Auctioning Substitutable Goods

Andrea Giovannucci [a], Juan A. Rodríguez-Aguilar [a], Jesús Cerquides [b]
[a] *Artificial Intelligence Research Institute, IIIA*
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain
{*andrea,jar*}*@iiia.csic.es*

[b] *Dept. de Matemàtica Aplicada i Anàlisi*
Universitat de Barcelona
Gran Via, 585, 08007, Barcelona, Spain
cerquide@maia.ub.es

Abstract. In this paper we extend the notion of multi-unit combinatorial reverse auction by adding a new dimension to the goods at auction. In such a new type of combinatorial auction a buyer can express substitutability relationships among goods: some goods can be substituted with others at a substitution cost. Substitutability relationships allow a buyer to introduce his uncertainty as to whether it is more convenient to buy some goods or others. We introduce such uncertainty in the winner determination problem (WDP) so that not only does the auction help allocate the optimal set of offers —taking into account substitutability relationships—, but also assesses the substitutability relationships that apply. In this way, the buyer finds out what goods to buy, to whom, and what *operations* (substitutions) to apply to the acquired goods in order to obtain the initially required ones.

Keywords. Combinatorial Auctions, Multi-Agent Systems

1. Introduction

Since many auctions involve the selling or buying of a variety of different assets, combinatorial auctions [1] (CA) have recently deserved much attention in the literature. In particular, a significant amount of work has been devoted to the problem of selecting the winning set of bids [5,6]. Nonetheless, while the literature has considered the possibility to express relationships among goods on the bidder side —such as complementarity and substitutability (e.g. [6])—, the impact of the eventual relationships among the different assets to sell/buy on the bid-taker side has not been conveniently addressed so far.

Consider that a company devoted to the assembly and repairing of personal computers (PCs) requires to assembly new PCs in order to fulfil his demand. Say that its warehouse contains most of the components composing each PC. However, there are no components to assemble motherboards (we consider that a motherboard is composed of 1 CPU, 4 RAM units, and 3 USB connectors). Therefore, the company would have to start a sourcing [2] process to acquire such components. For this purpose, it may opt for running a combinatorial reverse auction [6] with qualified providers. But before that, a professional buyer may realise that he faces a decision problem: shall he buy the required components to assemble them in house into motherboards, or buy already-assembled motherboards, or opt for a *mixed purchase* and buy some components to assemble them and some already-assembled motherboards? This concern is reasonable since the cost of components plus transformation (assembly) costs may eventually be higher than the cost of already-assembled motherboards. To tackle this issue, the buyer could think of running separate auctions for motherboards and their components, and after that decide whether to buy the whole or the parts. Notice though that besides impractical and costly (in general, the more transformation relationships among goods we consider, the larger number of auctions would be required) this method would be missing the opportunity represented by mixed purchases. Hence, the buyer requires a combinatorial reverse auction mechanism that provides: (a) a language to express required goods along with the relationships that hold among them; and (b) a winner determination solver that not only assesses what goods to buy and to whom, but also the transformations to apply to such goods in order to obtain the initially required ones.

In this paper we try to provide solutions to both issues. Firstly, notice that we can resort to a more general semantics when referring to relationships among goods: the semantics of *substitutability*. In the example above, if a buyer requires a motherboard, we can say that it can be *substituted* with 1 CPU, 4 RAM units, and 3 USB connectors at a certain *substitution* (transformation in our example) cost. Notice though that this notion of substitutability among goods is different from the classic notion of substitutability on the bidder side that we find in the CA literature [6]. Since commercial e-sourcing tools only allow buyers to express fixed number of units per required good as part of the so-called *Request for Quotation* (RFQ), we have extended this notion to allow for the definition of substitutability relationships among goods. Thus, we introduce a formal definition of a *Substitutability Network Structure* (SNS) that largely borrows from Place/Transition Nets [3], where transitions stand for substitution relationships and places stand for required goods. Secondly, we present the formalisation of the winner determination problem for multi-unit combinatorial reverse auctions with substitutability relationships among goods (MUCRASG) by applying the expressiveness power of multi-set theory. Additionally, we provide a mapping of our formal model to integer programming that takes into account substitutability relationships to asses the winning set of bids along with the substitutions to apply in order to obtain the buyer's initial requirements. Introducing these relationships allows a buyer to put together to compete bidders that otherwise would not be competing (e.g. CPU and memory manufactures compete with motherboard manufacturers).

