_2, 1\rangle$ represents the correct valuation. Assume that the allocation is $X = (0.5, 1)$. Then, $\sigma(v_1, X, X) = 0.5$, $\sigma(v_1) = 1$, and $\sigma(v_1 \wedge v_2, X) = \min\{0.5, 1\} = 0.5$. The valuation is the sum of valuations of v_1 and v_2 and $p\sigma(v_1 \wedge v_2, X)$, that is 0.5*2+1*3+0.5*1=4.5. The value can be calculated in a different way. Both commodities are worth 6 for a unit of the bundle, but only 0.5 unit of both of them are available. The rest volume of c_2, which is 0.5, is worth 3 for a unit. Then, in fact, the value is 6*0.5+0.5*3=4.5. Notice that in this language the intrinsic values are visible in simple offers, and in complex offers an increase of the value due to commodity combination is clearly visible.

5. Language properties

Languages $\mathcal{L}_{CG}^{simplex}$ and $\mathcal{L}_{CB}^{or}$ are not asymptotically fully expressive. In previous sections we have presented valuations that cannot be represented in these languages. Languages $\mathcal{L}_{CG}^{simplex*}$, $\mathcal{L}_{CB}^{XOR}$, $\mathcal{L}_{CGB}$ are asymptotically fully expressive, since they can model exclusive disjunction†. In that case, a space of allocations can be divided into any number of intervals. For each interval an atomic bid can be defined and each combination of atomic bids can be combined with XORs (in $\mathcal{L}_{CG}^{simplex}$ XOR is modeled by dummy good and single commodity in a logical expression corresponds to an atomic bid). Also other languages in $\mathcal{L}_{CB}$ which have XOR possibilities are asymptotically fully expressive.

†Notice that the free disposal assumption is crucial for asymptotically full expressiveness. Without free disposal assumption only $\mathcal{L}_{CG}^{simplex}$ is asymptotically fully expressive.

Let us define the size of a bid as a number of atomic formulas (e.g. atomic bids, bundle bids, elements of logical expression) contained in the bid. We consider a language to be better than other if it can express any valuation more compactly. This means that it needs less bids with smaller sizes. Exploiting the property of asymptotically full expressiveness may involve exponential number of bids or formulas in a bid. There is no proven dominating language in the meaning of succinctness.

Another aspect is the computational complexity of a language. $\mathcal{L}_{CG}^{simplex}$ is proven to be polynomially interpretable. The open question is if there is a sharp edge between asymptotically full expressiveness and polynomial complexity.

Perhaps the most important feature of a bidding language is the complexity for the bidding agent. The languages, that enable to model the utility in a natural and simple way, are preferred. Again, there is no language which could be acknowledged as the best according to this criterion. $\mathcal{L}_{CG}^{simplex}$ and $\mathcal{L}_{CB}^{or}$ naturally cover a case of additive valuations of goods. However, $\mathcal{L}_{CGB}$ appears to be the most intuitive in some non-additive valuations. Let us consider two following examples.

Example 4 *Suppose that an agent needs two complementary goods c_1 and c_2 with a joint value 10. The valuations for individual commodities are 1 and 2 for commodity c_1 and c_2, respectively. The maximal requested volume is 4. The required valuation is $10 * \min\{x_1, x_2\} + x_1 + 2x_2$, assuming that $x_1, x_2 \leq 4$.*

In $\mathcal{L}_{CG}^{simplex}$ the above valuation can be represented by the following bids:

$$(f(x_1, x_2) = x_1, 10, \mathcal{D} = \{(x_1, x_2) : x_1 = x_2, 0 \leq x_1, x_2 \leq 4) \tag{25}$$

$$(f(x_1, x_2) = x_1, 1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4) \tag{26}$$

$$(f(x_1, x_2) = x_2, 2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_2 \leq 4). \tag{27}$$

If a mechanism is preferring the most expensive bids, then these bids would represent the utility correctly. But, if a mechanism will choose the cheapest at first, then any of the bids for single commodities can be accepted instead of the first offer.

In $\mathcal{L}_{CB}$ language the utility can be defined as a set of atomic bids: $(13, \mathcal{D} = \{(x_1, x_2) : x_1 = x_2, 0 \leq x_1, x_2 \leq 4)\})$, $(1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4)\})$, $(2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4)\})$, which are combined with operator OR. Then, the task to maximize the value for the bidder is not within the mechanism, but is satisfied by definition (22).

