Combinatorial Auctions with k-wise Dependent Valuations*

Vincent Conitzer
Carnegie Mellon University
Computer Science Department
5000 Forbes Ave.
Pittsburgh, PA, 15213
conitzer@cs.cmu.edu

Tuomas Sandholm
Carnegie Mellon University
Computer Science Department
5000 Forbes Ave.
Pittsburgh, PA, 15213
sandholm@cs.cmu.edu

Paolo Santi
Istituto di Informatica e Telematica
Pisa, 56124, Italy
paolo.santi@iit.cnr.it

Abstract

We analyze the computational and communication complexity of combinatorial auctions from a new perspective: the degree of interdependency between the items for sale in the bidders' preferences. Denoting by $\mathbf{G_k}$ the class of valuations displaying up to k-wise dependencies, we consider the hierarchy $\mathbf{G_1} \subset \mathbf{G_2} \subset \cdots \subset \mathbf{G_m}$, where m is the number of items for sale. We show that the minimum non-trivial degree of interdependency (2-wise dependency) is sufficient to render NP-hard the problem of computing the optimal allocation (but we also exhibit a restricted class of such valuations for which computing the optimal allocation is easy). On the other hand, bidders' preferences can be communicated efficiently (i.e., exchanging a polynomial amount of information) as long as the interdependencies between items are limited to sets of cardinality up to k, where k is an arbitrary constant. The amount of communication required to transmit the bidders' preferences becomes super-polynomial (under the assumption that only value queries are allowed) when interdependencies occur between sets of cardinality $g(m)$, where $g(m)$ is an arbitrary function such that $g(m) \to \infty$ as $m \to \infty$. We also consider *approximate* elicitation, in which the auctioneer learns, asking polynomially many value queries, an approximation of the bidders' actual preferences.

Introduction

Combinatorial auctions (CAs) have emerged as an important mechanism for resource and task allocation in multiagent systems. CAs have been used to trade transportation services, pollution permits, land lots, spectrum licenses, and so on. In a CA, bidders can express *complementarities* (i.e., the value of a package of items being worth more than the sum of the values of the individual items in the package), and *substitutabilities* (i.e., the value of a package of items being worth less than the sum of the values of the individual items in the package). (Complementarities are also referred to as *super-additivity*, and substitutabilities as *sub-additivity*.) The function that, given a package of items, returns the bidder's value for that package, is called the *valuation function*, or simply *valuation*.

The implementation of CAs poses several challenges, including computing the optimal allocation of the items given the valuation functions of the agents (aka. the *winner determination* problem), and eliciting enough information about the bidders' valuation functions to determine a good allocation (aka. the *preference elicitation* problem).

Of these two problems, the interests of computer scientists were first focused on the winner determination problem. It is NP-hard (Rothkopf, Pekeč, & Harstad 1998) and even finding an approximation is NP-hard (Sandholm 2002). However, modern search algorithms can often solve the structured winner determination problems that arise in typical practical domains to optimality, even in the large (Sandholm 2006).

Therefore, the preference elicitation problem is in fact the bigger bottleneck in obtaining the economic efficiency that CAs can offer in principle. It is not feasible to have every bidder submit a valuation for every one of the exponentially many packages. Requiring all of this information is undesirable for several reasons. First, determining one's valuation for any specific bundle can be computationally demanding (Sandholm 1993; 2000; Parkes 1999b; Larson & Sandholm 2001)—thus requiring this computation for exponentially many packages is impractical. Second, communicating exponentially many bids can be prohibitive (e.g., wrt. network traffic). Finally, agents may prefer not to reveal their valuation information for reasons of privacy or long-term competitiveness (Rothkopf, Teisberg, & Kahn 1990).

