

Winner Determination for Mixed Multi-unit Combinatorial Auctions via Petri Nets

Andrea Giovannucci
J. A. Rodriguez-Aguilar
IIIA-CSIC

Jesus Cerquides
Universitat de Barcelona

Ulle Endriss
University of Amsterdam

ABSTRACT

Mixed Multi-Unit Combinatorial Auctions (MMUCAs) allow agents to bid for bundles of goods to buy, goods to sell, and transformations of goods. In particular, MMUCAs offer a high potential to be employed for the automated assembly of supply chains of agents offering goods and services, and in general MMUCAs extend and generalise several types of combinatorial auctions. Here we provide a formalism, based on an extension of Petri Nets, with which MMUCAs, and therefore all auction types subsumed by MMUCAs—and in particular combinatorial auctions for supply chain formation (SCF)—, can be formally analysed. As a second direct benefit, consequence of the provided mapping to Petri Nets, we manage to dramatically reduce the number of decision variables involved in the optimisation problem posed by MMUCAs from quadratic to linear for a wide class of MMUCA Winner Determination Problems (WDPs). Hence, we also make headway in the practical application of MMUCAs, and in particular to SCF.

General Terms

Algorithms, Economics

Keywords

Petri nets, Supply Chain Formation, Combinatorial Auctions, Integer Programming

1. INTRODUCTION

A combinatorial auction (CA) is an auction where bidders can buy (or sell) entire bundles of goods in a single transaction [2]. Although computationally very complex, selling items in bundles has the great advantage of eliminating the risk for a bidder of not being able to obtain complementary items at a reasonable price in a follow-up auction (think of a combinatorial auction for a pair of shoes, as opposed to two consecutive single-item auctions for each of the individual shoes). The study of the mathematical, game-theoretical

and algorithmic properties of combinatorial auctions has recently become a popular research topic in AI. This is due not only to their relevance to important application areas such as electronic commerce or supply chain management, but also to the range of deep research questions raised by this auction model.

In particular, supply chain formation (SCF) appears as a very promising application area where strong complementarities arise. Indeed, Walsh et al. [6] observe that production technologies often have to deal with strong complementarities: the inputs and outputs of a production process are strongly connected since a producer may risk to produce unsold goods, as well as fail to produce already sold goods when failing to obtain the inputs, thus losing credibility on the market. Hence, a supply chain can be regarded as an intricate network of producers (entities transforming input goods into output goods at a certain cost), and consumers interacting in a complex way. Nevertheless, the complementarities arising in SCF are different from the ones we do find in CAs. The complementarities in SCF arise because of the preconditions and postconditions of production processes: precedences and dependences along the supply chain must be taken into account. Hence, whilst in CAs the complementarities can be simply represented as relationships among goods, in SCF the complementarities involve not only goods, but also interrelated *transformation* (production) relationships along several levels of the supply chain.

In order to automate SCF, Walsh et al. [6] have introduced the notion of task dependency network (TDN) as a way of introducing such complementarities. Nonetheless, although TDNs are indeed valuable to model SCF, it is our belief that further requirements (namely *expressiveness*, *computability*, and *formal analysis*) must be addressed to fully support automated SCF.

As to *expressiveness requirements*, we shall need: to represent complementarities among production processes (e.g. if iron and copper are melt together in the same oven, the transformation can be offered at a lower cost than the service of transforming iron and copper separately); to represent production relationships with multiple output goods (e.g. the quartering of a cow to sell its parts); to offer some bidding language to express combinations of bids; to consider the notion of free disposal (buy goods or transformations that remain unused); to support the specification of the configuration to end up with (a supply chain manager may be interested in finishing the SCF process with a given surplus of goods); to support a wide range of supply chain topologies

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

Copyright 2007 IFAAMAS .

beyond acyclic nets. As to *computational requirements*, we must ensure computational tractability of SCF while preserving optimality. Finally, as to *formal requirements*, we advocate for counting on a formalism that supports the formal study of structural and behavioural properties of a supply chain.

To achieve these goals, we depart from the work in [1], which introduces a generalisation of the standard model of combinatorial auction and discusses the issues of bidding and winner determination. The winner determination is the problem, faced by the auctioneer, of choosing which goods to award to which bidder so as to maximise its revenue. Bidding is the process of transmitting one’s valuation function over the set of goods on offer to the auctioneer (truthfully or otherwise). This new auction extends and generalises a whole range of combinatorial auctions: single-unit CAs, multi-unit CA, double CAs, and supply chain formation CAs. It provides a bidding language that can express several types of complex bids, and allows for bids on combinations of production processes, as well as a general WDP solver working on any network topology. This auction model is called mixed multi-unit combinatorial auction (MMUCA). Notice that this must not to be confused with a double auction [2]. In particular, the *order* in which agents consume and produce goods is of central importance in our model and affects the definition of the winner determination problem. Notice that the use of MMUCAs shall guarantee the satisfaction of the expressiveness requirements above.

In order to provide a formalism to reason about MMUCAs, and therefore also about SCF, we firstly define a new type of Petri Nets [4], the so-called *Weighted Transition Petri Nets* (WTPN), to express the notion of transformation (production) cost. Petri nets are a well known graphical tool to analyse discrete dynamical systems. We resort to Petri Nets because they can naturally help capture the notions of transformation; they have a well-defined semantics that can naturally accommodate the notion of sequence of transformations; they have an integrated description of both states and actions to characterise the search space where transformations occur; they have a large number of formal analysis methods that allow the investigation of structural and dynamic properties of the net; they have a graphical representation that is intuitively very appealing to study problems related to the topology of the supply chain. Secondly, we introduce and solve a new type of reachability problem over WTPN to which we map the MMUCA WDP. Two major benefits, and therefore contributions, stem from this process. First of all, as a main benefit, we do manage to provide a formalism with which MMUCAs, and therefore all auction types subsumed by MMUCAs —and in particular CAs for SCF—, can be formally analysed. For instance, topological problems of a supply chain can be readily analysed by means of adapting Petri Nets tools. As a second benefit, direct consequence of the provided mapping to WTPNs, we manage to dramatically reduce the number of decision variables in the optimisation problem posed by MMUCAs from quadratic to linear for a wide class of MMUCA WDPs. Hence, we make headway in the practical application of MMUCAs, and in particular to SCF as intended. Both the formalism and the computational reduction allow us to comply with the formal and computational requirements mentioned above.