In $\mathcal{L}_{CGB}$ the utility can be represented in an intuitive way, reflecting the structure of the utility function: $\langle((1,0), 1, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_1 \leq 4)\}) \wedge ((0,1), 2, \mathcal{D} = \{(x_1, x_2) : 0 \leq x_2 \leq 4)\}), 10\rangle$. Notice that the numbers appearing in the definition of the valuation, are directly included in the bid. Price 10 of the complex bid is interpreted as a basic value of possessing any of commodity, and prices in atomic bids play a role of additional profit for choosing particular commodity.

EXAMPLE 5 *Suppose that an agent needs two substitutable goods* c_1 *and* c_2*. Each of them provides the basic valuation 10, but also each of them provides some additional bonus: 1, 2 in case of commodity* c_1 *and* c_2 *respectively. The maximal requested volume is 4. The required valuation is* $10*(x_1+x_2)+x_1+2x_2$*, assuming that* $x_1 + x_2 \leq 4$.

In $\mathcal{L}_{CG}$ the most natural way is to use OR operator, e.g.: $\langle (c_1, 11)\ OR\ (c_2, 12)\rangle$. In $\mathcal{L}_{CB}$, again, it is impossible to model this utility. $\mathcal{L}_{CGB}$ seems to be the most intuitive since, as in previous example, it directly reflects the structure of the utility function: $\langle(((1,0),1,\mathcal{D} = \{(x_1,x_2) : 0 \leq x_1 \leq 4)\}) \vee ((0,1),2,\mathcal{D} = \{(x_1,x_2) : 0 \leq x_2 \leq 4)\}),10\rangle$.

6. Summary

We have introduced three families of bidding languages for continuous auctions. They are based on the concepts derived from the well-defined combinatorial auctions. We have generalized the language families based on goods, bids, and on both of them to the continuous case. We have also generalized several notions, which create a solid ground for bidding languages in a continuous case. Some of the properties known from their equivalents in combinatorial auctions are preserved in the proposed languages. Most of introduced languages are asymptotically fully expressive, but they differ in succinctness and logical grounds from the agent point of view. No language is dominating in terms of these criteria.

Further work should include deeper analysis of the languages in the context of their expressiveness and succinctness for particular classes of valuation functions.

Acknowledgments

The research was supported by the Polish National Budget Funds 2010-2013 for science under the grant N N514 044438.

References

BENYOUCEF, M. and PRINGADI, R. (2006) *A BPEL Based Implementation of Online Auctions. Lecture Notes in Computer Science* **4652**, Springer.

BOUTILIER, C. and HOOS, H.H. (2001) Bidding languages for combinatorial auctions. *Proc. 17th Intl. Joint Conference on Artificial Intelligence.* AAAI Press, 1211—1217.

FUJISHIMA, Y., LEYTON-BROWN, K. and SHOHAM, Y. (1999) Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. *Proc. 16th International Joint Conferences on Artificial Intelligence.* Morgan Kaufmann Publishers Inc., 548—553.

Hoos, H.H. and Boutilier, C. (2000) Solving Combinatorial Auctions using Stochastic Local Search. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. AAAI Press, 22–29.

Kaleta, M. (2012) Security constrained network winner determination problem. *Automatyzacja Procesów Dyskretnych, Teoria i zastosowania*. Silesian University of Technology.

Kaleta, M. and Traczyk, T. (eds.) (2012) *Modeling Multi-commodity Trade: Information Exchange Methods. Advances in Intelligent and Soft Computing* **121**. Springer.

Lehmann, D., Müller, R. and Sandholm, T. (2006) The winner determination problem. Chap. 12 in: P. Cramton, Y. Shoham and R. Steinberg, eds., *Combinatorial Auctions*. MIT Press.

Nisan, N. (2000) Bidding and allocation in combinatorial auctions. *Proceedings of ACM Conference on Electronic Commerce*. ACM Press New York, 1–12.

Rolli, D. and Eberhart, A. (2005) An Auction Reference Model for Describing and Running Auctions. *Proc. of the Wirtschaftsinformatik*. Physica –Verlag HD, 289—308.