In what follows we provide an extended version of the example introduced above to illustrate the type of substitutability relationships that we are interested in representing. Figure 1 graphically represents the way a PC is assembled. Each circle (corresponding to a PTN *place*) represents a good to negotiate upon. Assembly/splitting operations are represented as horizontal bars connecting goods, likewise *transitions* in a PTN. The assembling and splitting operations are labelled with an indexed capital T, and shall be referred to as *substitutability relationships*. In particular $T1$ and $T2$ represent the effects of splitting operations whereas $T3$ and $T4$ stand for assembling operations. The values in parentheses, labelling good transformations, stand for the cost of each transformation every time it is *fired* (carried out). The arcs connecting a set of goods $G1$ to a transformation $T1$ indicates that the goods in $G1$ are an *input* to transformation $T1$. The arcs connecting a transformation $T1$ to a set of goods $G2$ indicates that goods in $G2$ are an *output* from transformation $T1$. In the example in figure 1, the $T2$ transformation, representing the way a motherboard is taken into pieces, has a motherboard as *input good* and CPUs, RAM memories, USBs and empty motherboards as *output goods*. We call *input weights* the labels on the arcs connecting *input goods* to transitions, and *output weights* the labels on the arcs connecting *output goods* to transitions. They indicate the units required of each *input good* to perform a transformation and the units generated per *output good* respectively. For instance, the labels on the arcs connected to $T3$ in figure 1 indicate that 1 motherboard is composed of 1 CPU, 4 RAM units, 3 USBs and 1 empty motherboard at a cost of 8 EUR.

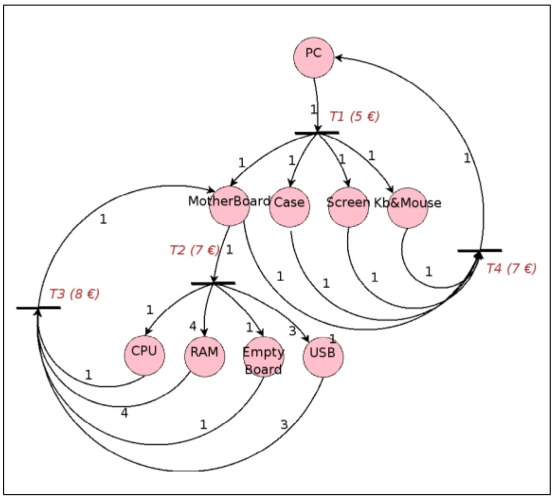

Figure 1. Graphical representation of an RFQ with substitutability relationships.

The paper is organised as follows. Section 2 provides some background knowledge on multisets and Place Transition Nets. Section 3 presents the formal model of multi-unit combinatorial reverse auctions with substitutability relationships among goods along with its winner determination problem and its mapping to integer programming. Finally, section 4 draws some conclusions and outlines directions for future research.

2. Background

A *multi-set* is an extension to the notion of set, considering the possibility of *multiple appearances* of the same element. A *multi-set* $\mathcal{M}_X$ over a set X is a function $\mathcal{M}_X : X \rightarrow \mathbb{N}$ mapping X to the cardinal numbers. For any $x \in X$, $\mathcal{M}_X(x) \in \mathbb{N}$ is called the *multiplicity* of x. An element $x \in X$ *belongs* to the multi-set $\mathcal{M}_X$ if $\mathcal{M}_X(x) \neq 0$ and we write $x \in \mathcal{M}_X$. We denote the set of multi-sets over X by X_{MS}.

In what follows we recall some definitions for *Place/Transition Nets* (PTN) [3]:

Definition 2.1. A *Place/Transition Net* (PTN) is a tuple $PTN = (G, T, A, E, \mathcal{M}_0)$ where: G is a set of *places*; T is a finite set of *transitions* such that $P \cap T = \emptyset$; $A \subseteq$

$(G \times T) \cup (T \times G)$ is a set of *arcs*; $E : A \rightarrow \mathbb{N}^+$ is an *arc expression* function; and $\mathcal{M}_0 \in G_{MS}$ is the *initial marking*.

Definition 2.2. A *Place Transition Net Structure* $N = (G, T, A, E)$ does not specify any initial marking. A Place Transition Net with a given initial marking $\mathcal{M}_0$ is denoted by $PTN = (N, \mathcal{M}_0)$.