The paper is organised as follows. In Section 2 we describe MMUCAs and recall the Integer Programming (IP) formula-

tion of their WDP [1]. In section 3 we introduce a new type of Petri Nets, the so-called Petri Nets with Weighted Transitions (WTPN). Furthermore, we define and solve a new reachability problem over WTPN, the Constrained Maximum Weight Occurrence Sequence Problem (CMWOSP). Next, in section 4 we show the equivalence between the MMUCAs WDP and the CMWOSP. Finally, section 5 concludes with a discussion of related work and an outlook on future work.

2. MMUCA

Let G be the finite set of all the types of goods. A *transformation* is a pair of multisets over G : $(\mathcal{I}, \mathcal{O}) \in \mathbb{N}^G \times \mathbb{N}^G$. An agent offering the transformation $(\mathcal{I}, \mathcal{O})$ declares that it can deliver \mathcal{O} *after* having received \mathcal{I} . In our setting, bidders can offer any number of such transformations, including several copies of the same transformation. That is, agents will be negotiating over *multisets of transformations* $\mathcal{D} \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)}$. For example, $\{(\{\}, \{a\}), (\{b\}, \{c\})\}$ means that the agent in question is able to deliver a (no input required) and that it is able to deliver c if provided with b . Note that this is not the same as $\{(\{b\}, \{a, c\})\}$. In the former case, if another agent is able to produce b if provided with a , we can get c from nothing; in the latter case this would not work.

In an MMUCA, agents negotiate over bundles of transformations. Hence, a *valuation* $v : \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)} \rightarrow \mathbb{R}$ is a (typically partial) mapping from multisets of transformations to the real numbers. Intuitively, $v(\mathcal{D}) = p$ means that the agent equipped with valuation v is willing to make a payment of p in return for being allocated all the transformations in \mathcal{D} (in case p is a negative number, this means that the agent will accept the deal if it *receives* an amount of $|p|$). For instance, $v(\{(\{oven, dough\}, \{oven, cake\})\}) = -20$ means that I can produce a cake for \$20 if given an oven and some dough, and that I will return the oven again afterwards

An *atomic bid* $b = (\{(\mathcal{I}^1, \mathcal{O}^1), \dots, (\mathcal{I}^n, \mathcal{O}^n)\}, p, l)$ specifies a finite multiset of finite transformations, a price p , and a bid owner identifier l . To make the semantics of such an atomic bid precise, we need to decide whether or not we want to make a *free disposal* assumption. We can distinguish two types of free disposal. As to free disposal *at the bidder’s side*, there are two possible free disposals sub-types: *good free disposal* and *transformation free disposal*. *Good free disposal* means that a bidder would always be prepared to accept more goods and give fewer goods away, without requiring a change in payment; whereas *transformation free disposal* means that it is allowed that some transformations are sold but not employed in the transformation process. As to free disposal *at the auctioneer’s side*, we only have *good free disposal*, meaning that the auctioneer may accept more and give away fewer goods. Both these free disposals affect the definition of what constitutes a valid solution to the winner determination problem.

A suitable *bidding language* should allow a bidder to encode choices between alternative bids and the like [5]. Informally, an OR-combination of several bids signifies that the bidder would be happy to accept any number of the sub-bids specified, if paid the sum of the associated prices. An XOR-combination of bids expresses that the bidder is prepared to accept at most one of them. More details about bidding languages for MMUCAs can be found in [1].

The *input* to the WDP consists of a complex bid expression for each bidder, a multiset \mathcal{U}_{in} of goods the auctioneer holds to begin with, and a multiset \mathcal{U}_{out} of goods the auctioneer expects to end up with.

In standard combinatorial auctions, a solution to the WDP is a set of atomic bids to accept. In our setting, however, the *order* in which the auctioneer “uses” the accepted transformations matters. For instance, if the auctioneer holds a to begin with, then checking whether accepting the two bids $Bid_1 = (\{a\}, \{b\}, 10)$ and $Bid_2 = (\{b\}, \{c\}, 20)$ is feasible involves realising that we have to use Bid_1 before Bid_2 . Thus, a *solution* to the WDP will be a *sequence of transformations*. A *valid solution* has to meet two conditions:

(1) *Bidder constraints*: The multiset of transformations in the sequence has to *respect the bids* submitted by the bidders. This is a standard requirement. For instance, if a bidder submits an XOR-combination of transformations, at most one of them may be accepted. With no transformation free disposal, if a bidder submits an offer over a bundle of *transformations*, all of them must be employed in the transformation sequence, whereas in the case of transformation free disposal any number of the transformations in the bundle can be included into the solution sequence, but the price to be paid is the total price of the bid.

(2) *Auctioneer constraints*: The sequence of transformations has to be *implementable*: (a) check that \mathcal{U}_{in} is a superset of the input set of the first transformation; (b) then update the set of goods held by the auctioneer after each transformation and check that it is a superset of the input set of the next transformation; (c) finally check that the set of items held by the auctioneer in the end is a superset (the same set in the case of no good free disposal) of \mathcal{U}_{out} . An *optimal solution* is a valid solution that maximises the sum of prices associated with the atomic bids selected.

For the formal definition of the WDP, we restrict ourselves to bids in the XOR-language, which is known to be fully expressive for MMUCAs [1]. However, the results can easily be extended to other bidding languages, in particular languages including an OR-operator. Let B be the set of all atomic bids. Recall that an atomic bid $b = (\mathcal{D}_b, p_b, l_b)$ consists of a multiset of transformations, a price, and a label indicating the owner of the bid, i.e. $\mathcal{D}_b \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)}$, $p_b \in \mathbb{R}$, and $l_b \in L$. L is the set of bidders. B_l is the set of all bids submitted by bidder $l \in L$.

For each bid b , let t_{bk} be a unique label for the k th transformation in \mathcal{D}_b (for some arbitrary but fixed ordering of \mathcal{D}_b). Let $(\mathcal{I}_{bk}, \mathcal{O}_{bk})$ be the actual transformation labelled by t_{bk} . Finally, let T be the set of all t_{bk} ; that is, $|T|$ is the overall number of transformations mentioned anywhere in the bids.

