A Generalised Network Flow Approach to Combinatorial Auctions

Madalina Croitoru, Cornelius Croitoru

A generalised network flow approach to combinatorial auctions

Madalina Croitoru [1], **Cornelius Croitoru**[2]
[1]LIRMM (CNRS and Univ. Montpellier), France
[2]Universitatea Al. I. Cuza, Iasi, Romania

Abstract

In this paper we address the problem of (1) representing bids for combinatorial auctions and (2) employing those structures for "reasoning". We propose a graph-based language who's novelty lies (1) in the use of generalized network flows to represent the bids and (2) in the interpretation of winner determination as an adequate aggregation of individual preferences. We motivate the language both from representational and reasoning points of view and show how our language represents the same class of expressivity of bids more concisely compared to existing work.

1 Introduction

In every Artificial Intelligence system addressing a given problem there is a need to (1) represent the state of the world and (2) reason about possible ways to solve the problem. In this paper we address the problem of (1) representing bids for combinatorial auctions and (2) employing those structures for "reasoning" (winner determination). The proposed language we detail is a visual, graph-based language based on network flow modelling techniques that demonstrate better conciseness within the same expressivity classes.

Combinatorial auctions (CAs) can be looked at as a way of approaching allocation problems involving multiple heterogeneous goods. Bidding is the problem of representing one's valuation function over the set of goods on offer. It plays a key role in both central aspects of the allocation problem: preference elicitation and winner-determination (WD). As a consequence bidding languages have not only to address representational issues but also to provide subsequent manipulation techniques for reasoning aspects. Our motivation for introducing a new language is based on the fact that existing languages cannot concisely represent some structured valuations that might occur in practical scenarios. Moreover, these languages were not designed with partial value revelation in mind; this is especially important in domains where the valuation problem is hard. Following from above mentioned representational choices the algorithms for winner determination cannot fully take advantage of the structural optimisation potential of the problem at hand.

This paper proposes a visual language for combinatorial auctions based on generalised flow networks. The nodes of the network will represent either (1) resources, (2) bundles of resources or (3) composite nodes used for calculation of certain partial valuations. The flow defined on the edges will allow concise description of an exponential number of bids. The same structure will be used for the auctioneer to unify all bidders' valuations. The winner determination problem will then be translated into a special MAX FLOW problem on the proposed network structure.

The paper is structured as follows. In section 2 we motivate our language from a representational viewpoint and demonstrate its conciseness. The formal, rigorous semantics of the language are introduced in section 3. Based on the constructs from section 3 we guarantee the soundness of the syntax further introduced in section 4. Section 5 concludes the paper.

2 Motivation

A hypergraph (or a *bundle system*) is a pair $H = (R, \mathcal{B})$ where R is a finite set (the *resources* set, the set of *goods*) called the *vertex set* of H and $\mathcal{B}$ is a family of subsets of R. The members of $\mathcal{B}$ are called *hyperedges* and they are subsets of resources, or *bundles*. A hypergraph H can be explicitly represented in visual manner by a bipartite graph $B(H)$ having one vertex class corresponding to the resources set R and the other class corresponding to the H's bundles, and connecting by an edge a bundle vertex to its corresponding (members) resource vertices. This is shown in Figure 1.

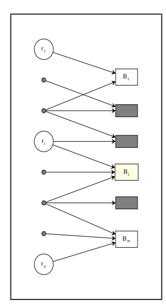

Figure 1: Bipartite graph representation of a bundle system

The bipartite graph has $|R| + |\mathcal{B}|$ vertices and $\sum_{B \in \mathcal{B}} |B|$ edges. If H is given explicitly, this is a concise and intuitive representation for a bundle system. The (directed) edges of this bipartite graph suggest the containment relation of resources to bundles. However, if H is given by using some constructive (or implicitly) rules, the bipartite representation must be extended in order to be an effective representational tool. For example, if $\mathcal{B}$ is the family of all bundles having $\lceil \frac{|R|}{2} \rceil$ resources, then the corresponding bipartite graph has an exponential number of bundle vertices and edges. We will extend the above containment relation (of resources to bundles) by using paths (a resource belongs to a bundle if and only if there is a certain path from the resource vertex to the bundle vertex) and a mechanism to express which path must be considered in order to instantiate a given bundle. This mechanism is based on a simple extension of network flows, which is described bellow.

In our representational networks we will use the following graphical primitives depicted in Figure 2:

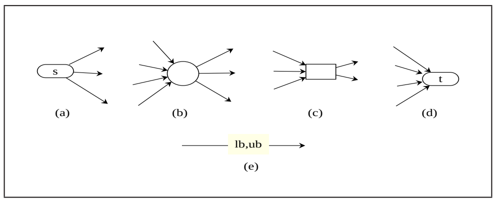

Figure 2: Elements of a bidflow network

The node (a) is the *start* node of the network (sometimes the label *start* is used instead of s). From this node the *flow* is pushed (on the arcs leaving it) in the network. The flow on each arc is a nonnegative integer value. If the flow f_{ij} on an the arc ij is positive it must satisfy the restriction $lb_{ij} \leq f_{ij} \leq ub_{ij}$, where the *lower bound* lb (sometimes denoted by l), and the *upper bound* ub (sometimes called *capacity* and denoted by c) are indicated as labels on the arc, as in construction (e) in Figure 2. The arcs without bound constraints (having $lb = 0$ and $ub = \infty$) are not labelled.