Early approaches to addressing the preference elicitation problem (although it was not called that then) involved designing different ascending combinatorial auctions (e.g., (Parkes 1999a; Wurman & Wellman 2000; Ausubel & Milgrom 2002; de Vries, Schummer, & Vohra 2003)). More recently, the general preference elicitation framework for CAs was introduced, where the auctioneer is enhanced by elicitor software that incrementally elicits the bidders' preferences using queries until enough information has been elicited to determine the right allocation of items to bidders (Conen & Sandholm 2001).[1] Several elicitation algorithms, based

*This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship. This work was done while the third author was visiting CMU.

[1] If enough information is elicited to determine the optimal al-

on different classes of queries (e.g., value, rank, and order queries), have been proposed (Conen & Sandholm 2001; 2002; Hudson & Sandholm 2004). Ascending auctions are a special case of the framework, where the queries are demand queries, and the prices are restricted to being increasing.

Unfortunately, a recent result (Nisan & Segal 2005) shows that elicitation algorithms have no hope of considerably reducing the communication complexity in the worst case. In fact, obtaining a better approximation than that generated by auctioning off all objects as a bundle requires the exchange of an exponential amount of information. Thus, the communication burden produced by *any* combinatorial auction design that aims at producing a non-trivial approximation of the optimal allocation is overwhelming, unless the bidders' valuation functions display some structure. Of course, in practice, such structure is very likely to be present, because otherwise bidders' cognitive limitations would presumably prevent them from producing a separate value for each of the exponentially many bundles.

Given the winner determination and preference elicitation hardness results, several authors have presented restricted CA settings, in which solving either the winner determination problem, or the preference elicitation problem, or both, are easy even in the worst case (LaMura 1999; Zinkevich, Blum, & Sandholm 2003; Blum *et al.* 2004; Lahaie & Parkes 2004; Santi, Conitzer, & Sandholm 2004; Chang, Li, & Smith 2003; Rothkopf, Pekeč, & Harstad 1998; Lehmann, O'Callaghan, & Shoham 2002; Nisan 2000; Tennenholtz 2000; Sandholm 2002; Sandholm *et al.* 2002; Sandholm & Suri 2003; Conitzer, Derryberry, & Sandholm 2004). The challenge is to identify classes of valuations that are sufficiently general (in the sense that they allow the bidders to express super-, or subadditivity, or both, among items) and realistic, yet easy to solve.

In this paper, we analyze the complexity of winner determination and preference elicitation in CAs from a new perspective: the degree of mutual interdependency between the items. In general, a set of items displays some form of interdependency when their value as a bundle is different from the sum of their values as single items, resulting in complementarity or substitutability between the objects. It can be argued that this is the distinguishing feature of CAs.

The degree of mutual interdependency between objects is clearly related to the computational and communication efficiency of CAs. When there is no interdependency, then the bidders' preferences are linear (i.e., the valuation of a bundle is the sum of the values of the items it contains); in this situation, computing the optimal allocation is straightforward, and communication complexity is not an issue. However, this case does not require a CA, since the items could be auctioned sequentially with the same economic efficiency. On the other hand, when the degree of mutual interdependency is maximal (i.e., up to m-wise dependencies are exhibited in the bidders' valuations, where m is the number of items for sale), we have fully general valuations, and both winner determination and preference elicitation are hard.

location, and what the optimal allocation would have been with each bidder's bids removed in turn, then answering the elicitor's queries truthfully can be made an ex post equilibrium strategy using the Vickrey-Clarke-Groves mechanism, as proposed in (Conen & Sandholm 2001).

In this paper, we study the case where the degree of interdependency between items is somewhere between 1 and m (of course, some of the items in the CA may exhibit lower degree of interdependency). We believe that this type of valuation is likely to arise in many economic scenarios. For instance, when the items for sale are related to a geometric or geographic property (e.g., spectrum frequencies, railroad tracks, land slots,...), it is reasonable to assume that only items that are geometrically/geographically close display some form of interdependency. Another consideration that motivates our interest in k-wise dependent valuations is that, due to cognitive limitations, it might be difficult for a bidder to understand the inter-relationships between a large group of items.