The auctioneer has to decide which transformations to accept and in which order to implement them. Thus, we define a decision variable $x_{bk}^m \in \{0, 1\}$, where x_{bk}^m takes on value 1 if the transformation t_{bk} is selected at the m th position of the solution sequence, and 0 otherwise. We also introduce several sets of auxiliary binary decision variables: x^m takes on value 1 iff any transition at all is selected at the m th position of the solution sequence; x_{bk} takes on value 1 iff transition t_{bk} is present anywhere in the sequence; and x_b takes on value 1 iff any of the transformations in the atomic bid b are selected. Let $(\mathcal{I}^m, \mathcal{O}^m)$ be the m th transformation in the solution sequence, i.e. the $t_{b,k}$ such that

$$x_{b,k}^m = 1.$$

Given this sequence, we can obtain the set of goods held by the auctioneer after each transformation. For instance, say that the auctioneer begins with $\mathcal{U}_{in} = \{a, a, d, d\}$. If we apply the first transformation $(\mathcal{I}^1, \mathcal{O}^1) = (\{a, a\}, \{c\})$ (from two units of a produce one unit of c), the auctioneer ends up with $\mathcal{M}^1 = \{c, d, d\}$. Formally, we can express this operation as an equation over multisets:

$$\mathcal{M}^1(g) = \mathcal{U}_{in}(g) + \mathcal{O}^1(g) - \mathcal{I}^1(g)$$

The application of the transformation is only possible because two units of good a are available. This condition maps to:

$$\mathcal{U}_{in}(g) \geq \mathcal{I}^1(g)$$

Let $\mathcal{M}^m \in \mathbb{N}^G$ be the goods held by the auctioneer after applying the m th transformation. We can generalise the two equations above as follows (let $\mathcal{M}^0 = \mathcal{U}_{in}$):

$$\mathcal{M}^m(g) = \mathcal{M}^{m-1}(g) + \mathcal{O}^m(g) - \mathcal{I}^m(g) \quad (1)$$

$$\mathcal{I}^m(g) \leq \mathcal{M}^{m-1}(g) \quad (2)$$

Given this, notice that not all the solution sequences will be valid. In what follows we express as linear equations all the constraints that a *valid solution sequence* must fulfil.

- (1) $x_b \geq x_{bk} \ (\forall b \in B, k = 1, 2, \dots, |D_b|)$. Selecting a transformation in a bid b implies selecting bid b . Recall that $x_b = 1$ means that bid b has been selected. In case of no transformation free disposal, simply substitute \geq with $=$.
- (2) $\sum_{b \in B_l} x_b \leq 1 \ (\forall l \in L)$. The atomic bids submitted by each bidder are mutually exclusive (XOR).
- (3) $x_{bk} = \sum_{m=1}^{|T|} x_{bk}^m \ (\forall b \in B, k = 1, 2, \dots, |D_b|)$. A transformation can be selected at most for a single position in the solution sequence.
- (4) $x^m = \sum_{b \in B} \sum_{k=1}^{|D_b|} x_{bk}^m \ (m = 1, 2, \dots, |T|)$. At most one transformation is selected at each position of the solution sequence.
- (5) $x^m \geq x^{m+1} \ (m = 1, 2, \dots, |T|)$. There should be no gaps in the sequence.
- (6) We capture the condition expressed by equations (1) and (2). These must hold for each transformation $(\mathcal{I}^m, \mathcal{O}^m)$ in the solution sequence and each good $g \in G$. This condition ensures that all transformations have enough input goods available.

Firstly, the multiset of goods held by the auctioneer after performing m steps of the transformation sequence can be computed recursively, by equation ((1)), as follows:

$$\mathcal{M}^m(g) = \mathcal{U}_{in}(g) + \sum_{\ell=1}^m \sum_{b \in B} \sum_{k=1}^{|D_b|} x_{bk}^\ell \cdot (\mathcal{O}_{bk}(g) - \mathcal{I}_{bk}(g)) \quad (\forall g \in G, m = 1, 2, \dots, |T|) \quad (3)$$

Second, we formulate the constraint enforcing that enough goods must be available at step m to perform the next transformation (cf. equation ((2))):

$$\sum_{b \in B} \sum_{k=1}^{|D_b|} x_{bk}^m \cdot \mathcal{I}_{bk}(g) \leq \mathcal{M}^{m-1}(g) \quad (\forall g \in G, \forall m) \quad (4)$$

- (7) $\mathcal{M}^{|T|}(g) \geq \mathcal{U}_{out}(g) \ (\forall g \in G)$. After having performed all the selected transformations, the set of goods held by the auctioneer must be a superset of the final goods. This works correctly, because $\mathcal{M}^{|T|}(g) = \mathcal{M}^m(g)$ for the highest m with $x^m = 1$. In case of no goods free disposal simply substitute \geq by $=$.

Therefore, solving the WDP for MMUCAs with XOR-bids amounts to maximising $\sum_{b \in B} x_b \cdot p_b$, while fulfilling constraints (1)–(7).

Finally, a valid solution Σ (an optimal solution sequence) is obtained from the solution of the IP by making transition t_{bk} the m th element of the solution sequence iff $x_{bk}^m = 1$.

The number of decision variables in the above integer program is of the order of $|T|^2$ (corresponding to x_{bk}^m). This represents a serious computational cost. Thus, in what follows, we try to reduce the number of variables required to solve the problem at the price of reducing the class of solvable problems.

3. PETRI NETS WITH WEIGHTED TRANSITIONS (WTPN)

In this section we introduce and extend the concept of Place Transition Net (PTN). Based on such an extension, we provide the definition of a new optimisation problem on Petri nets.

3.1 Background on Petri Nets

Place Transition Nets, also known as Petri Nets [4], are a powerful mathematical and graphical tool for the description of discrete distributed systems. An example of Petri net is showed in figure 1. A PTN is a bipartite graph: it has *place* nodes, *transition* nodes, and directed arcs connecting places to transitions and transitions to places. The places connected to a transition by means of input arcs are called the *input places* of the transition, and the ones connected by outgoing arcs from the transition are the *output places* of the transition. Places can contain tokens. A distribution of tokens over the set of places is called a *marking*, and it stands for the state of the Petri net. We say that a transition is enabled in a marking if all its input places contain tokens in that marking. If the transition is enabled it can *fire* consuming tokens of the input places and producing tokens in the output places.