The nodes of type (b) are *transit nodes*, that is nodes which automatically distribute the total incoming flow (the sum of the flows on all arcs entering such a node) on the arcs leaving it. In other words, in these nodes the flow conservation law holds. They can have name-labels inside of the oval, for modelling or referring necessities.

The start node is connected by an arc labelled $0, 1$ to a node of type (b) labelled r, for each resource $r \in R$. From the flow conservation law (which holds in the transit node labelled r) and by the integrality of flows, either the flow on the arc sr is 1, and there is exactly one arc with flow value 1 leaving the node r, or the flow on the arc sr is 0, meaning that the resource r will belong to no bundle. In the former case, a path with positive flows on its arcs will be constructed, which eventually will reach a type (c) node.

The nodes of type (c) are *bundle nodes*, which pass-on the incoming flow exactly on one arc leaving them. The flow on this arc is set to 1. Furthermore, a bundle node b is “on” only if all the flows on the arcs entering it are positive. The intuition is that such a node collects all the resources $r \in R$ which belongs to a path starting from s, having positive flows on its arcs and ending in b. If the bundle represented by the node b is a not a member of $\mathcal{B}$ (the bundle system to be represented by the network) then the arcs leaving b are used to simulate (disjoint) unions in order to construct such a member via paths with positive arc flows. Of course, the structure of the network will prevent the existence of cycles. If the bundle represented by the node b is a member of $\mathcal{B}$, then there is an arc bt, leaving b and entering the *terminus node* of the network (type (d) in Figure 2, labelled *end*).

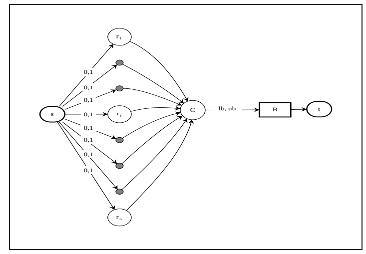

Figure 3: An exponential sized bundle system

Let us consider again the bundle system $H = (R, \mathcal{B})$, where $\mathcal{B}$ is the family of all bundles having $\lceil \frac{n}{2} \rceil$ resources ($n = |R|$). The network representing H using the above principles is given in Figure 3. If the ub and lb values on the arc cb are set to $\lceil \frac{n}{2} \rceil$, then for each $\lceil \frac{n}{2} \rceil$-subset S of R we can consider the flow f^S by putting: $f^S_{sr} = 1, f^S_{rc} = 1\ \forall s \in S$; $f^S_{sr} = 0, f^S_{rc} = 0\ \forall r \in R - S$; $f^S_{cb} = \lceil \frac{n}{2} \rceil$; and $f^S_{bt} = 1$. Clearly, the bundle represented by the node b is S. Conversely, it is not difficult to see that each non null flow f in this network generates a $\lceil \frac{n}{2} \rceil$-bundle B^f of R, by considering $B^f = \{r \in R | f(sr) = 1\}$. Note that the network has only $2|R| + 4$ nodes and $2|R| + 2$ arcs. It follows that the internal data structures have total polynomial size and also the number of variables (arc flows values) used is small.

A nice property of this type of representation is that if we are interested in an *induced subhypergraph*, that is to consider the members of $\mathcal{B}$ contained in some subset $S \subseteq R$, then it suffices to block the flow on the arcs sr for $r \in R - S$, by considering $ub_{sr} = 0$. This is clearly important for the use of v-basis as described in section 3. The restriction given by the Corollary in that section, could also be avoided in a succinct way (equivalently, considering Nisan's OR* language [8]) by adding a transit node, a new bundle node and an arc with ub set to 1 as described in Figure 4.

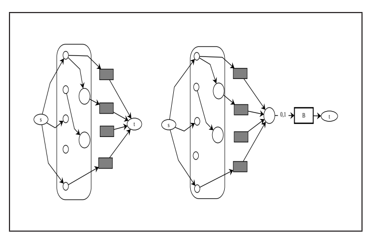

Figure 4: Implementing OR* trick.

In order to represent valuation functions using bids (i.e.v-basis) it is necessary to describe a mechanism to specify the the bid value of a bundle. This is obtained by associating *val-*

ues to the flows via special labels of network nodes. These labels indicate simple local functions which must be incrementally applied to values of the tail nodes from which the flow enter in the current node. The resulting network is called NETBID and is formally described in the section 4.

