Solving Sequential Mixed Auctions with Integer Programming

Boris Mikhaylov, Jesus Cerquides, and Juan A. Rodriguez-Aguilar

Artificial Intelligence Research Institute,
IIIA Spanish National Research Council, CSIC
{boris,cerquide,jar}@iiia.csic.es

Abstract. Mixed multi-unit combinatorial auctions (MMUCAs) offer a high potential to be employed for the automated assembly of supply chains of agents. However, in order for mixed auctions to be effectively applied to supply chain formation, we must ensure computational tractability and reduce bidders' uncertainty. With this aim, we introduce Sequential Mixed Auctions (SMAs), a novel auction model conceived to help bidders collaboratively discover supply chain structures. Thus, an SMA allows bidders progressively build a supply chain structure through successive auction rounds.

1 Introduction

According to [7], "Supply Chain Formation (SCF) is the process of determining the participants in a supply chain, who will exchange what with whom, and the terms of the exchanges". Combinatorial Auctions (CAs) [2] are a negotiation mechanism well suited to deal with complementarities among the goods at trade. Since production technologies often have to deal with strong complementarities, SCF automation appears as a very promising application area for CAs. However, whilst in CAs the complementarities can be simply represented as relationships among goods, in SCF the complementarities involve not only goods, but also *transformations* (production relationships) along several levels of the supply chain.

The first attempt to deal with the SCF problem by means of CAs was done by Walsh et al. in [7]. Later on, mixed multi-unit combinatorial auctions (MMUCAs), a generalization of the standard model of CAs, are introduced in [1]. Rather than negotiating over goods, in MMUCAs the auctioneer and the bidders can negotiate over *transformations*, each one characterized by a set of input goods and a set of output goods. A bidder offering a transformation is willing to produce its output goods after having received its input goods along with the payment specified in the bid. While in standard CAs, a solution to the winner determination problem (WDP) is a set of atomic bids to accept, in MMUCAs, the *order* in which the auctioneer "uses" the accepted transformations matters. Thus, a *solution* to the WDP is a *sequence of transformations*. For instance, if bidder *Joe* offers to make dough if provided with butter and eggs, and bidder

J.A. Lozano, J.A. Gámez, and J.A. Moreno (Eds.): CAEPIA 2011, LNAI 7023, pp. 42–53, 2011.

Lou offers to bake a cake if provided with enough dough, the auctioneer can accept both bids whenever he uses Joe's transformation before Lou's to obtain cakes. Unfortunately, the MMUCA WDP has been proved to be NP-complete [1]. Although reasonably fast solvers have been introduced [4], MMUCA still turns out to be impractical in real-world procurement scenarios. Furthermore, a bidder in MMUCA only knows the desired outcome of the supply chain and the current stock goods. Hence, it is difficult, specially for providers placed in the intermediate levels of the supply chain, to decide what to bid for. Therefore, in order for mixed auctions to be effectively applied to SCF, we must ensure computational tractability and reduce bidders' uncertainty. With this aim, we introduce Sequential Mixed Auctions (SMAs), a novel auction model conceived to help bidders collaboratively discover supply chain structures.

SMAs propose to solve a SCF problem by means of a sequence of auctions. The first auctioning round starts with the desired outcome of the supply chain as requested goods and the stock goods as available goods. During the first auction, bidders are only allowed to bid for transformations that either (i) produce goods in the set of requested goods or (ii) consume goods from the available goods. After selecting the best set of transformations, the auctioneer updates the set of requested and available goods after the execution of these transformations and then it will start a new auction. The process continues till no bids can be found that improve the supply chain. Notice that each auction in the sequence involves only a small part of the whole supply chain, instead of the whole one as MMUCAs do. Thus, auctions in an SMA are much less computationally demanding than a MMUCA. Moreover, the incremental nature of an SMA provides its participants with valuable information at the end of each auction round to guide their bidding.

The paper is organised as follows. Section 2 provides some background of mixed auctions, whereas section 3 formally states the WDP and section 4 details a mixed integer pogram that solves it. Finally, section 5 concludes.

2 Background: Mixed Auctions

Next we summarise the work in [1], which introduces mixed auctions (MMUCAs) as a generalisation of CAs and discusses the issues of bidding and winner determination.