Another paper that independently introduces essentially the same model appeared in October 2004 in a DIMACS workshop (Chevaleyre *et al.* 2004)—a few weeks after we had presented the work in this paper at another DIMACS workshop (the DIMACS Workshop on Computational Issues in Auction Design, Rutgers University, New Jersey). In any case, the results do not overlap, except for the NP-completeness result that we give later, which is stronger than one presented in the other paper (Chevaleyre *et al.* 2004) since we prove hardness in a more restricted setting.

When considering communication complexity, we will focus our attention on a restricted case of preference elicitation, in which the elicitor can ask only *value queries* (what is the value of a particular bundle?) to the bidders. Our interest in value queries is due to the fact that, from the bidders' point of view, these queries are very intuitive and easy to understand. They are also, together with demand queries, the most commonly studied query class in the CA literature.

2-wise dependent valuations

Let I denote the set of items for sale (also called the *grand bundle*), with $|I| = m$. A *valuation function* on I (*valuation* for short) is a function $v : 2^I \mapsto \mathbb{R}^+$ that assigns to any bundle $S \subseteq I$ its valuation. To make the notation less cumbersome, in this paper we will use notation $a, b, \ldots$ to denote singletons, $ab, bc, \ldots$ to denote two-item bundles, and so on.

In the following, we will focus on the valuation function of an arbitrary bidder A. Let us consider an arbitrary pair a, b of items in I.

We have:

- if $v(ab) = v(a) + v(b)$ then a and b are *independent*;
- if $v(ab) > v(a) + v(b)$ then a and b are *super-additive*;
- if $v(ab) < v(a) + v(b)$ then a and b are *sub-additive*.

Given the dependencies between any pair of items in I, let the *2-wise dependency graph* G_2 be constructed as follows:

- let there be a node for every item in I;
- label node a with $v(a)$;[2]

[2]Slightly abusing the notation, we use a to denote both the item and the corresponding node in the graph.

– if a and b are super- or sub-additive, put an (undirected) edge (a, b) in the graph, and label the edge with $v(ab) - (v(a) + v(b))$.

Under the assumption that there exist only 2-wise item dependencies in the valuation function of bidder A, the G_2 graph can be used to calculate the valuation of any possible subset S of I as follows: consider the subgraph G^S of G_2 induced by node set S; sum up all the node and edge labels in G^S. Formally, the class of *2-wise dependent valuations* is exactly the class of valuations for which this computation produces the correct valuation of any subset S, *i.e.* the class of valuations that can be accurately represented by their G_2 graphs.

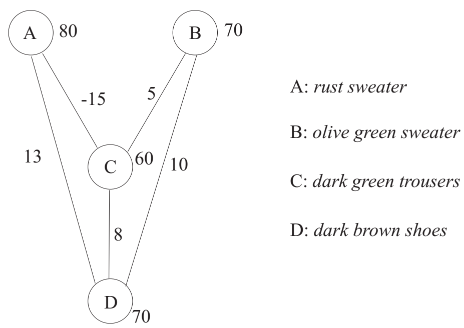


Figure 1: 2-wise dependency graph representing the bidder's valuation in the auction of fashion clothing.

An example of a 2-wise dependent valuation could be the following. Consider an auction of fashion clothing. In this scenario, it seems reasonable to assume that items display super- of sub-additivity depending on how good they look together. In Figure 1, there are four items for sale: a rust sweater, an olive green sweater, dark green trousers, and a pair of dark brown shoes. The items have values as singletons (e.g., the rust sweater is worth \$80 to the bidder), and show 2-wise dependencies when bundled together. For instance, the bundle composed of the olive green sweater, dark green trousers and dark brown shoes has a super-additive valuation (\$223 instead of \$200), because these items together form a nice outfit. Conversely, the rust sweater and the dark green trousers clash, so their value as a bundle is sub-additive (\$125 instead of \$140).