3 Semantics

This section presents that rationale that led to the contribution of the paper. The results obtained in this section build upon [4] and will lay the foundations for the semantics of the language in a rigorous, complete manner. A *Combinatorial Auction* (CA) can be interpreted as an abstraction of a marked-based centralized distributed system for the determination of *adequate* allocations of heterogenous indivisible resources. In such an *Adequate Resource Allocation* (ARA) system, there is central node a, the *auctioneer*, and a set of n nodes, $I = \{1, \ldots, n\}$, the *bidders*, which concurrently demand bundles of resources from a common set of available resources, $R = \{r_1, \ldots, r_m\}$, held by the auctioneer.

The auctioneer broadcasts R to all n bidders, asking them to submit in a specified common language, the *bidding language*, their *R-valuations* over bundles of resources. Bidder's i R-valuation, v_i, is a non-negative real function on $\mathcal{P}(R)$, expressing for each bundle $S \subseteq R$ the *individual* interest (value), $v_i(S)$, of bidder i in obtaining S. Naturally, it is assumed that $v_i(\emptyset) = 0$, and $v_i(S) \leq v_i(T)$ whenever $S \subseteq T$. No bidder i knows the valuation of any other $n-1$ bidders, but all the participants in the system agreed on an *adequate outcome*: ***(1)*** Based on bidders' R-valuations, the auctioneer will determine a resources *allocation* $\mathbf{O} = (O_1, \ldots, O_n)$, specifying for each bidder i her obtained bundle O_i. $\mathbf{O}$ is a (weak) n-partition of R, that is $O_i \cap O_j = \emptyset$, for any different bidders i and j, and $\cup_{i=1,n} O_i = R$. The global (social) value of the outcome is $v_a(\mathbf{O}) = \sum_{j=1,n} v_j(O_j)$; ***(2)*** $\mathbf{O}$ is an *adequate allocation*: if some bidder i does not receive its most wanted bundle (there is $S \subseteq R$ such that $v_i(S) > v_i(O_i)$) this is explained by the fact that the social (global) value of the outcome $v_a(\mathbf{O})$, would not increase if she will receive S: $v_a(\mathbf{O}) = \sum_{j=1,n} v_j(O_j) \leq \sum_{j=1,n} v_j(O'_j) = v_a(\mathbf{O}')$, for any allocation $\mathbf{O}' = (O'_1, \ldots, O'_n)$ having $O'_i = S$.

It is not difficult to see that there exists always such adequate allocation: let $O^* = (O^*_1, \ldots, O^*_n)$ be such that $v_a(O^*) = \max\{v_a(O) | O \text{ is a } n\text{-partition of } R\}$; if there is i and S such that $v_i(S) > v_i(O^*)$, then, by the choice of O^*, $v_a(O^*) \geq v_a(O')$ for every n-partition O' with $O'_i = S$. Conversely, if O is an adequate allocation and O^* is a maximum value allocation such that $v_a(O^*) > v_a(O)$, then (by the non-negativity property of the valuations) there is a bidder i such that $v_i(O_i) < v_i(O^*_i)$. Since O is adequate, taking $S = O^*_i$, it follows that $v_a(O) \geq v_a(O^*)$, (since O^* is a n-partition of R with $O^*_i = S$), a contradiction. We have obtained that *an allocation is adequate if and only if it is a maximum value allocation.*

The task of the auctioneer finding a maximum value allocation for a given set of bidder valuations $\{v_1, \ldots, v_n\}$, is called in the CA's field the *Winner Determination Problem* (WDP). This is a NP-hard problem, being equivalent to weighted set-packing. It tends to be solvable in many practical cases, but care is often required in formulating the problem to capture structure that is present in the domain ([9]).

WDP can be parameterized by the set R of resources, considering a fixed set I of bidders and bidders' R-valuations $\{v_i | i \in I\}$. Therefore we can write $WDP(R)$ and its corresponding maximum value $v_a(R)$. With these notations, $WDP(S)$ and $v_a(S)$ are well defined for each subset $S \subseteq R$ (by considering the restriction of v_i to $\mathcal{P}(S)$).

In this way, we have obtained a global R-valuation v_a assigning to each bundle $S \subseteq R$ the maximum value of an S-allocation to the bidders from I. By the above observation, this maximum value is the value of an adequate S-allocation. Therefore WDP can be interpreted as *the problem of constructing a social aggregation of the R-valuations of the bidders.*

If we denote by $\mathcal{V}(R)$ the set of all R-valuations, it is natural to consider in our ARA system the set of superadditive R-valuations due to the synergies among the resources: $\mathcal{SV}(R) = \{v \in \mathcal{V}(R) | v(B_1 \cup B_2) \geq v(B_1) + v(B_2) \text{ for all } B_1, B_2 \subseteq R, B_1 \cap B_2 = \emptyset\}$.