The graphical representation of a PTN structure is composed of the following graphical elements: places are represented as circles, transitions are represented as bars, arcs connect places to transitions or transitions to places, and E labels arcs with values.

Definition 2.3. A *marking* is a multi-set over G. A *step* is a non-empty and finite multi-set over T. The *initial marking* $M_0 \in G_{MS}$ denotes the initial tokens distribution.

Definition 2.4. A step $\mathcal{S} \in T_{MS}$ is *enabled* in a marking $\mathcal{M} \in G_{MS}$ if the following property is satisfied: $\forall g \in G\ \sum_{t \in \mathcal{S}} E(g,t)\mathcal{S}(t) \leq \mathcal{M}(g)$.

Definition 2.5. Let the step $\mathcal{S}$ be enabled in a marking $\mathcal{M}_1$. Then, the step $\mathcal{S}$ may *occur*, changing the marking $\mathcal{M}_1$ to another marking $\mathcal{M}_2 \in G_{MS}$, defined as follows:

$$\forall g \in G : \mathcal{M}_2(g) = \mathcal{M}_1(g) + \textstyle\sum_{t \in \mathcal{S}} Z(g,t)\mathcal{S}(t)$$

where $Z(g,t) = E(g,t) - E(t,g)$. Moreover, we say that the $\mathcal{M}_2$ marking is *directly reachable* from the $\mathcal{M}_1$ marking by the occurrence of step $\mathcal{S}$, and we denote it by $\mathcal{M}_1[\mathcal{S} > \mathcal{M}_2$.

Definition 2.6. A *finite occurrence sequence* is a finite sequence of steps and markings $\mathcal{M}_1[\mathcal{S}_1 > \mathcal{M}_2 ... \mathcal{M}_n[\mathcal{S}_n > \mathcal{M}_{n+1}$ such that $n \in \mathbb{N}$ and $\mathcal{M}_i[\mathcal{S}_i > \mathcal{M}_{i+1}\ \forall i \in \{1, .., n\}$. $\mathcal{M}_1$ is called the *start marking*, while $\mathcal{M}_{n+1}$ is called the *end marking*.

We also define the *firing count vector* $\mathcal{K} \in T_{MS}$, associated to the finite occurrence sequence, as the union of all its steps: $\mathcal{K} = \bigcup_{i \in \{1,2,..,n\}} \mathcal{S}_i$

Definition 2.7. A marking $\mathcal{M}''$ is *reachable* from a marking $\mathcal{M}'$ iff there exists a finite occurrence sequence having $\mathcal{M}'$ as start marking and $\mathcal{M}''$ as end marking. In this case we say that $\mathcal{M}''$ is *reachable* from $\mathcal{M}'$ in n steps and we denote it as $\mathcal{M}'[\mathcal{K} > \mathcal{M}''$, where $\mathcal{K} = \bigcup_{i=1..n} \mathcal{S}_i$. Furthermore the start and end markings are related by the following equation:

$$\forall g \in G : \mathcal{M}''(g) = \mathcal{M}'(g) + \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t) \qquad (1)$$

The set of all possible markings reachable from a marking $\mathcal{M}$ is called its *reachability set*, and is denoted as $[\mathcal{M} >$.

Proposition 2.8. *In an acyclic Petri Net (with no directed circuits) a marking $\mathcal{M}''$ is reachable from a marking $\mathcal{M}'$ iff there exists a multi set $\mathcal{K} \in T_{MS}$ such that expression 1 holds ([4]).*

As a consequence, the reachability set $[M_0 >$ is represented by:

$$[M_0 >= \{\mathcal{M} \mid \exists \mathcal{K} \in T_{MS}\ s.t.\ \forall g \in G : \mathcal{M}(g) = \mathcal{M}_0(g) + \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t)\} \qquad (2)$$

3. Multi-Unit Combinatorial Reverse Auctions with Substitutability Relationships

3.1. Winner Determination Problem for MUCRASGs

A Substitutability Network Structure describes the different ways in which our business is allowed to transform goods and at which cost. More formally, we define it as follows:

Definition 3.1 (Substitutability network structure). A *Substitutability network structure* (SNS) is a pair $S = (N, C)$, where:

- N is a Place-Transition Net Structure $N = (G, T, A, E)$ such that: the *places* in G represent a set of goods to negotiate upon; the *transitions* in T represent a set of possible *substitutability relationships* among goods; the *directed arcs* in A connect goods to substitutability relationships; and the weights assigned by the *arc expression* function E indicate the number of units of each good that are either consumed or produced by a substitution. The values of E are the arc labels in figures 1 and 2.
- $C : T \to \mathbb{R}^+ \cup \{0\}$ is a cost function that associates a *substitution cost* to each *substitutability relationship*. The values of C are enclosed in parenthesis next to each transition in figures 1 and 2.