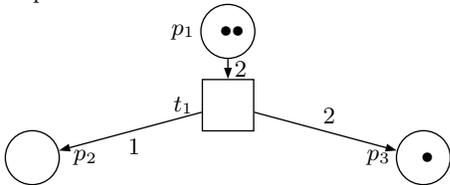


Figure 1: Example of a Place Transition Net

More formally, following [4], a *Place/Transition Net Structure* (PTNS) is a tuple $N = (P, T, A, E)$ such that: (1) P is a set of *places*; (2) T is a finite set of *transitions* such that $P \cap T = \{\}$; (3) $A \subseteq (P \times T) \cup (T \times P)$ is a set of *arcs*; (4) $E : A \rightarrow \mathbb{N}^+$ is an *arc expression* function (it represents the weights associated to the arcs, it stands for the number of input/output tokens consumed/produced by the transition). A *marking* $\mathcal{M} : P \rightarrow \mathbb{N}$ of a PTNS is a distribution of tokens over P . $\mathcal{M}(p) = k$ means that in marking \mathcal{M} place

$p \in P$ contains k tokens. The marking of picture 1 is represented by $\mathcal{M}(p_1) = 2, \mathcal{M}(p_2) = 0, \mathcal{M}(p_3) = 1$. A PTNS with a given initial marking \mathcal{M}_0 is called a *Place/Transition Net* (PTN). The graphical representation of a PTNS is composed of the following graphical elements: places are represented as circles, transitions are represented as rectangles, arcs connect places to transitions or transitions to places, and E labels arcs with values.

A transition $t \in T$ is said to be *enabled* if each input place p of t is marked with at least $E(p, t)$ tokens. $E(p, t)$ represents the weight of the arc connecting p to t . More formally, a transition p is enabled in a marking \mathcal{M} iff:

$$E(p, t) \leq \mathcal{M}(p) \quad \forall p \in P \quad (5)$$

The equation above enforces that enough tokens are present in every input place of t . For instance in figure 1 transition t_1 is enabled since $E(p_1, t_1) = 2 \leq \mathcal{M}(p_1) = 2$.

An enabled transition may or may not fire. A *firing* of an enabled transition removes $E(p, t)$ tokens from each input place p and adds $E(t, p)$ tokens to each output place p . A firing of a transition t will change marking \mathcal{M}_{k-1} to a marking \mathcal{M}_k . The new marking can be computed employing the following equation:

$$\mathcal{M}_k(p) = \mathcal{M}_{k-1}(p) + Z(t, p) \quad \forall p \in P \quad (6)$$

where $Z(t, p) = E(t, p) - E(p, t)$.

A sequence of firings will result in a series of markings. A marking \mathcal{M}_n is said to be reachable from a marking \mathcal{M}_0 if there exists a sequence of firings transforming \mathcal{M}_0 into \mathcal{M}_n . All the markings reachable from \mathcal{M}_0 in a PTN S are noted as $R(S, \mathcal{M}_0)$. A *finite firing sequence* is a finite sequence of transitions and markings: $J = \mathcal{M}_0 t_1 \mathcal{M}_1 \dots \mathcal{M}_{n-1} t_n \mathcal{M}_n$ or simply $J = t_1 t_2 \dots t_n$. In this case \mathcal{M}_n is reachable from \mathcal{M}_0 , and we write $\mathcal{M}_0[J > \mathcal{M}_n$. \mathcal{M}_0 is called the *start marking*, while \mathcal{M}_n is called the *end marking*.

Furthermore, the start and end markings are related by the following equation:

$$\forall p \in P \quad \mathcal{M}_n(p) = \mathcal{M}_0(p) + \sum_{t \in J} Z(t, p). \quad (7)$$

For a Petri Net N with r transitions and n places, the *incidence matrix* $A = [a_{i,j}]$ is an $r \times n$ matrix of integers and its typical entry is given by $a_{i,j} = a_{i,j}^+ - a_{i,j}^-$, where $a_{i,j}^+ = E(t_i, p_j)$ its the weight of the arc connecting the t_i transition to its output place p_j place, whereas $a_{i,j}^- = E(p_j, t_i)$. Notice that in this new representation a transition t_i is enabled in a marking if $a_{i,j}^- \leq \mathcal{M}(p_j)$. We can represent a marking \mathcal{M}_k as an $n \times 1$ column vector such that the j -th entry of \mathcal{M}_k represents the number of tokens present in place p_j after the k -th firing in some firing sequence. We also define the *firing vector* u_k as a vector of $r-1$ zeros and a 1 in the i -th position indicating that in marking \mathcal{M}_k transition t_i is fired. We can now express equation (6) in matrix form:

$$\mathcal{M}_k = \mathcal{M}_{k-1} + A^T u_k \quad k = 1, 2, \dots \quad (8)$$

Say that \mathcal{M}_d is reachable from \mathcal{M}_0 , then there exists a firing sequence $\{u_1, u_2, \dots, u_d\}$ bringing from \mathcal{M}_0 to \mathcal{M}_d . Therefore, a *necessary condition on reachability* can be expressed

¹Notice that for economy of notation we implicitly assume that $E(p, t) = 0$ if $(p, t) \notin A$ and $E(t, p) = 0$ if $(t, p) \notin A$.

in terms of a matrix equation:

$$\mathcal{M}_0[J > \mathcal{M}_d \Rightarrow (M_d = M_0 + A^T \mathbf{x}) \quad (9)$$

where $\mathbf{x} = \sum_{k=1}^d u_k$ is an $r \times 1$ column vector of nonnegative integers and is called the *firing count vector*. The i -th entry $\mathbf{x}[i]$ of the vector encodes the number of times a transition t_i must be fired to transform \mathcal{M}_0 into \mathcal{M}_d .

In [4], Murata shows that in an *acyclic* Petri Net (a net in which no directed cycles exist), the previous condition is not only necessary, but also sufficient. That is, if there exists a solution to equation (9), an occurrence sequence reaching \mathcal{M}_d from \mathcal{M}_0 is guaranteed to exist, and \mathbf{x} represents its firing count vector. In fact, Murata further extends the class of petri nets for which the condition is still sufficient. These particular nets (trap-circuit, siphon-circuit, TCC, and SCC nets) have special topologies with particular types of circuits. For such nets, the state equation represents all the reachable states if the initial marking \mathcal{M}_0 satisfies some constraints. Other efforts have been done for extending the validity of the state equation to more classes of petri nets. Due to lack of space, we cannot report all these results. Here we just remark that a whole corpus of tools for analysing the petri nets reachability problem can be directly imported to solve our problem.