It is not difficult to see that if all $v_i \in I$ are superadditive then v_a is superadditive. Indeed, if $B_1, B_2 \subseteq R$, $B_1 \cap B_2 = \emptyset$, then $v_a(B_1) + v_a(B_2) = \sum_{i \in I} v_i(O^1_i) + \sum_{i \in I} v_i(O^2_i)$, where $\mathbf{O}^1$ is a maximum B_1-allocation and $\mathbf{O}^2$ is a maximum B_2-allocation; since $\mathbf{O}$ with $O_i = O^1_i \cup O^2_i$ $(i \in I)$ is a $B_1 \cup B_2$-allocation and $v_i(O^1_i) + v_i(O^2_i) \leq v_i(O^1_i \cup O^2_i)$, it follows that $v_a(B_1) + v_a(B_2) \leq v_a(B_1 \cup B_2)$.

The following lemma gives an interesting characterization of superadditive bidding.

Let us denote $\mathcal{V}^{OR}(R) = \{v \in \mathcal{V}(R) | v(B) = \max_{A \subseteq B}[v(A) + v(B - A)] \text{ for all } B \subseteq R\}$. Then,

Lemma 1 $\mathcal{SV}(R) = \mathcal{V}^{OR}(R)$.

Proof. If $v \in \mathcal{SV}(R)$ then for each $B \subseteq R$ and $A \subseteq B$ we have $v(B) \geq v(A \cup (B - A)) \geq v(A) + v(B - A)$, therefore $v(B) \geq \max_{A \subseteq B}[v(A) + v(B - A)]$. Since, for $A = \emptyset$, we have $v(B) = v(\emptyset) + v(B)$, it follows that $v(B) = \max_{A \subseteq B}[v(A) + v(B - A)]$, that is $v \in \mathcal{V}^{OR}(R)$. Conversely, let $v \in \mathcal{V}^{OR}(R)$. If $B_1, B_2 \subseteq R$, $B_1 \cap B_2 = \emptyset$, then $v(B_1 \cup B_2) = \max_{A \subseteq B_1 \cup B_2}[v(A) + v(B_1 \cup B_2 - A)] \geq v(B_1) + v(B_1 \cup B_2 - B_1) = v(B_1) + v(B_2)$, that is $v \in \mathcal{SV}(R)$.□

Combining this remark on the superadditivity of v_a and Lemma 1 we obtain:

Theorem 1 *If in an ARA system all bidders' R-valuations are superadditive, then the aggregate R-valuation v_a satisfies $v_a(A) = \max_{B \subseteq A}[v_a(B) + v_a(A - B)]$ for all $A \subseteq R$.*

Let $v \in \mathcal{V}(R)$. A *v-basis* is any $\mathcal{B} \subseteq \mathcal{P}(R)$ such that for each $A \subseteq R$ we have $v(A) = \max_{B \in \mathcal{B}, B \subseteq A}[v(B) + v(A - B)]$. In other words, if $\mathcal{B}$ is a v-basis, then the value of $v(A)$ is uniquely determined by the values of v on the elements of the basis contained in A, for each $A \subseteq R$. The elements of a v-basis, $B \in \mathcal{B}$, are called bundles and the pairs $(B, v(B))_{B \in \mathcal{B}}$ are called bids.

Clearly, $v \in \mathcal{V}^{OR}(R)$ if and only if $\mathcal{P}(R)$ is a v-basis. On the other hand, if $\mathcal{B} \subseteq \mathcal{P}(R)$ is a v-basis and $B \subseteq R$, then $\mathcal{B} \cup \{B\}$ is a v-basis too. Therefore, $v \in \mathcal{V}(R)$ has a v-basis iff $\mathcal{P}(R)$ is a v-basis. Using Lemma 1, we obtain the well known result (Nisan, [8]):

Corollary *A R-valuation $v \in \mathcal{V}(R)$ has a v-basis iff $v \in \mathcal{SV}(R)$.*

Let now consider an ARA system in which all bidders' R-valuations are superadditive. Each bidder $i \in I$ sends to the auctioneer its v_i-basis $\mathcal{B}_i$. The aggregate R-valuation v_a can be represented by a v_a-basis $\mathcal{B}_a$, which is obtained by merging the individual basis $\mathcal{B}_i$ in a very simply way: $\mathcal{B}_a = \cup_{i \in I} \mathcal{B}_i$ and if $B \in \mathcal{B}_a$ then $v_a(B) = \max\{v_i(B) | i \in I$ and $B \in \mathcal{B}_i\}$.

Indeed, by theorem 1, we have $v_a(A) = \max_{B \subseteq A}[v_a(B) + v_a(A - B)]$, for all $A \subseteq R$. If $\mathbf{O} = (O_1, \ldots, O_n)$ is a maximum A-allocation, then $v_a(A) = \sum_{i \in I} v_i(O_i)$. If $v_i(O_i) > 0$, then it is not difficult to see that $v_i(O_i) = v_a(O_i) \geq v_j(O_i)$ for all $j \in I$ and $v_a(A) = v_a(O_i) + \sum_{j \in I - \{i\}} v_j(O_j) = v_a(O_i) + v_a(A - O_i)$.