Let G be the finite set of all the types of goods. A *transformation* is a pair of multi-sets 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 $\mathcal{I}$. Bidders can offer any number of such transformations, including several copies of the same transformation. That is, agents negotiate over *multi-sets 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 can deliver a (no input required) and that it can deliver c if provided with b.

Since agents negotiate over bundles of transformations, a *valuation* $v : \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)} \to \mathbb{R}$ is a mapping from multi-sets of transformations to real numbers. Intuitively, $v(\mathcal{D}) = p$ means that the agent equipped with valuation v is willing

to make a payment of p for being allocated all the transformations in $\mathcal{D}$ (in case p is negative, this means that the agent will accept the deal if it *receives* an amount of $|p|$). For instance, valuation $v(\{(\{line, ring, head, 6 \cdot screws, screwdriver\}, \{cylinder, screwdriver\})\}) = -10$ means that some agent can assemble a cylinder for 10€ when provided with a (cylinder) line, a (cylinder) ring, a (cylinder) head, six screws, and a screwdriver, and returns the screwdriver once done.[1]

An *atomic bid* $b = (\{(\mathcal{I}^1, \mathcal{O}^1), \ldots, (\mathcal{I}^n, \mathcal{O}^n)\}, p, \beta)$ specifies a finite multi-set of finite transformations, a price p and the bidder β. A *bidding language* allows a bidder to encode choices between alternative bids and the like [6]. Informally, an OR-combination of several bids means 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. The XOR-language is known to be fully expressive for MMUCAs [1]. Bids in MMUCAs are composed of transformations. Each transformation expresses either an offer to buy, to sell, or to transform some good(s) into (an)other good(s). Thus, transformations are the building blocks composing bids. We can classify the types of transformations over which agents bid as follows:

1. Output Transformations are those with no input good(s). Thus, an O-transformation represents a bidder's offer to sell some good(s).
2. Input Transformations are those with no output good(s). Thus, an I-transformation represents a bidder's offer to buy some good(s).
3. Input-Output Transformations are those whose input and output good(s) are not empty. An IO-transformation stands for a bidder's offer to deliver some good(s) after receiving some other good(s): *I can deliver* $\mathcal{O}$ *after having received* $\mathcal{I}$. They can model a wide range of different processes in real-world situations (e.g. assembly, transformation, or exchange).

The *input* to the WDP consists of a complex bid expression for each bidder, a multi-set $\mathcal{U}_{in}$ of (stock) goods the auctioneer holds to begin with, and a multi-set $\mathcal{U}_{out}$ of (required) goods the auctioneer expects to end up with. In standard CAs, a solution to the WDP is a set of atomic bids to accept. As to MMUCAs, 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 $Bid_1 = (\{(\{a\}, \{b\})\}, 10, id_1)$ and $Bid_2 = (\{(\{b\}, \{c\})\}, 20, id_2)$ is feasible involves realizing that we have to use Bid_1 before Bid_2. Thus, a *valid solution* to the WDP will be a *sequence of transformations* that satisfies:

(1) *Bidder constraints:* The multi-set 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

[1] We use $6 \cdot screws$ as a shorthand to represent six identical elements in the multi-set.

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 $\mathcal{U}_{out}$.

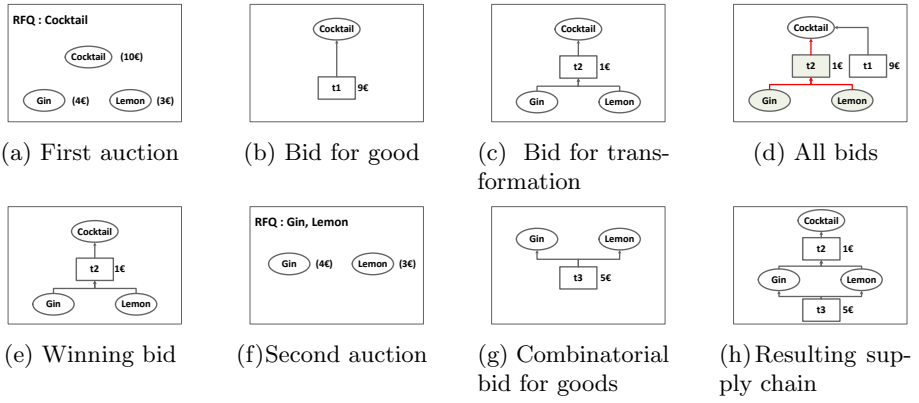
An *optimal* solution is a valid solution that maximizes the sum of prices associated with the atomic bids selected.