3.2 Adding Weights to Petri Nets

We can finally extend the notion of Petri net by associating a *cost* to each transition to obtain a *Weighted Transition Petri Net* (WTPN). Thus, a WTPN is defined as a tuple $(P, T, A, E, \mathcal{M}_0, C)$ where:

- $P, T, A, E, \mathcal{M}_0$ are defined exactly as in a PTN.
- $C : T \rightarrow \mathbb{R}$ is a cost function that associates a cost to each transition.

A WTPN preserves all the properties of a PN but allows the quantitative representation of the cost of an operation/transition. With this tool, we can define the notion of cost C_{TS} of a sequence of transitions $J = \{u_1, u_2, \dots, u_d\}$ by adding the costs of all the transitions in the sequence:

$$C_{TS}(J) = \sum_{k=1}^d u_k \cdot c_T = c_T \cdot \sum_{k=1}^d u_k = \mathbf{x}^T \cdot c_T \quad (10)$$

where c_T is an $r \times 1$ cost vector, in which the i -th element is the cost associate to transition t_i . Therefore, the cost of each occurrence sequence is a linear function of its firing count vector.

3.3 Constrained Maximum Weight Occurrence Sequence Problem

Since there is a cost associated to each transition, one may be interested in finding a minimum/maximum cost firing sequence leading from a given initial marking to a given final marking. More importantly, one may be interested in finding a maximum/minimum cost firing sequence leading from a given initial marking \mathcal{M}_0 to a final marking \mathcal{M}_d that *fulfils a set of inequality constraints*. For instance, we may want to impose that in the final marking \mathcal{M}_d each place contains exactly one token, formally $\mathcal{M}_d(p) = 1, \forall p \in P$. With this aim we define the Constrained Maximum Weight Occurrence Sequence Problem (CMWOSP).

DEFINITION 1 (CMWOSP). *Given a WTPN $S = (P, T, A, E, \mathcal{M}_0, C)$, a set of inequality/equality constraints that a final marking \mathcal{M}_d must fulfil, expressed as:*

$$\forall p \in P \quad \mathcal{M}_d(p) \sim_p h_p \quad (11)$$

where $h_p \in \mathbb{N}$ and $\sim_p \in \{>, \geq, =\}^2$, find an occurrence sequence $J_{max} = \{u_1, u_2, \dots, u_d\}$ that brings the initial marking \mathcal{M}_0 to a final marking \mathcal{M}_d such that: (1) \mathcal{M}_d fulfils all the constraints in equation (11); and (2) J_{max} maximises the total cost C_{TS} .

LEMMA 1. *Consider an WTPN S and the following integer program:*

$$\max \quad \mathbf{x}^T c_T \quad (13)$$

$$\text{subject to} \quad A^T \mathbf{x} + M_0 \sim \mathbf{h} \quad (14)$$

All the integer solutions \mathbf{x}^{max} to this IP represent the firing count vectors of all the optimal solutions J_{max} to the CMWOSP defined by $\langle \sim, \mathbf{h} \rangle$ iff $A^T \mathbf{x} + M_0$ represents all the reachable states $R(S, \mathcal{M}_0)$.

PROOF. Notice that equation (14) simply imposes that the end marking fulfils the constraints defined by $\langle \sim, \mathbf{h} \rangle$. Now, the cost $C_{TS}(J)$ associated to the firing sequence represented by \mathbf{x} is equivalent to $\mathbf{x}^T c_T$ (see equation (10)). This is exactly the quantity being maximised in (13). As a result, a solution \mathbf{x}^* to the IP defined by equations (13) and (14) maximises the sum of the costs associated to fired transitions, while ensuring that the final marking is reachable and fulfils the constraints defined by $\langle \sim, \mathbf{h} \rangle$. \square

According to the results explained at the end of section 3.1, it is possible to express the reachability set with the state equation for a broad class of petri nets (e.g. acyclic, TCC, ..., etc.). In all these cases the integer program above is solution to the CMWOSP. Hence, we solve the CMWOSP problem in two steps. First, we determine the optimal firing count vector $\mathbf{x}_{J_{max}}$ by solving the Integer Linear Program (ILP) in equations (13) and (14). Then, we construct J_{max} from $\mathbf{x}_{J_{max}}$. Various techniques are available to solve this problem [7]. In what follows we explain how to build J_{max} for acyclic nets. Let $\mathbf{x}_{J_{max}}$ be the solution of the ILP. We can easily construct an occurrence sequence, with firing count vector $\mathbf{x}_{J_{max}}$, for which each step is enabled. Since S is acyclic, we can establish a partial order among transitions so that $t_1 < t_2$ iff t_2 uses as input some output of t_1 . We can construct an occurrence sequence J_{max} by ordering the transitions in the firing count vector $\mathbf{x}_{J_{max}}$ non-decreasingly according to our partial ordering. Every step in the so ordered occurrence sequence is guaranteed to be enabled. The occurrence sequence J_{max} is consequently the solution to our CMWOSP.

4. MAPPING MMUCA TO WTPN

In this section we demonstrate that an instance of the MMUCA WDP can be transformed into an instance of the CMWOSP problem. We introduce this mapping because it allows, for some classes of WTPN, to reduce the number of decision variables of the IP formulation.

²These inequations can be expressed in matrix form:

$$\mathcal{M}_d \sim \mathbf{h} \quad (12)$$

The idea behind the transformation of the WDP into a CMWOSP is that an atomic transformation can be viewed as a transition in a WTPN. Consider the example in figure 1, and say that each place represents a good. Let p_1 be water H_2O (each token represents a mole of water), p_2 be hydrogen H_2 , and p_3 be oxygen O_2 . Transition t_1 represents an hydrolysis process: 2 moles of water are transformed into 1 mole of oxygen and two moles of hydrogen at the cost associated to t_1 . Thus, a transition can graphically express an atomic transformation in which the input places are the input goods of the transformation, the output places are the output goods of the transformation, and the transition cost is the cost associated to the transformation. Equivalently, a bid offering goods can be represented as a transition with only output places, and a bid asking for goods is equivalent to a transition with only input places.