Furthermore, since $\mathcal{B}_i$ is a v_i-basis there is $B_i \in \mathcal{B}_i$ such that $v_i(O_i) = v_i(B_i) + v_i(O_i - B_i)$, $v_i(O_i) = v_a(B_i)$ and, moreover, $v_a(A) = v_a(B_i) + v_a(A - B_i)$.

We obtained the following interesting representational theorem:

Theorem 2 *If in an ARA system the bidder superadditive R-valuations v_i are represented using v_i-basis $\mathcal{B}_i$ for each $i \in I$, then the aggregate R-valuation v_a is represented by the v_a-basis $\mathcal{B}_a = \cup_{i \in I} \mathcal{B}_i$, by taking $v_a(B) = \max\{v_i(B) | i \in I$ and $B \in \mathcal{B}_i\}$, for all $B \in \mathcal{B}_a$.*

We note here that Lemma 1 can be extended to obtain a similar characterization of a subclass of additive R-valuations.

Let us consider $\mathcal{SUPV}(R)$, the set of all *supermodular R-valuations*, that is $\mathcal{SUPV}(R) = \{v \in \mathcal{V}(R) | \, v(B_1 \cup B_2) \geq v(B_1) + v(B_2) - v(B_1 \cap B_2)$ for all $B_1, B_2 \subseteq R\}$.

Clearly, $\mathcal{SUPV}(R) \subset \mathcal{SV}(R)$.

Also, we restrict the set of OR-valuations, by considering *strongly OR-valuations* (sOR-valuations): $\mathcal{V}^{sOR}(R) = \{v \in \mathcal{V}(r) | \, v(B) = \max_{A_1, A_2 \subseteq B}[v(A_1) + v(A_2) - v(A_1 \cap A_2)]$ for all $B \subseteq R\}$.

Lemma 2 $\mathcal{SUPV}(R) = \mathcal{V}^{sOR}(R)$.

Proof. If $v \in \mathcal{SUPV}(R)$, then for each $B \subseteq R$ and $A_1, A_2 \subseteq B$ we have $v(B) \geq v(A_1 \cup A_2) \geq v(A_1) + v(A_2) - v(A_1 \cap A_2)$, therefore $v(B) \geq \max_{A_1, A_2 \subseteq B}[v(A_1) + v(A_2) - v(A_1 \cap A_2]$. Since, for $A_1 = B$ and $A_2 = \emptyset$, we have $v(B) = v(B) + v(\emptyset) - v(B \cap \emptyset)$, it follows that $v(B) = \max_{A_1, A_2 \subseteq B}[v(A_1) + v(A_2) - v(A_1 \cap A_2]$, that is $v \in \mathcal{V}^{sOR}(R)$. Conversely, let $v \in \mathcal{V}^{sOR}(R)$. If $B_1, B_2 \subseteq R$, then $v(B_1 \cup B_2) = \max_{A_1, A_2 \subseteq B_1 \cup B_2}[v(A_1) + v(A_2) - v(A_1 \cap A_2)] \geq v(B_1) + v(B_2) - v(B_1 \cap B_2)$, that is, $v \in \mathcal{SUPV}(R)$.□

As above, we have

Theorem 3 *If in an ARA system all bidders' R-valuations are supermodular, then the aggregate R-valuation v_a satisfies $v_a(B) = \max_{A_1, A_2 \subseteq B}[v(A_1) + v(A_2) - v(A_1 \cap A_2)]$ for all $B \subseteq R$.*

4 Syntax

The proposed language is based on the following two novel ideas: (1) The use of generalized network flows to represent the bids; and (2) The interpretation of the WD as an adequate aggregation of individual preferences.

In the new language, each bidder submits to the arbitrator a generalized flow network representing its bids. We call such a network flow NETBID and it will represent the valuation of the bidder. More precisely, if the set of resources is $R = \{r_1, r_2, \ldots, r_m\}$, then in the NETBID of each agent there is a special node $START$ connected to m nodes r_j by directed edges having capacity 1. An integer flow in NETBID will represent an assignment of resources to the agent by considering the set of resources r_j with flow value 1 on the directed edge $(START, r_j)$. The node r_j is an usual node, that is, it satisfies the conservation law: the total (sum) of incoming flows equals the total flow of outcoming flows. In the network there are also *bundle* nodes which do not satisfy the conservation law, which are used to combine (via their inputs flows) different goods in subset of goods. The combination is conducted by the (integer) directed edges flows together with appropriate lower and capacity bounds. For example, the additive valuation, $v(S) = |S|$ for each subset S of R can be represented by the NETBID in Figure 5.

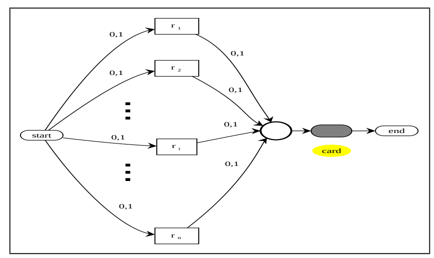

Figure 5: Additive valuation network

There is an important improvement over other existing graph-oriented bidding languages namely the possibility that a bundle node to represent an entire hypergraph having as vertices the resource set R. Furthermore, the nodes values are given by using labels on bundles nodes, which are positive real numbers or even procedural functions having as arguments the values of the incoming flows. This has as consequence a higher expressiveness of the bidding language.