The WDP for MMUCAs is a complex computational problem. In fact, one of the fundamental issues limiting the applicability of MMUCAs to real-world scenarios is the computational complexity of the WDP, which is proved in [1] to be $\mathcal{NP}$-complete. Although [4] introduces an integer program to efficiently solve the WDP that drastically outperforms the original IP described in [1], the computational complexity impedes scalability. The next section introduces a new mixed auction model that allows to tame complexity while reducing bidders' uncertainty.

3 Sequential Mixed Auctions

An SMA proposes to solve the SCF problem by means of a sequence of auctions. The first auction in the sequence starts with the desired outcome of the supply chain as requested goods and the stocked goods as available goods. During the first auction, bidders are only allowed to bid for transformations that either: (i) produce goods in the set of requested goods; or (ii) consume goods from the available goods. After selecting the winning bids (the best set of transformations), the auctioneer updates the set of requested and available goods after the execution of these transformations. Moreover, the winning bids are included as part of the supply chain. Thereafter, the auctioneer starts a new auction in the sequence. The process continues until no bids can improve the supply chain. Hence, the purpose of the auctioneer is to use a sequence of auctions to progressively build the structure of the supply chain.

Figure 1 illustrates the operation of an SMA. Say that a cocktail bar intends to form a supply chain using an SMA to produce a gin & lemon cocktail. Assume that the bar knows approximate market prices for a gin & lemon cocktail as well as for its ingredients. The auctioneer starts the first auction in the SMA issuing a request for quotation (RFQ) for a gin & lemon cocktail. Figure 1a depicts the RFQ along with each good's market price in brackets (e.g. the expected market price of 1 liter of gin is 4€). During the first auction, the auctioneer received two bids: one offering to deliver a cocktail for 9€ (figure 1b); and another one to make a cocktail for 1€ when provided with lemon and gin (figure 1c). The auctioneer must now choose the winning bid out of the bids in figure 1d. However, notice that the bid in figure 1c can only be used whenever some provider(s) offer gin and lemon. Thus, the auctioneer assesses the *expected price* of the bid using



(a) First auction (b) Bid for good (c) Bid for transformation (d) All bids

(e) Winning bid (f) Second auction (g) Combinatorial bid for goods (h) Resulting supply chain

Fig. 1. Example of sequential mixed auction

the market prices of gin (4€) and lemon (3€). Since the expected price is $8(=1+4+3)$€, the auctioneer chooses this bid as the winning bid and discards the bid in figure 1b, namely buying the cocktail for 9€.

At this point, the structure of the supply chain is the one depicted in figure 1e. Nonetheless, in order to run the supply chain, the auctioneer must still find providers of gin and lemon. With this aim, the auctioneer starts a new auction of the SMA by issuing an RFQ for gin and lemon (figure 1f). According to our example, this time the auctioneer only receives the combinatorial bid in figure 1g, which offers both lemon and gin for 5€. Since the bid is cheaper than the overall market price of both gin and lemon (4€+3€), this bid is selected as the winning bid of the second auction. Figure 1h shows the resulting structure of the supply chain after the second auction. Since there are no further goods to allocate, the auctioneer closes the SMA. The resulting supply chain produces a cocktail at the cost of 6€.

Although the SMA in this example obtains the optimal solution, this is not always the case. In general, at the end of each auction the auctioneer discards some bids because other bids are *expected* to lead to cheaper solutions. For instance, the bid in figure 1b is discarded to favour the bid in figure 1c. Therefore, since discarded bids might eventually lead to better solutions during subsequent auctions, unlike an MMUCA, an SMA is not guaranteed to obtain an optimal solution (sequence of transformations). Although SMAs may lose optimality, the example anticipates how an SMA help cope with computational complexity and bidders' uncertainty. Firstly, an SMA breaks the formation of a supply chain into several auctions, instead of running a single auction with all bids as MMUCA does. Secondly, after each auction in an SMA, bidders are informed about the needs of the supply chain. Therefore, the auctioneer guides bidders after each tier of the supply chain is formed, hence reducing their uncertainty with respect to participating in MMUCAs (MMUCA bidders only know the expected final outcome of the supply chain!).