Consider the example of figure 2, it represents all the bids submitted within a MMUCA. Bid bid_1 offers two moles of water at the price of \$10, bid bid_2 the same at a price of \$14, bid bid_3 is a bid on the hidrolisis process for \$8. Bids bid_4 and bid_5 represent the offers to buy the products of the reaction for \$23 and \$25 respectively (the minus represents the fact that bidders pay money). In this example the optimal solution is straightforward: first buy two moles of water offered in bid_1 , then process the water through the transformation offered in bid_3 , and then sell the products of the reaction to bid bid_5 . The total gain of the supply chain is $25 - (8 + 10) = \$7$. Notice carefully that this solution is exactly the solution of the CMWOSP defined on the WTPN of figure 2 with initial marking empty and destination marking \mathcal{M}_d satisfying the following constraints: $\mathcal{M}_d(p_{H_2O}) \geq 0, \mathcal{M}_d(p_{O_2}) \geq 0$, and $\mathcal{M}_d(p_{H_2}) \geq 0^3$.

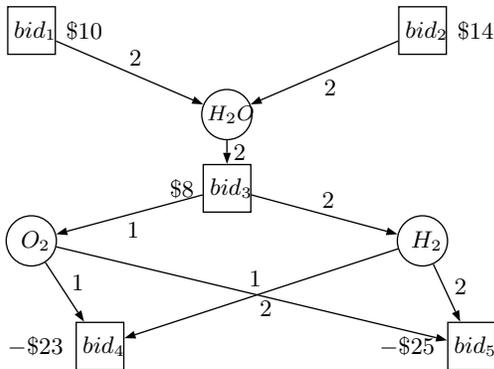


Figure 2: Example of a MMUCA in form of WTPN.

Given the example above we want to show that if we build a Petri net joining all the atomic transformations received within bids, we set the initial marking to the goods initially available to the auctioneer, and we set some constraints on the destination marking, the solution to the CMWOSP corresponds to the solution of the MMUCA WDP.

Informally this is the kind of mapping we intend to demonstrate. We obtain several advantages from this mapping. We can straightforwardly import a series of results and tools valid for Petri nets, as for instance tools to analyse static and behavioural properties of the net, tools to study the reachability problem, etc. As a major benefit, in this paper we manage to dramatically reduce the complexity of the

³In case of no free disposal replace \geq with $=$.

WDP. We succeed in composing an IP solver with a number of variables linear in the number of total atomic transformations. Recall that in the general case (see section 2) a quadratic number of variable was required.

In the previous example we restricted ourselves to the case in which providers can only submit one bids. We also considered only bids over a single atomic transformation, i.e. $|D_b| = 1$. Next, we progressively relax all these constraints. First of all, we explain how to represent a bid on a bundle of transformations on a WTPN.

In a combinatorial bid on transformations we have to ensure that if at least an atomic transformation in a bid is included in the solution sequence, the price that has to be paid to (received by) the bidder is the price of the whole bid. Therefore, we must guarantee the payment of the price p_b associated to a bid b , whenever one of the transformations in it is selected. We achieve this by introducing some auxiliary places and transitions. The example in table 1(a) represents a bid on the set of transformations $\{t_{b,1}, t_{b,2}, t_{b,3}\}$, whose associated price is $p_b = \$20$. For each bid b we introduce an auxiliary transition t_b (bid transitions) and an auxiliary place c_b . For each atomic transformation $t_{b,k}, k = 1 \dots |D_b|$, we add an auxiliary place $c_{b,k}$ ($c_{b,1}, c_{b,2}$, and $c_{b,3}$ in table 1(a)) (control places). It is easy to check that this particular topology allows for firing any subset of $\{t_{b,1}, t_{b,2}, t_{b,3}\}$. Notice also that firing at least one of the three transitions implies previously firing transition t_b , since this guarantees having tokens in the $c_{b,i}$ places. We associate the bid cost $p_b = \$20$ to transition t_b . Thus, we guarantee that firing at least one of the transitions implies firing also t_b , and therefore that money is paid to t_b . Any legal firing sequence on the net of table 1(a) guarantees that selecting at least one of the $t_{b,k}$ implies also selecting t_b .

In case of no free disposal on transformations the requirements are different. In this case either all or none of the $t_{b,k}$ must fire, and in the case they all fire, t_b must fire too. The topology in table 1(a) alone cannot guarantee such property. For instance, a marking in which transformations $t_{b,1}$ and $t_{b,2}$ fire, but not $t_{b,3}$, is legal but does not comply with the no-free disposal assumption. In order to enforce no free disposal on transformations, we simply impose some constraints on the final configuration of the net. Say that we impose that in the final configuration $c_{b,1}, c_{b,2}$ and $c_{b,3}$ contain no tokens ($\mathcal{M}_d(c_{b,1}) = 0, \mathcal{M}_d(c_{b,2}) = 0, \mathcal{M}_d(c_{b,3}) = 0$). This implies that all the legal firing sequences leading to the final configuration \mathcal{M}_d contain either none or all the three transitions $t_{b,1}, t_{b,2}, t_{b,3}$. In fact the only possible firing sequences are, either no firings $J = \{\}$, or $J = \{t_b t_{b,1}, t_b t_{b,2}, t_b t_{b,3}\}, J = \{t_b t_{b,1} t_{b,2}, t_b t_{b,1} t_{b,3}, \dots$

Finally, we have to express the XOR relationships among the bids that come from the same bidder. Say that a bidder submits two bids b and b' in XOR, i.e. they cannot be jointly selected. Table 1(b) depicts bids b and b' : bid b is over transformations $t_{b,1}, t_{b,2}$ and $t_{b,3}$; and bid b' is over transformations $t_{b',1}$ and $t_{b',2}$. The cost associated to b is $c(t_b) = \$20$, and the cost associated to b' is $c(t_{b'}) = \$10$. Notice that we introduce a new place, labelled with p_{XOR} , called XOR place. Notice, too, that we get rid of places c_b and $c_{b'}$. It is easy to verify that this topology inhibits the firing of both b and b' : when either of them fires, it consumes the unique token in p_{XOR} . This structure applies also to the case of m bids in XOR among them.