Once the NETBID has been constructed, any maximum value flow (in the sense described above) will represent the valuation function of the agent. In particular cases it is not difficult for a rational bidder to construct a NETBID representing his preferences. For example, the NETBID in Figure 6 expresses that the bidder is interested in a bundle consisting in two or three resources of type E , together with the resource M which adds 10 to the values sum of the particular resources of type E:

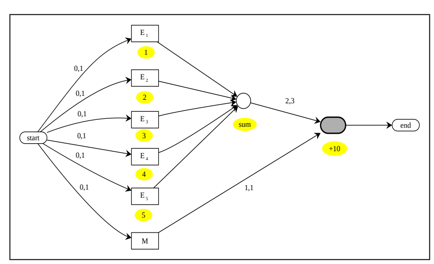

Figure 6: Tbbl valuation network

An important extension of our flows is that if the flow is null on some particular arc then it is not necessary that the lower bounds and capacity constraint to be verified.

Formally a NETBID can be defined as follows.

Definition 1 *A R-**NETBID** is a tuple* $\mathcal{N} = (D, \textsc{start}, \textsc{end}, c, l, \lambda)$*:*

1. *$D = (V, E)$ is an acyclic digraph with two distinguished nodes $\textsc{start}, \textsc{end} \in V$; the other nodes, $V - \{\textsc{start}, \textsc{end}\}$, are partitioned $R \cup B \cup I$: R is the set of resources nodes, B is the set of bundles nodes and I is the set of interior nodes. There is a directed edge $(\textsc{start}, r) \in E$ for each $r \in R$, and at least a directed edge $(b, v) \in E$ for each $b \in B$. There are no other directed edges entering in a resource node. The remaining directed edges connect resources nodes to bundle or interior nodes, interior nodes to bundle or interior nodes, bundle nodes to interior nodes or $\textsc{end}$ node.*
2. *c, l are nonnegative integer partial functions defined on the set of directed edges of D; if $(i, j) \in E$ and c is defined on (i, j) then $c((i, j)) \in \mathbf{Z}_+$, denoted c_{ij}, is the capacity of directed edge (i, j); $l((i, j)) \in \mathbf{Z}_+$, if defined, is the lower bound on the directed edge (i, j) and is denoted l_{ij}; if (i, j) has assigned a capacity and a lower bound then $l_{ij} \leq c_{ij}$. All directed edges $(\textsc{start}, r)$ have capacity 1 and the lower bound 0. No directed edge $(b, \textsc{end})$ has capacity and lower bound.*
3. *λ is a labelling function on $V - \{\textsc{start}, \textsc{end}\}$ which assign to a vertex v a pair of rules $(\lambda_1(v), \lambda_2(v))$ (which will be described in the next definitions).*

Definition 2 *Let $\mathcal{N} = (D, \textsc{start}, \textsc{end}, c, l, \lambda)$ be a R-NETBID. A **bidflow** in $\mathcal{N}$ is a function $f : E(D) \rightarrow \mathbf{Z}_+$ satisfying the following properties (f_{ij} denotes $f((i, j))$):*

1. *For each directed edge $(i, j) \in E$: if $f_{ij} > 0$ and c_{ij} is defined, then $f_{ij} \leq c_{ij}$; if $f_{ij} > 0$ and l_{ij} is defined, then $f_{ij} \geq l_{ij}$.*
2. *If $v \in V - \{\textsc{start}, \textsc{end}\}$ has $\lambda_1(v) = conservation$ then $\sum_{(i,v) \in E(D)} f_{iv} = \sum_{(v,i) \in E(D)} f_{vi}$.*
3. *For each $v \in B$, $f_{vu} \in \{0, 1\}$; there is exactly one vertex u such that $f_{vu} = 1$ and this happens if and only if for each $w \in R \cup I$, such that $(w, v) \in E(D)$, we have $f_{wv} > 0$.*

The set of all bidflows in $\mathcal{N}$ is denoted by $\mathcal{F}^{\mathcal{N}}$.

In order to simplify our presentation we have considered here that for each $v \in V - \{START, END\}$, $\lambda_1(v) \in \{conservation, bundle\}$ giving rise to the flow rules described above. In all the figures considered here, the function $\lambda_1(v)$ is illustrated by the color of the node v: a gray node is a bundle node and a white node is a conservation node. It is possible to use the $\lambda_1(v)$ to have *transformation* internal nodes as [3].