3.1 Defining the Winner Determination Problem

An SMA is essentially a sequence of auctions (henceforth step auctions). For instance, the SMA in figure 1 is composed of two consecutive auctions. Each step auction receives a set of stock goods and final goods along with the bids submitted by bidders. Then the auctioneer solves the WDP to assess the winning bids as well as the remaining stock goods and required final goods, which are passed on to the next auction in the sequence. When solving the WDP, we assume that the auctioneer is aware of the market prices of goods so that it can compute the expected price of bids when necessary. The sequence of auctions continues till an auction either: (i) obtains a set of winning bids that produce the required goods while consuming all the stock goods; or (ii) does not receive any bids that can improve the supply chain. At this point, the winning bids of the last step auction stand for the SMA solution.

Next, we focus on formally defining the WDP faced by the auctioneer during each step auction of an SMA. Henceforth, we consider that the auctioneer holds a multi-set $\mathcal{U}_{in}$ of stock goods to begin with and expects to end up with a multi-set $\mathcal{U}_{out}$ of required (needed) goods. These are the input to the first auction in the SMA. For the formal definition of the WDP, we restrict ourselves to bids in the XOR-language, which is known to be fully expressive. Let C be the set of bidders. Let B be the set of all atomic bids. An atomic bid $b = (\mathcal{D}_b, p_b, \beta_b)$ 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 $\beta_b \in C$. Let B_β be the set of all atomic bids submitted by bidder $\beta \in C$. Note that a bid can offer several units of the very same transformation. For each bid b, let t_{bk} be a unique label for the kth different transformation offered by bid b. Let $(\mathcal{I}_{bk}, \mathcal{O}_{bk})$ be the actual transformation labelled by t_{bk}.

At the l-th step auction in an SMA, let T_l be the set of labels t_{bk} for all transformations mentioned anywhere in the bids received by the auctioneer. The auctioneer has to decide which transformations to accept and the order to implement them. Thus, an allocation sequence Σ_l is an ordered list of a subset of the transformations in T_l. We write $t_{bk} \in \Sigma_l$ to say that the k-th transformation in bid b has been selected and $|\Sigma_l|_{t_{bk}}$ for the number of times that t_{bk} appears in Σ_l. Intuitively, an allocation sequence for a step auction is a valid solution iff: (i) it fulfills the semantics of the bidding language; (ii) it inherits all the transformations in the valid solution of the previous step auction while preserving their ordering; and (iii) the new transformations in the sequence (not inherited from the previous step auction) offer to either buy produced goods or sell required goods from the previous step auction. The last condition ensures that the new transformations accepted by the auctioneer either provide goods required to use the transformations in the previous step auction or consume goods produced in the same step auction. We are now ready to define under what circumstances a sequence of transformations constitutes a valid solution for a step auction in an SMA.

Definition 1 (Valid solution of a step auction). *Given an SMA, let Σ_{l-1} be a valid solution, and Buy_{l-1} and $Sell_{l-1}$ the multi-sets presenting the required*

and available goods after the $(l-1)$-th step auction. An allocation sequence Σ_l of the l-th step auction for a given set of atomic bids B is said to be a valid solution iff:

1. *Σ_l either contains all or none of the transformations belonging to the same atomic bid.*
2. *Σ_l does not contain two transformations belonging to different atomic bids by the same bidder.*
3. *Each transformation in Σ_{l-1} belongs also to Σ_l.*
4. *Σ_l preservers the order of transformations in Σ_{l-1}. Thus, for every two transformation $t, t' \in \Sigma_{l-1}$, if t appears before t' in the allocation sequence Σ_{l-1}, it also appears before t' in the allocation sequence Σ_l.*
5. *For each transformation in Σ_l that is not in Σ_{l-1}, either some of its input goods are in $Sell_{l-1}$, some of its output goods are in Buy_{l-1}, or both.*

For the first auction of the SMA, $\langle \{\}, \mathcal{U}_{out}, \mathcal{U}_{in} \rangle$ stand for the valid solution, and the needed and stock goods.