Note that the setting at hand (2-wise dependent valuations) *is not* equivalent to allowing only bids on bundles composed of at most two items: what we are bounding here is the degree of interdependency between item valuations when several items are bundled together, and not the cardinality of the bundle. Indeed, the class of 2-wise dependent valuations is significantly different from those corresponding to existing bidding languages. For example, even the simple example given above cannot be expressed using ORs and XORs over bundles of two items.

The class of 2-wise dependent valuations is efficient from the communication complexity point of view. In fact, it is easy to see that any valuation in this class can be elicited by asking only $\frac{m(m+1)}{2}$ value queries (m single item queries, and $\frac{m(m-1)}{2}$ queries for the two item bundles). The price that must be paid for this is that not all possible preferences can be expressed using G_2 graphs.

An interesting comparison can be made between the expressive power of 2-wise dependent valuations and that of other classes of valuations, such as those presented in (Zinkevich, Blum, & Sandholm 2003), which can also be elicited asking a polynomial number of value queries. We omit this comparison due to space constraint.

Remark. *Costly disposal* can be easily expressed using 2-wise dependencies graphs. Costly disposal models those situations in which the bidder incurs a cost for disposing of undesired items. Thus, the monotonicity assumption typical of the *free disposal* setting, i.e. that $v(S') \geq v(S)$ for any $S' \supseteq S$, need not hold. For instance, the fact that the bidder values a at 2 and b at 5, wants at most one of the items, and incurs a cost of 1 for disposing of an extra item, can be represented using the G_2 graph which assigns weight 2 to node a, 5 to node b, and weight -3 to the edge (a, b). To the best of our knowledge, 2-wise dependent valuations are the only known class of valuation functions that can express costly disposal and can be elicited asking a polynomial number of queries. In fact, the classes of easy to elicit valuations defined in (Blum *et al.* 2004; Nisan & Segal 2005; Zinkevich, Blum, & Sandholm 2003), as well as the preference elicitation techniques proposed in (Hudson & Sandholm 2004) and referred therein, are based on the free disposal assumption.

Learning *almost* 2-wise dependent valuations

Let $\mathbf{G_2}$ denote the class of valuation functions that can be expressed using a G_2 graph. In this section, we consider the case in which the valuation function v does not belong to $\mathbf{G_2}$, but it can be well approximated by some $v' \in \mathbf{G_2}$. (Care must be taken in using these approximations of valuations: for example, using approximations of the bidders' preferences may break the incentive compatibility of the VCG (Vickrey 1961; Clarke 1971; Groves 1973) mechanism. Of course, most real-world combinatorial auctions do not actually use the VCG mechanism due to problems from which it suffers (Ausubel & Milgrom 2006; Conitzer & Sandholm 2004). Moreover, if the costs of assessing one's valuations are taken into account, a recent result shows that *no* mechanism (not even VCG) is incentive compatible (Larson & Sandholm 2005).)

Bidder preferences can be represented using the *hypercube representation,* which is defined as follows. Given the set I of items for sale, we build the undirected graph H_I introducing a node for any subset of I (including the empty set), and an edge between any two nodes S_1, S_2 such that $S_1 \subset S_2$ and $|S_1| = |S_2| - 1$. It is immediate that H_I is a binary hypercube of dimension m. Nodes in H_I can be partitioned into levels according to the cardinality of the corresponding subset: level 0 contains the empty set, level 1 the m singletons, and so on.

The valuation function v can be represented using H_I by assigning a weight to each node of H_I as follows. We assign weight 0 to the empty set, and weight $v(a)$ to any singleton

a. Let us now consider a node at level 2, say node ab.[3] The weight of the node is $v(ab) - (v(a) + v(b))$. At the general step i, we assign to node S_1, with $|S_1| = i$, the weight $v(S_1) - \sum_{S \subset S_1} w(S)$, where $w(S)$ denotes the weight of the node corresponding to subset S. The hypercube representation of valuation v on item set I is denoted $H_I(v)$. It is easy to see that any valuation function v admits a hypercube representation, and this representation is unique.