Definition 3 *Let f be a bidflow in the R-NETBID $\mathcal{N} = (D, \textsc{start}, \textsc{end}, c, l, \lambda)$. The **value of** f, $val(f)$, is defined as $val(f) = \sum_{b \in B} val(b) f_{bEND}$, where $val(v)$ is*

$$val(v) = \begin{cases} 0 & \text{if } v = \textsc{start} \\ \lambda_2(D_f^{-1}(v)) & \text{if } v \neq \textsc{start}, \textsc{end}. \end{cases}$$

$D_f^{-1}(v)$ denotes the set of all vertices $w \in V(D)$ such that $(w, v) \in E(D)$ and $f_{wv} > 0$. $\lambda_2(D_f^{-1}(v))$ is the rule (specified by the second label associated to vertex v) of computing $val(v)$ from the values of its predecessors which send flows into v.

Definition 4 *Let $\mathcal{N} = (D, \textsc{start}, \textsc{end}, c, l, \lambda)$ be a R-NETBID. The R-**valuation** designated by $\mathcal{N}$ is the function $v_{\mathcal{N}} : \mathcal{P}(R) \rightarrow \mathbf{R}_+$, where for each $S \subseteq R$,*
$v_{\mathcal{N}}(S) = \max\{val(f) | f \in \mathcal{F}^{\mathcal{N}}, f_{START\, r} = 0 \forall r \in R - S\}$.

By the above two definitions, the value associated by $\mathcal{N}$ to a set S of resources is the maximum sum of the values of the (disjoint) bundles which are contained in the set (assignment) S. This is in concordance with the definition of a v-basis given in section 3 for a superadditive valuation v. However, the NETBID structure defined above is more flexible in order to express any valuation. If the bidder desires to express that at most k bundles from some set of bundle nodes must be considered, then these nodes are connected to a new interior node and this last node linked to a new superbundle node by a directed edge having as lower bound 1 and capacity k. Clearly, any valuation represented in a XOR language can be obtained in such way and any R-valuation can be represented.

The NETBIDS submitted by the bidders are merged by the arbitrator in a common NETBID sharing only the nodes corresponding to $START$ and R, and also a common END node in which are projected the corresponding END nodes of the individual NETBIDS. This common NETBID is a symbolic representation of the aggregate valuation of the society. We consider the following definition.

Definition 5 *Let $\mathcal{N}_i = (D_i, \textsc{start}_i, \textsc{end}_i, c_i, l_i, \lambda_i)$ be the R-NETBID of the agent $i \in I$. The aggregation R-NETBID of $\{\mathcal{N}_i | i \in I\}$ is the R-NETBID $\mathcal{N}_a = (D_a, \textsc{start}_a, \textsc{end}_a, c_a, l_a, \lambda_a)$, where $D_a = (V_a, E_a)$ has*
$V_a = \{\textsc{start}_a, \textsc{end}_a\} \cup R \cup B_a \cup I_a$, B_a (respectively I_a) being the disjoint union of all individual bundle node sets B_i (respectively, internal nodes I_i);
$E_a = \{\textsc{start}_a\} \times R \cup B_a \times \{\textsc{end}_a\} \cup \cup_{i \in I} (E_i - (\{\textsc{start}_i\} \times R \cup B_i \times \{\textsc{end}_i\}))$.
All directed edges $(\textsc{start}_a, r)$ have the capacity $c_a = 1$ and the lower bound $l_a = 0$. No directed edge $(b, \textsc{end}_a)$ has capacity and lower bound. All the remainder capacities and lower bounds are obtained from the corresponding values in the individual NETBIDS. Similarly are constructed the label rules $\lambda_a(v)$. If a resource node r_a has different λ_2 values in some local networks, then r_a is connected to new copies of it by directed edges with $c_a = 1$ and $l_a = 0$, and this new nodes are connected by directed edges corresponding to the local NETBID.

This definition is illustrated in Figure 7.

From this construction, the following theorem can be proved

Theorem 4 *If in a FRA system each bidder's i R-valuation, v_i, is represented by R-NETBID $\mathcal{N}_i$ ($i \in I$), then the aggregate R-valuation v_a is designated by the aggregate R-NETBID $\mathcal{N}_a$, that is, $v_a = v_{\mathcal{N}_a}$.*

Proof. Let $S \subseteq R$. If If $\mathbf{O} = (O_1, \ldots, O_n)$ is a maximum S-allocation, that is, $v_a(S) = \sum_{i \in I} v_i(O_i)$, then for each $i \in \{1, \ldots, n\} = I$, $v_i(O_i) = v_{\mathcal{N}_i}(O_i)$ can be obtained as $val(f_i)$,

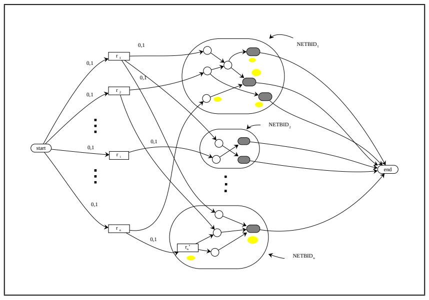

Figure 7: Aggregate NETBID

for a maximum value bidflow f_i in $\mathcal{N}_i$. The flows $(f_i)_{i \in I}$ induce a flow f_a in the NETBID $\mathcal{N}_a$, with $val(f_a) = \sum_{i \in I} v_i(O_i)$, that is $val(f_a) = v_a(S)$. Conversely, each maximum value bidflow f_a in $\mathcal{N}_a$ can be decomposed into disjoint bidflows f_i in the NETBID $\mathcal{N}_i$, which must be maximum value bidflows. It follows that $val(f_a) = \sum_{i \in I} val(f_i) = \sum_{i \in I} v_{\mathcal{N}_i}(S) = v_a(S)$.□