In what follows we formally define the mapping from

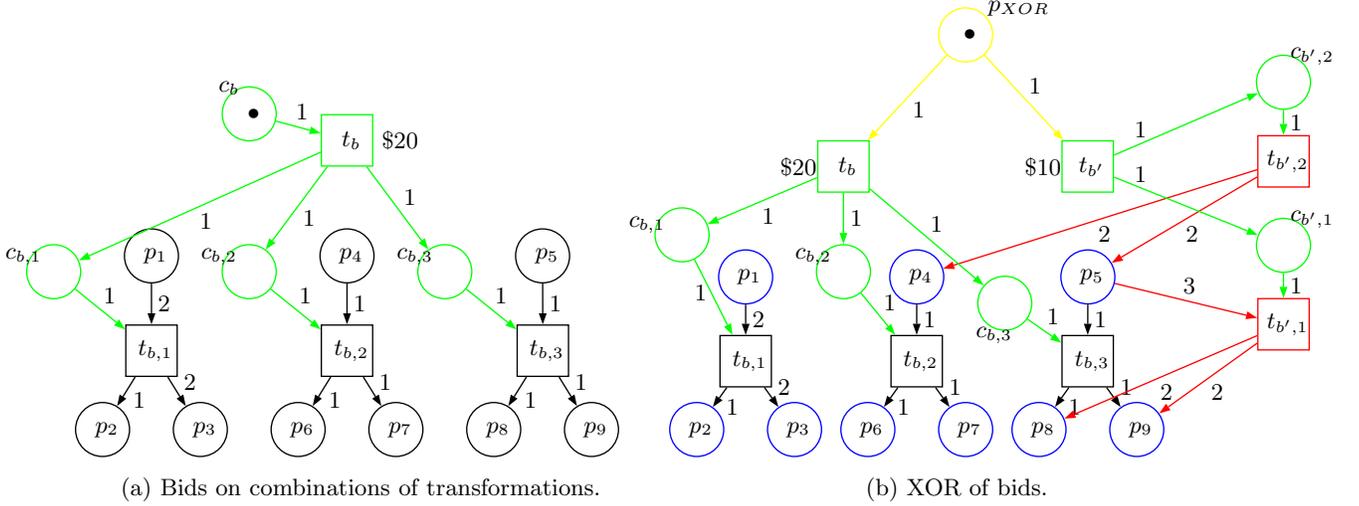


Table 1: Graphical representation of a bid.

MMUCA to CMWOSP over a WTPN. Such WTPN is built as follows:

- For each good $g \in G$ add a place p_g . We call these places *good places*. The set of all the good places is noted as P_G . $P_G = \{p_1, \dots, p_9\}$ in table 1(b).
- For each bid $b \in B$ add a transition t_b to the WTPN. We call these transitions *bid transitions*. The set of all bid transitions is noted as T_B . For example, $T_B = \{t_b, t_{b'}\}$ in table 1(b).
- For each bidder $l \in L$ add a place p_{XOR_l} . We call these places *XOR places*. The set of all XOR places is denoted as P_{XOR} . For instance, $P_{XOR} = \{p_{XOR}\}$ in table 1(b).
- For each atomic transformation in each bid $t_{b,k}, b \in B, k = 1, \dots, |D_b|$ add:
 - a transition $t_{b,k}$ (atomic transition). The set of all atomic transitions is denoted as T_{CT} . For example $T_{CT} = \{t_{b,1}, t_{b,2}, t_{b,3}, t_{b',1}, t_{b',2}\}$ in table 1(b).
 - a place $c_{b,k}$ to the WTPN (*control place*). The set of all control places is denoted as P_{CP} . For example, $P_{CP} = \{c_{b,1}, \dots, c_{b',2}\}$ in table 1(b).
- Build a set of arcs A_{AT} connecting the places representing the input goods and output goods of each atomic transformations $t_{b,k}$ to the transition representing it ($t_{b,k}$)⁴. The input goods are connected by incoming arcs whereas the output goods by outgoing arcs. For instance transition $t_{b,1}$ in figure corresponds to the atomic transformation $t_{b,1} = (\mathcal{I}_{b,1}, \mathcal{O}_{b,1}) = (\{g_1\}, \{g_2, g_3\})$. Therefore, place p_1 , representing input good g_1 , is connected to transition $t_{b,1}$ by means of an incoming arc; and places p_2 and p_3 , representing output goods g_2 and g_3 , are connected to $t_{b,1}$ by means

of outgoing arcs. More formally, for each atomic transformation $t_{b,k} = (\mathcal{I}_{b,k}, \mathcal{O}_{b,k})$ in each bid $b \in B$, with $k = 1, \dots, |D_b|$:

$$(p_g, t_{b,k}) \in A_{AT} \iff g \in \mathcal{I}_{b,k} \quad (15)$$

$$(t_{b,k}, p_g) \in A_{AT} \iff g \in \mathcal{O}_{b,k} \quad (16)$$

For example, $A_{AT} = \{(p_1, t_{b,1}), (t_{b,1}, p_2), (t_{b,1}, p_3), \dots, (p_5, t_{b',1}), (t_{b',1}, p_8), (t_{b',1}, p_9)\}$ in table 1(b).

- Build a set of arcs A_{BT} such that: (1) bid transitions t_b are connected to control places $c_{b,k}$; and (2) control places $c_{b,k}$ are connected to atomic transitions $t_{b,k}$:

$$(t_b, c_{b,k}) \in A_{BT} \quad b \in B, 1 \leq k \leq |D_b| \quad (17)$$

$$(c_{b,k}, t_{b,k}) \in A_{BT} \quad b \in B, 1 \leq k \leq |D_b| \quad (18)$$

For example $A_{BT} = \{(t_b, c_{b,1}), (c_{b,1}, t_{b,1}), (t_b, c_{b,2}), (c_{b,2}, t_{b,2}), \dots, (t_{b'}, c_{b',1}), (c_{b',1}, t_{b',1})\}$ in table 1(b).

- Build a set of arcs A_{XOR} connecting all the $p_{XOR_l}, l \in L$ places to the bid transitions t_b corresponding to bids coming from the same provider l .

$$(p_{XOR_l}, t_b) \in A_{XOR} \iff b \in B_l$$

$A_{XOR} = \{(p_{XOR}, t_b), (p_{XOR}, t_{b'})\}$ in table 1(b).

- The arc expression function is built as follows:

$$E(p_g, t_{b,k}) = \mathcal{I}_{b,k}(g) \iff (p_g, t_{b,k}) \in A_{AT} \quad (19)$$

$$E(t_{b,k}, p_g) = \mathcal{O}_{b,k}(g) \iff (t_{b,k}, p_g) \in A_{AT} \quad (20)$$

$$E(t, p) = 1 \iff (t, p) \in A_{BT} \cup A_{XOR} \quad (21)$$

$$E(p, t) = 1 \iff (p, t) \in A_{BT} \cup A_{XOR} \quad (22)$$

- The bid cost function $C : B \rightarrow \mathbb{R}$ is built as follows. In the case of atomic transformations the cost associated is 0: $c(t_{b,k}) = 0$ for each $b \in B, k = 1, \dots, |D_b|$. The price associated to bid b is associated to the firing of bid transition t_b , therefore $c(t_b) = p_b$, with $b \in B$.