Given the hypercube representation $H_I(v)$ of v, the valuation of any bundle S can be obtained by summing up the weights of all the nodes S' in $H_I(f)$ such that $S' \subseteq S$. These are the only weights contained in the sub-hypercube of $H_I(v)$ "rooted" at S. We denote this sub-hypercube with $H_I^S(v)$.

We will use the concept of the distance of a valuation function from a class, defined as follows.

Definition 1 *Let v be an arbitrary valuation function, and $\mathbf{C}$ be an arbitrary class of valuation functions. Given a function $v' \in \mathbf{C}$, we say that v' is a δ-approximation of v if $|v(S) - v'(S)| \leq \delta$ for every bundle S. The distance between v and $\mathbf{C}$, denoted $d(v, \mathbf{C})$, is defined as $\min\{\delta | \exists v' \in \mathbf{C}$ such that v' is a δ-approximation of $v\}$.*

If the valuation function v to be elicited is a δ-approximation of a 2-wise dependency function v', then the following theorem shows that a $O(m^2\delta)$-approximation of v can be learned asking $\frac{m(m+1)}{2}$ value queries.

Theorem 1 *Assume that the valuation function v is a δ-approximation of v', for some $v' \in \mathbf{G_2}$. Then, a function $g \in \mathbf{G_2}$ can be learned asking the $\frac{m(m+1)}{2}$ value queries on bundles of size 1 and 2, such that for any bundle of items S,*

$$|v(S) - g(S)| \leq \delta \left(1 + \frac{|S|(|S|-1)}{2}\right) .$$

Proof: Due to space limitations, we omit the proofs of most of the results in this paper. ∎

Theorem 2 *The bound stated in Theorem 1 is tight for the elicitation technique used in the proof. (For any value of m, there exist valuation functions v, v', with $v' \in \mathbf{G_2}$ such that v' is a δ-approximation of v, and the function g learned in polynomial time is a $\delta\left(1 + \frac{m(m-1)}{2}\right)$-approximation of v.)*

Next, let us consider valuations such that all the weights in the corresponding H_I graph are non-negative. We call these valuations *strongly super-modular* valuations.[4] It is not hard to see that strongly super-modular valuations are hard to elicit with value queries, because they require exponentially many values to specify. The following theorem gives an upper bound on the distance between any strongly super-modular valuation and the $\mathbf{G_2}$ class, which contains easy to elicit valuations.

Theorem 3 *Let v be an arbitrary strongly super-modular valuation, and let v_2 be the unique valuation function in $\mathbf{G_2}$ that coincides with v on the singletons and two-item bundles. Let $c_i(v) = \max_{S,|S|=i}\{v(S) - v_2(S)\}$, and let $M(v) = \max_{i=3,\ldots,m} \frac{2c_i(v)}{i(i+1)}$. Then, there exists a function $v' \in \mathbf{G_2}$ such that $|v(S) - v'(S)| \leq \frac{M(v)}{2} \cdot \frac{|S|(|S|+1)}{2}$ for any bundle S. Thus, we have $d(v, \mathbf{G_2}) \leq \frac{M(v)}{2} \cdot \frac{m(m+1)}{2}$.*

The following theorem shows that the bound stated in Theorem 3 is tight.

Theorem 4 *There exist strongly super-modular valuations v such that $d(v, \mathbf{G_2}) = \frac{M(v)}{2} \cdot \frac{m(m+1)}{2}$.*

The results stated in theorems 1 and 3 can be combined into the following theorem, which gives an upper bound on the error that we have when an arbitrary strongly super-modular valuation v is approximated using a function in $\mathbf{G_2}$ using polynomially many queries.

Theorem 5 *Let v be an arbitrary strongly super-modular valuation. Then, a function $g \in \mathbf{G_2}$ can be learned asking $\frac{m(m+1)}{2}$ value queries such that $|v(S) - g(S)|$ is at most:*

$$\frac{M(v)}{2}\left(\frac{|S|(|S|+1)}{2}\right)\left(1 + \frac{|S|(|S|-1)}{2}\right)$$

for any bundle S, where $M(v)$ is defined as in the statement of Theorem 3.