The maximum value of a bidflow in the NETBID $\mathcal{N}_a$ is the social welfare value: the computation of this value implicitly solves the WD problem (by a simple bookkeeping of agents owning the winning bundles in the aggregate NETBID).

5 Discussion

Several bidding languages for CAs have previously been proposed, arguably the most compelling of which allow bidders to explicitly represent the logical structure of their valuation over goods via standard logical operators. These are referred as "logical bidding languages" (e.g. [8]). For instance, an OR bid specifies a set of $< bundle, price >$ pairs, where the bidder is willing to buy any number of the specified bundles for their respectively specified prices. This is equivalent to specifying a set of single-bundle bids. An XOR bid specifies a set of $< bundle, price >$ pairs, where the bidder is willing to pay for only one of the bundles for its corresponding price. Nisan's OR* language [8] provides constraints within an OR bid via "phantom variables" (see also [6]). One explanation of restricting operators to just OR and XOR in the logical framework adopted by these languages, is given by the characteristics of the accompanying WD-solving methodology the language designers proposed. Boutilier and Holger [1] made the next logical step with the LGB language, which allows for arbitrarily nested levels combining goods and bundles by the standard propositional logic operators: OR, XOR, and AND. Day [5] introduces bid tables and bid matrices as a bidding language more connected to the economic literature on restricted preferences and assignment games. Cavallo and colleagues [2], introduce TBBL, a tree-based bidding language that has several novel properties. In TBBL, valuations are expressed in a tree structure, where internal nodes in the tree correspond to operators for combining subsets of goods, and individual goods are represented at the leaves. TBBL allows agents to express preferences for both buying and selling goods in the same tree. Thus, it is applicable to a combinatorial exchange (CE), a generalization of a CA that is important in many multiagent systems. TBBL also provides an explicit semantics for partial value information: a bidder can specify an upper and lower bound on their true valuation, to be refined during bidding. TBBL is a logical tree-based bidding language for CEs. It is fully expressive, yet designed to be as concise and structured as possible. Finally, Cerquides and colleagues [3], explicitly addresses the case of bidding languages for CEs by extending the classical $< bundle, price >$ view of a bid to a $< transformation, price >$ pair. This work is extended by Giovanucci and colleagues [7], which provide an interesting Petri Nets formalism to reason about these CAs extensions.

In this paper we proposed a new visual framework for bidding languages. We have motivated our approach by analyzing adequate resource allocation systems (semantics) and introduced our work in a theoretical manner. We also presented a number of intuitive examples with the purpose of highlighting the advantages of our work. We believe that when bidders are able to express a wide variety of preferences to a sealed-bid or proxy agent, NETBID flows allow to iteratively generate an economically satisfactory market outcome. Moreover, the format for the representation of bidder preferences serves to reinforce the global perspective on the implementation of combinatorial auctions using a new computational technique (CSP based) for determining auction outcomes. We are currently pursuing this line of work for practical evaluation.

References

[1] C. Boutilier and H. Holger. Bidding languages for combinatorial auctions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI).*, pages 1211–1217, 2001.

[2] R. Cavallo, D. Parkes, A. Juda, A. Kirsch, A. Kulesza, S. Lahaie, B. Lubin, L. Michael, and J. Shneidman. Tbbl: A tree-based bidding language for iterative combinatorial exchanges. In *International Joint Conferences on A.I.: Workshop on Advances in Preference Handling*, 2005.

[3] J. Cerquides, U. Endriss, A. Giovannucci, and J. Rodriguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *International Joint Conferences on A.I.*, pages 1221–1226, 2007.

[4] M. Croitoru, P. Lewis, and C. Croitoru. Bidflow: a new graph-based bidding language for combinatorial auctions. In *Proceedings of the 18th European Conference on Artificial Intelligence*, 2008. to appear.

[5] R. Day. *Expressing Preferences with Price-Vector Agents in Combinatorial Auctions*. PhD thesis, University of Maryland, College Park., 2004.

[6] Y. Fujisima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, 1999.

[7] A. Giovannucci, J. Rodriguez-Aguilar, J. Cerquides, and U. Endriss. Winner determination for mixed multi-unit combinatorial auctions via petri nets. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007.

[8] N. Nisan. Bidding and allocations in combinatorial auctions. In *ACM Conference on Electronic Commerce (EC-2000)*, 2000.

[9] M. Rothkopf, A. Pekec, and R. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44:1131–1147, 1998.