Notice that T represents the set of possible substitutions among subsets of G. The arcs in A relate either goods to substitutions or substitutions to goods. The goods connected to a substitutability relationship by incoming arcs (*input goods*) can substitute the goods connected to the very same substitutability relationship by outgoing arcs (*output goods*). The weights on the arcs connected to a substitutability relationship indicate the number of units of input and output goods consumed and produced respectively by the substitution.

Given a place transition net $PTN = (N, \mathcal{M}_0)$, if we consider $\mathcal{M}_0$ as a good configuration, PTN defines the space of good configurations *reachable* by means of applying substitutions to $\mathcal{M}_0$. The application of substitutions is obtained by firing transitions on the PTN. Henceforth, we define the concepts of *substitution step*, *enabling of a substitution step*, *occurrence of a substitution step* and *substitution sequence* as the equivalent to, respectively, *step*, *enabling of a step*, *occurrence of a step*, and *finite occurrence sequence* on a PTN.

We also need to define the concept of substitution cost, taking into account the cost of transforming good configuration $\mathcal{M}_0$ into another good configuration $\mathcal{M}_1 \in [M_0 >$ by means of some substitution sequence. The $\mathcal{K}$ multiset associated to substitution sequence $\mathcal{M}_0[\mathcal{K} > \mathcal{M}_1$ accounts for the number of times a transition is fired. Thus, the cost of transforming good configuration $\mathcal{M}_0$ into good configuration $\mathcal{M}_1$ amounts to adding the substitution cost of each transition in the $\mathcal{K}$ substitution sequence, namely:

Definition 3.2 (Substitution Cost). Given a firing count vector $\mathcal{K}$ associated to a substitution sequence, we define the substitution cost associated to it as:

$$C(\mathcal{K}) = \sum_{t \in \mathcal{K}} c(t)\mathcal{K}(t) \tag{3}$$

In the following example we formally specify the Substitutability Network Structure $S = (N, C)$ graphically represented in figure 2: $G = \{g_1, g_2, g_3, g_4\}$; $T = \{T_1\}$; $A = \{(g_1, T_1), (g_2, T_1), (T_1, g_3), (T_1, g_4)\}$; $E(g_1, t_1)$=3, $E(g_2, t_1)$=4, $E(t_1, g_3)$=2, $E(t_1, g_4)$=1; and $C(T_1) = 200$ EUR. It describes a buyer's capacity of transforming a pair of goods $(g1, g2)$ into a pair $(g3, g4)$ by means of substitution $t1$. The arc labels indicate that 3 units of good g_1 and 4 units of item g_2 can be transformed into (substituted for) 2 units of good g_3 and one unit of good g_4. C sets the substitution cost of T_1 to 200 EUR. Say that we assign an initial marking $\mathcal{M}_0$ to S: $\mathcal{M}_0(g_1) = 6$,

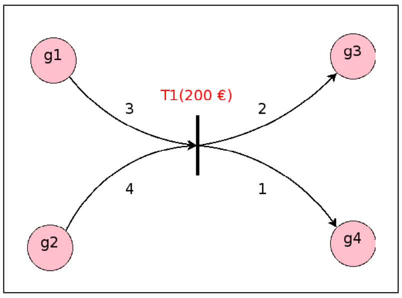

Figure 2. Graphical representation of a substitutability relationship

$\mathcal{M}_0(g_2) = 8$, $\mathcal{M}_0(g_3) = 0$, $\mathcal{M}_0(g_4) = 0$. The underlying PTN allows to transform it via substitutability relationship t_1 into a new good configuration $\mathcal{M}_1$: $\mathcal{M}_1(g_1) = 0$, $\mathcal{M}_1(g_2) = 0$, $\mathcal{M}_1(g_3) = 4$, $\mathcal{M}_1(g_4) = 2$. In such a case, the firing count vector would be $\mathcal{K} = \{t_1, t_1\}$ (t_1 is fired twice), and the substitution cost would amount to $C(\mathcal{K}) = 2 * 200EUR = 400EUR$.