In order to assess the expected revenue of a valid solution Σ_l, we must first compute the goods that the auctioneer should buy and sell in the market to implement the solution, namely to *use* all the transformations in the sequence. First, we compute the units of each good produced by a sequence Σ_l as:

$$P_l(g) = \sum_{t_{bk} \in \Sigma_l} |\Sigma_l|_{t_{bk}} \cdot [\mathcal{O}_{bk}(g) - \mathcal{I}_{bk}(g)] \tag{1}$$

Hence, we can obtain the number of units of each good held by the auctioneer after all the transformations in the sequence are used as:

$$Q_l(g) = \mathcal{U}_{in}(g) + P_l(g) - \mathcal{U}_{out}(g)$$

Now, we assess the units of each good to buy or sell in the market as:

$$Buy_l(g) = \begin{cases} 0 & \text{if } Q_l(g) > 0 \\ -Q_l(g) & \text{otherwise} \end{cases} \qquad Sell_l(g) = \begin{cases} Q_l(g) & \text{if } Q_l(g) > 0 \\ 0 & \text{otherwise} \end{cases}.$$

Notice that in fact Buy_l and $Sell_l$ stand for the remaining required goods and available stock goods that step auction l passes on to the next auction. Now, assuming that the auctioneer knows the expected market prices at which goods can be bought ($P^- : G \to \mathbb{R}$) and sold ($P^+ : G \to \mathbb{R}$), the acutioneer can compute the *expected* revenue of a valid solution.

Definition 2 (Expected revenue of a valid solution). *The expected revenue of a valid solution Σ_l for step auction l is the sum of the prices associated with the selected atomic bids minus the expected prices of the goods that are required to be bought (Buy_l) plus the expected prices of the goods that are sold ($Sell_l$) in the market, namely:*

$$\sum_{b \in Selected(\Sigma_l)} p_b - \sum_{g \in G} Buy_l(g) P^-(g) + \sum_{g \in G} Sell_l(g) P^+(g)$$

where $Selected(\Sigma_l) = \{b \in B | \exists k : t_{bk} \in \Sigma_l\}$ is the set of selected atomic bids.

Definition 3 (WDP of a step auction). *Given multi-sets $\mathcal{U}_{in}$ and $\mathcal{U}_{out}$ of initial and final goods for an SMA, a set of atomic bids B for step auction l, a valid solution Σ_{l-1} of step auction $l-1$, and Buy_{l-1} and $Sell_{l-1}$ contain the units of each good to buy or sell in the market after step auction $l-1$, the winner determination problem for step auction l is the problem of finding a valid solution Σ_l that maximizes the expected revenue for the auctioneer.*

We say that a sequence of auctions is *complete* when the solution to the last step auction (n) in the sequence either: (i) produces all the required goods and consumes all the stock goods ($Buy_n = \{\}$ and $Sell_n = \{\}$); or (ii) is equal to the solution to the previous step auction (namely $\Sigma_n = \Sigma_{n-1}$. The first condition occurs when the auctioneer fully satisfies his initial requirements, while the second condition occurs when the auctioneer cannot find further bids that improve the current solution, hence the current supply chain. Whatever the case, the valid solution Σ_n contains the solution to the SMA.

4 Solving an SMA Step by Means of a Mixed Integer Linear Program

As outlined above, solving an SMA amounts to solve the WDP for the first step auction, then for the second step auction and so on and so forth until the SMA is complete. Hence, the key computational problem is how to solve the WDP for a step auction. In this section, we first summarise the Connected Component IP Solver (CCIP), which maps the original MMUCA WDP into a mixed integer program. Then, we modify it to solve an SMA step auction.

4.1 Connected Component IP Solver (CCIP)

In order to create a linear program we need to determine the maximum length of the allocation sequence. The worst case is assuming that every possible bid is accepted and hence every transformation in used. The total number of transformations (including multiple copies of the same transformation) can be assessed as $r = \sum_{b \in B} |\mathcal{D}_b|$.

Let T_b be the set of all t_{bk} for bid b. Let T be the set of all t_{bk}, namely the set of all distinguishable transformations (no copies of the very same transformation) mentioned anywhere in the bids. The auctioneer has to decide which transformations to accept and the order to implement them. At this aim, we represent each solution with a partial sequence $J : \{1, \ldots, r\} \to T$. We employ the following decision variables: x_b is a binary variable that takes value 1 if bid b is accepted; x_{bk}^m is a binary variable that takes value 1 only if transformation t_{bk} is selected at the m-th position within the solution sequence (i.e. $J(m) = t_{bk}$).