⁴In this case we employ the same letters to indicate an atomic transformation and its corresponding transition in the WTPN.

- The initial marking is defined as

$$\mathcal{M}_0(p) = \begin{cases} \mathcal{U}_{in}(g) & p \in P_G \\ 1 & p \in P_{XOR} \\ 0 & otherwise \end{cases} \quad (23)$$

Thus, given a multiset of available goods \mathcal{U}_{in} , a set of required goods \mathcal{U}_{out} , and a set of bids in the XOR language⁵, solving the WDP for MMUCAs amounts to solve the CMWOSP defined on the WTPN $S^* = (P_G \cup P_{CP} \cup P_{XOR}, T \cup T_{CT}, A_{AT} \cup A_{BT} \cup A_{XOR}, E, C, \mathcal{M}_0)$, with destination marking \mathcal{M}_d fulfilling the following constraints⁶:

$$\mathcal{M}_d(p) \geq \mathcal{U}_{out}(g) \quad p_g \in P_G \quad (24)$$

$$\mathcal{M}_d(p) \geq 0 \quad p \in P_{CP} \quad (25)$$

$$\mathcal{M}_d(p) \geq 0 \quad otherwise \quad (26)$$

The converse holds as well.

Each firing sequence J_{max} solution to the CMWOSP can be transformed into an optimal solution sequence of the MMUCA WDP. First, for each $t_b \in T_{BT}$ such that $\exists x_{b,k} = 1$ set $x_b = 1$. Then, build an auxiliary sequence $J^* = J_{max} \setminus T_{BT}$ taking out the bid transitions (T_{BT}) from J_{max} . Next, if the $m - th$ element of the sequence J^* is $t_{b,k}$ set $x_{b,k}^m = 1$ and $x_b = 1$ and $x^m = 1$.

And the converse, given an optimal solution sequence represented by decision variables $x_{b,k}^m, x_b$, and x^m , it is possible to build an optimal firing sequence J_{max} solution to the CMWOSP. The WTPN must be built as explained in this section. The optimal firing sequence is composed as follows. First, compose a firing sequence J^* in which the $m - th$ element is the transition $t_{b,k}$ such $x_{b,k}^m = 1$. Secondly, compose J_{max} by inserting transition t_b if $x_b = 1$. Transition t_b must be inserted in J_{max} just before the first occurrence of a transition $t_{b,k}$ in the sequence.

We have not given a formal demonstration of the soundness of the mapping we defined. However, in what follows we provide the intuition behind this mapping. If we consider equations (3) and (4), we can easily check that they represent exactly the firing rules for a petri net. In particular equation (3) maps to equation (6) and equation (4) maps to equation (5).

5. CONCLUSIONS AND RELATED WORK

The Mixed Multi-Unit Combinatorial Auction, defined in [1] subsumes a range of combinatorial auction models discussed in the literature (see e.g. [2]). In particular, it increments considerably the expressiveness and the range of solvable problems for Combinatorial Auctions for supply chain formation [6]. MMUCAs incorporate the concept of a *sequence* of exchanges, which is required if the intention is to model some sort of production process.

Walsh et al. [6] tackle a similar problem to ours focusing on supply chain formation. Although their contribution is very significant, their work is extended by MMUCAs along several dimensions. Firstly, they do not allow a provider to

⁵Notice that in the case of OR language we could state exactly the same if we make appropriate changes to the WTPN. We should just represent all the bids as in table 1(a), i.e. omitting the XOR places.

⁶In case of no free disposal on goods simply substitute \geq with $=$ in equation (24). In case of no free disposal on transformations simply substitute \geq with $=$ in equation (25).

submit bids on combinations of transformations. Secondly, they do not define a bidding language (in fact, their agents submit a bid with a single transformation each). Finally, the transformation net that defines the supply chain has to fulfil strict criteria: acyclicity, transformations can only produce one output good, etc.

In this work, we identify as our main contribution the extension of a graphical formalism for analysing the structural and behavioural properties of MMUCAs, and therefore of all the types of auctions subsumed by MMUCAs, and in particular of combinatorial auctions for supply chain formation. As a first benefit of providing such an extension, we demonstrate how to dramatically reduce the number of decision variables (from quadratic to linear with respect to the number of transformations) for a broad subclass of MMUCAs WDPs (in particular, when the WTPNs underlying the MMUCA WDP is acyclic, trap-circuit, siphon-circuit, TCC, or SCC). Hence, we make headway in the practical application of MMUCAs as intended. To attain these goals we have resorted to Petri nets by defining a new type of Petri nets, the so-called weighted transition Petri nets (WTPN), and by defining a new optimisation problem over WTPN, the so-called CMWOSP.

The results we propose can be easily extended along several dimensions. First, it is known from the literature that it is possible to increase the classes of petri nets for which the state equation represents the whole reachability set. As an example one may add linear side constraints to the state equation [3]. Therefore, we would like to assess the applicability of these types of techniques to our problem. Second, the validity of the mapping from MMUCA WDP to WTPNS is not restricted to bids in the XOR language, but in fact it can easily cope with other languages. For instance, as explained in section 4, the extension to the OR language is trivial. Third, and most important, our mapping allows to analyse structural and behavioural properties of the solutions to the MMUCA WDP.

6. REFERENCES

- [1] J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodriguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *Proc. of the 20th Intl. Joint Conferences on Artif. Intelligence (IJCAI)*, pages 1221–1226, Hyderabad, India, 2007.
- [2] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [3] J. Esparza and S. Melzer. Verification of safety properties using integer programming: Beyond the state equation. *Formal Methods in System Design*, 16:159–189, 2004.
- [4] T. Murata. Petri nets: Properties, analysis and applications. In *IEEE*, volume 77, pages 541–580, 1989.
- [5] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006.
- [6] W. E. Walsh, M. P. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Proc. of the 2nd ACM Conference on Electronic Commerce*, 2000.
- [7] T. Watanabe. The legal firing sequence problem of petri nets. *IEICE Transactions on Information and Systems*, 3:397–406, 2000.