In a classic multi-unit combinatorial reverse auction scenario, a Request For Quotation (RFQ), a buyer's requirement, can be expressed as a multiset $\mathcal{U} \in G_{MS}$ whose multiplicity indicates the number of units required per good. In the example of figure 2, if $\mathcal{U}(g_1) = 2, \mathcal{U}(g_2) = 2, \mathcal{U}(g_3) = 2, \mathcal{U}(g_4) = 1$, $\mathcal{U}$ would be representing a buyer's need for 2 units of g_1, g_2, and g_3, and 1 unit of g_4. Nonetheless, since substitutability relationships hold among goods, the buyer may have different alternatives depending on the bids he receives:

1. $\mathcal{M}_0 = [g_1\ g_1\ g_2\ g_2\ g_3\ g_3\ g_4]$. Buy all items as requested.
2. $\mathcal{M}_1 = [g_1\ g_1\ g_1\ g_1\ g_1\ g_2\ g_2\ g_2\ g_2\ g_2\ g_2]$. Buy 5 units of item g_1 and 6 units of item g_2 to transform 2 units and 4 units into 2 units of g_3 and 1 unit of g_4 at cost $c = 200$ EUR. The overall cost results from the cost of the acquired units plus transformation cost c. Thus, there is an extra cost.

Notice that both possibilities allow the buyer to obtain his initial requirement, namely 2 unit of g_1, 2 units of g_2, 2 units of g_3, and 1 unit of g_4, each one at a different cost. Notice also that a bid can be represented as a multiset $\mathcal{B} \in G_{MS}$, whose multiplicity indicates the number of units offered per good.

Definition 3.3 (Winner Determination Problem). Given a set of bids B, their costs $p : B \to \mathbb{R}^+ \cup \{0\}$, an RFQ $\mathcal{U}$, and a substitutability network structure $S = (N, C)$, the winner determination problem amounts to selecting a subset of bids ($W \subseteq B$) and to assessing a substitution sequence to apply to them in order to fulfil the requirements in $\mathcal{U}$ while minimising the total cost of the substitution sequence plus W.

We begin by defining the set of possible auction outcomes. A possible auction outcome is a pair $(W, \mathcal{K})$, where $W \subseteq B$ contains a set of bids, and $\mathcal{K}$ represents a substitution sequence. The application of $\mathcal{K}$ to $PTN = (N, \cup_{\mathcal{B} \in B} \mathcal{B})$ allows to obtain a good configuration that fulfils a buyer's requirement $\mathcal{U}$. More formally, the set of possible auction outcomes is defined as:

$$\Omega = \{(W, \mathcal{K}), W \subseteq B, \mathcal{K} \in T_{MS} \mid \exists \mathcal{X} \in G_{MS}\ s.t.\ (\bigcup_{\mathcal{B} \in W} \mathcal{B})[\mathcal{K} > \mathcal{X}, \mathcal{X} \supseteq \mathcal{U}\}. \quad (4)$$

For each outcome $(W, \mathcal{K})$, we associate an auction *outcome cost* $c(W, \mathcal{K}) = \sum_{\mathcal{B} \in W} p(\mathcal{B}) + C(\mathcal{K})$

Given a set of auction outcomes, the aim of the WDP for a MUCRASG is to find the optimal outcome $(W^{opt}, \mathcal{K}^{opt}) \in \Omega$ that minimises the outcome cost $c(W, \mathcal{K})$. Formally,

$$(W^{opt}, \mathcal{K}^{opt}) = \arg \min_{(W, \mathcal{K}) \in \Omega} c(W, \mathcal{K}) \quad (5)$$