Let S be a solution template, restricting the positions at which transformations can be executed. For each position m, $S(m)$ is a set containing the subset of transformations that can be executed at position m in the sequence. For more

information on how to define solution templates see [4]. We will only allow as solutions partial sequences fulfilling S. Hence we only have to create those variables x_{bk}^m such that $t_{bk} \in S(m)$.

We introduce two additional definitions to ease encoding the solver. First, x_{bk} represents the number of times that transformation t_{bk} is used in the solution sequence, assessed as:

$$x_{bk} = \sum_{m \in S^{-1}(t_{bk})} x_{bk}^m.$$

Second, we need to see how to assess the stock of a given good after m steps in terms of the decision variables we have defined. Say that we represent with the multiset of goods $Stock^m$ the quantity of resources available to the auctioneer after performing m steps.

Now, $\mathcal{P}^m(g)$, the units of good g available at the end of step m can be assessed as

$$\mathcal{P}^m(g) = \sum_{i=1}^{m} \sum_{t_{bk} \in S(m)} x_{bk}^i \cdot (\mathcal{O}_{bk}(g) - \mathcal{I}_{bk}(g)).$$

That is, for each time step from 1 to m, we add the output goods of the transformation executed at that step and remove its input goods. Hence, the stock of good g after step m is:

$$Stock^m(g) = \mathcal{U}_{in}(g) + \mathcal{P}^m(g) \tag{2}$$

Finally, the constraints that a valid solution has to fulfil in solver CCIP are:

1. We enforce that, whenever a bid is selected each of the transformations in that bid is used as many times as it is offered in the bid. Formally:

$$x_{bk} = x_b \cdot |\mathcal{D}_b|_{t_{bk}} \qquad \forall b \in B, \quad \forall k \in \{1, \ldots, |T_b|\} \tag{3}$$

   where $|\mathcal{D}_b|_{t_{bk}}$ is the multiplicity of transformation t_{bk} in bid b.

2. We enforce that at most one bid can be accepted per bidder (XOR constraint):

$$\sum_{b \in B_\beta} x_b \leq 1 \qquad \forall \beta \in C. \tag{4}$$

3. We enforce that enough goods are available to use the corresponding transformations at each position of the solution sequence. This constraint, represented by equation 5 below, is only needed for some of the positions (see [3] for a detailed description on L_F, the set of positions in which it must be enforced). Formally:

$$Stock^{m-1}(g) \geq \sum_{t_{bk} \in S(m)} x_{bk}^m \cdot \mathcal{I}_{bk}(g) \qquad \forall g \in G, \quad \forall m \in L_F \tag{5}$$

4. We enforce that, after having performed all the selected transformations, the goods held by the auctioneer must be more than the goods that he requested, namely at least $\mathcal{U}_{out}$:

$$Stock^m(g) \geq \mathcal{U}_{out}(g) \qquad \forall g \in G \tag{6}$$

Hence, solving the MMUCA WDP amounts to maximising the objective function:

$$\sum_{b \in B} x_b \cdot p_b \tag{7}$$

subject to inequations 3 to 6.

4.2 Solving the WDP for an SMA Step

Next we focus on modyfing the CCIP solver so that it can solve an SMA step auction. Recall that $\Sigma_{l-1} = \langle t^1, \ldots, t^q \rangle$ is the sequence describing the transformations accepted in a previous auction and their execution order and that B contains the set of bids received at this step.

There are two modifications that need to be put in place. First, we need to ensure that every transformation accepted in the previous SMA step, does also appear in solution of the current step and that the order in which they are executed is also maintained in the new solution. Second, we need to take into account the possibility of buying from and selling to the market.