3.2. Mapping to Integer Programming

We model the problem of assessing $(W^{opt}, \mathcal{M}^{opt})$ as an Integer Programming problem. For this purpose, we need to express as integer variables: (1) generic subset of bids ($W \subseteq B$); and a generic firing vector sequence ($\mathcal{K}$) associated to a substitution. In order to represent W we assign a binary decision variable $x_{\mathcal{B}}$ to each bid $\mathcal{B} \in B$, standing for $\mathcal{B}$ is being included ($x_{\mathcal{B}} = 1$) or not ($x_{\mathcal{B}} = 0$) in W. A multiset is uniquely determined by its mapping function $\mathcal{K} : T \rightarrow \mathbb{N}$. Hence, we represent multisets $\mathcal{K}\ T$ by considering an integer bounded decision variable q_t for each $t \in T$. Each q_t represents the multiplicity of element t in the $\mathcal{K}$ multi-set. Thus, the translation into integer programming of expression (5) becomes:

$$min \sum_{\mathcal{B} \in B} x_B p(\mathcal{B}) + \sum_{t \in T} q_t c(t)$$

subject to $x_{\mathcal{B}} \in \{0, 1\}, q_t \in \{0, 1, ...max_t\}$

Now we have to capture the side constraints enforcing that the selected bids, along with the transformations applied to them, fulfil $\mathcal{U}$, the initial buyer's requirement. For this purpose we translate expression 4 into linear programming. We consider a set of PTNs such that $PTN = (N, \mathcal{L})$, where $\mathcal{L} = \cup_{\mathcal{B} \in W} \mathcal{B}$.

Moreover. we must consider all the finite occurrence sequences of $PTN = (N, \mathcal{L})$ that transform $\mathcal{L}$ into a configuration that at least fulfils $\mathcal{U}$. Under the hypothesis of N being acyclic we explicit the reachability set of $\mathcal{L}$ as follows:

$$\forall g \in G: \ \mathcal{M}(g) = \mathcal{L}(g) + \sum_{t \in \mathcal{K}} Z(g, t) \mathcal{K}(t). \quad (6)$$

Next, we select the elements in the reachability set $[\mathcal{L} >$ that at least fulfil $\mathcal{U}$:

$$\forall g \in G: \ \mathcal{L}(g) + \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t) \geq \mathcal{U}(g) \quad (7)$$

Hence, expressing $\mathcal{L}$ as $\sum_{\mathcal{B} \in B} x_{\mathcal{B}}\mathcal{B}(g)$ we finally obtain the side constraints represented by expression 4 as:

$$\forall g \in G \ \ \sum_{\mathcal{B} \in B} x_{\mathcal{B}}\mathcal{B}(g) + \sum_{t \in T} Z(g,t)q_t \geq \mathcal{U}(g).$$

4. Conclusions

In this paper we have presented a formalisation and an integer programming solution to the winner determination problem of a new type of multi-unit combinatorial reverse auction that allows for expressing substitutability relationships on the buyer side. Several advantages derive from such a new type of auction. On the one hand, it allows a buyer to incorporate his uncertainty as to whether it is better to buy a required bundle of goods, or alternatively buy some goods to transform them into the former ones, or even opt for a mixed purchase and buy some goods as required and some others to be transformed. This is achieved by introducing substitutability relationships among goods into the winner determination problem. Therefore, not only does the winner determination solver assess what goods to buy and to whom, but also the transformations to apply to such goods in order to obtain the initially required ones. To the best of our knowledge, this is the first type of auction aimed at also handling buyers' uncertainty. As a side effect, the introduction of substitutability relationships is expected to increase competitiveness among bidders, and thus obtain better deals since bidders that otherwise would not be competing are put together to compete. Finally, our integer programming solution can be readily implemented with the aid of linear programming libraries.

Acknowledgements

This work has been supported by project Web-i(2) (TIC-2003-08763-C02-01).

Andrea Giovannucci enjoys the BEC.09.01.04/05-164 CSIC scholarship.

References

[1] Sven de Vries and Rakesh Vohra. Combinatorial auctions: A survey. *INFORMS Journal of Computing*, 15(3):284–309, 2003.

[2] Aberdeen Group. Making e-sourcing strategic: from tactical technology to core business strategy. Technical report, Aberdeen Group, 2002.

[3] Kurt Jensen. *Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use*, volume 1, chapter 2, pages 78–80. Springer, 1997.

[4] T. Murata. Petri nets: Properties, analysis and applications. In *IEEE*, volume 77, pages 541–580, 1989.

[5] Michael H. Rothkopf, Aleksandar Pekec, and Ronald M. Harstad. Computationally manageable combinational auctions. *Manage. Sci.*, 44(8):1131–1147, 1998.

[6] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.