To ensure that all the transformations accepted in the previous SMA step, do also appear in solution of the current step we add an additional bidder to C, and a single atomic bid to B, namely b_0. b_0 is a combinatorial bid offering the set of all the transformations accepted in the previous SMA step (that is, the transformations in Σ_{l-1}) at no cost. Hence, from now on we can refer to any transformation $t^k \in \Sigma_{l-1}$ as $t_{b_0 k}$. Now, to ensure that this bid is taken, we add the additional constraint

$$x_{b_0} = 1. \tag{8}$$

We do also need to ensure that the ordering in the sequence of transformations accepted in the previous SMA step is maintained in the solution of this step. To encode this, for any transformation t_{bk} we define the time at which it is fired as $f_{bk} = \sum_{m \in S^{-1}(t_{bk})} m \cdot x^m_{bk}$. Now, for each consecutive pair of transformations we need to ensure that they are placed in the correct order in the solution sequence of this SMA step. That is:

$$f_{b_0 k} \leq f_{b_0 k+1} \qquad \forall k \in \{1, \cdots, q-1\}. \tag{9}$$

We take into account the possibility of buying from and selling to the market by introducing a new integer decision variable y_g , for each good at trade. y_g represents the number of units of good g that will be bought directly from the market. We can assume that goods are bought from the market before the solution sequence starts its execution. Hence, we can modify the expression for $Stock^m$ in equation 2 to consider the goods bought from the market:

$$Stock^m(g) = \mathcal{U}_{in}(g) + \mathcal{P}^m(g) + y_g. \tag{10}$$

Furthermore, we need to modify the objective function in 7 so that it takes into account the costs of buying from the market and the benefits obtained by selling to the market as described in definition 2. Note that $Sell(g)$, the units of good g that will be sold to the market can be assessed as the stock for that good after executing the last step minus the requirements of the auctioneer for that good, that is $Sell(g) = Stock^r(g) - U_{out}(g)$. The objective function for a step auction is written as:

$$\sum_{b \in B} x_b \cdot p_b - \sum_{g \in G} y_g \cdot P^-(g) + \sum_{g \in G} Sell(g) \cdot P^+(g) \tag{11}$$

Hence, solving the SMA step WDP amounts to maximising objective function 11 subject to constraints 8 and 9 and to the regular MMUCA constraints 3 to 6 with the definition of *Stock* modified as in equation 10.

5 Conclusions and Future Work

In this work, we continue the approach introduced in [5] to make mixed auctions applicable to supply chain formation in real-world procurement scenarios.

Following [5], to cope with the extensive computing times of MMUCA and bidder's uncertainties we moved the supply chain formation process from a single auction to a sequence of auctions. At each step auction, bidders are only allowed to bid on transformations that consume available goods or produce requested goods. After selecting the best set of transformations, the auctioneer updates the set of requested and available goods. The sequence ends when supply chain cannot be further improved. Each auction deals with just a small part of the supply chain. Thus, while solving the WDP for an individual auction we deal with small subsets of bidders, goods and transformations of former MMUCA. Preliminary results in [5] have shown savings on solution times up to 6 times while maintaining a reasonable quality.

The main contribution of this paper, the mapping of a step auction to a mixed integer linear program, is less restrictive that the approach introduced in [5], based on keeping a strict ordering among transformations. Hence, future experimental results using this mapping are expected to increase solution quality.

Acknowledgements. Work funded by projects EVE (TIN2009-14702-C02-01 and 02), AT (CONSOLIDER CSD2007-0022), Generalitat de Catalunya (2009-SGR-1434) and CSIC 201050I008.

References

1. Cerquides, J., Endriss, U., Giovannucci, A., Rodriguez-Aguilar, J.A.: Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In: IJCAI, Hyderabad, India, pp. 1221–1226 (2007)

2. Cramton, P., Shoham, Y., Steinberg, R. (eds.): Combinatorial Auctions. MIT Press (2006)
3. Giovannucci, A.: Computationally Manageable Combinatorial Auctions for Supply Chain Automation. PhD thesis, Universitat Autonoma de Barcelona (2007)
4. Giovannucci, A., Vinyals, M., Cerquides, J., Rodriguez-Aguilar, J.A.: Computationally-efficient winner determination for mixed multi-unit combinatorial auctions. In: AAMAS, Estoril, Portugal, May 12-16, pp. 1071–1078 (2008)
5. Mikhaylov, B., Cerquides, J., Rodriguez-Aguilar, J.A.: Sequential mixed auctions for supply chain formation. In: International Conference on Electronic Commerce (2011)
6. Nisan, N.: Bidding Languages for Combinatorial Auctions. ch. 9 Combinatorial Auctions. MIT Press (2006)
7. Walsh, W.E., Wellman, M.P.: Decentralized supply chain formation: A market protocol and competitive equilibrium analysis. Journal of Artificial Intelligence Research 19, 513–567 (2003)