Although the bound on the approximation error stated in Theorem 5 is considerable, it is interesting that if v has a certain property (which is not sufficient to make it easy to elicit), then the approximation error that we have if we ask only $\frac{m(m+1)}{2}$ out of the $2^m - 1$ possible value queries can be bounded in a non-trivial way.

The approximation bound of Theorem 5 is composed of two factors: the first factor, $\frac{M(v)}{2} \cdot \frac{|S|(|S|+1)}{2}$, is due to the fact that v in general is this far away from $\mathbf{G_2}$ valuations; the second factor, $\left(1 + \frac{|S|(|S|-1)}{2}\right)$, derives from the fact that the elicitor does not know the function $v' \in \mathbf{G_2}$ that best approximates v. While the first factor in the approximation error in general cannot be improved, since it derives from the fact that $v \notin \mathbf{G_2}$, a natural question is whether the elicitor might do better than the function g. We leave this as an open problem.

Allocation with $\mathrm{G_2}$ valuations

In this section, we investigate the computational complexity of the winner determination problem when all the bidders participating in the auction have valuation functions in $\mathbf{G}_2$. We focus on computing the *optimal* allocation (as is required, for example, for executing the VCG mechanism).

Theorem 6 *Computing the optimal allocation in a CA where all the bidders have 2-wise dependent valuation functions is NP-complete, even when each bidder places only*

[3] Slightly abusing the notation, we denote with ab both the bundle composed by the two items a and b, and the corresponding node in H_I.

[4] The reason for this name is the following. If a valuation has the property that all the weights in the corresponding H_I graph are non-negative, then it is super-modular. On the other hand, there exist super-modular valuations such that some of the weights in the corresponding hypercube are strictly negative. (Super-modular valuations are valuations with increasing marginal utility.)

values of 0 on individual items, and places nonzero values on only two (adjacent) edges (in fact, a value of 1 *on each of these edges).*

Proof: It is easy to see that (the decision variant of) the problem is in NP: for any assignment of items to bidders, we can compute the value of that assignment to each bidder in polynomial time, and sum these values to get the assignment's total value.

To show that the problem is NP-hard, we reduce an arbitrary instance of the NP-complete EXACT-COVER-BY-3-SETS problem to an instance of the winner determination problem as follows. Recall that in an EXACT-COVER-BY-3-SETS problem instance, we are given a set S with $|S| = m$, and subsets $S_1, S_2, \ldots, S_n$ with $|S_j| = 3$ for all j, and are asked whether $\frac{m}{3}$ of the subsets cover S. Then, in our clearing problem instance, let there be an item i_s for every $s \in S$, and, for every $S_j = \{s_j^1, s_j^2, s_j^3\}$ (where s_j^1, s_j^2, s_j^3 is an arbitrary ordering of the elements of the subset), a bid b_{S_j} which places a value of 1 on edges $(i_{s_j^1}, i_{s_j^2})$ and $(i_{s_j^2}, i_{s_j^3})$, and places a value of 0 on everything else (including all vertices). We are asked whether it is possible to obtain a value of $\frac{2m}{3}$ in this auction. We show the instances are equivalent.

First suppose there exists an exact cover by 3-sets. Then, for each S_j in the cover, give the bidder corresponding to b_{S_j} the items $i_{s_j^1}, i_{s_j^2}, i_{s_j^3}$. This is a valid allocation because none of these sets of items overlap (because none of the sets in the cover overlap). Moreover, because each such bidder's value in this allocation is 2, and there are $\frac{m}{3}$ such bidders, the total value of the allocation is $\frac{2m}{3}$. So there exists an allocation that achieves the target value.

Now suppose there is an allocation that achieves the target value. Let $n(b)$ be the number of items allocated to the bidder corresponding to bid b, and let $v(b)$ be the value of the allocation to that bidder. Then the following must hold: if this bidder receives at least one item, we must have $\frac{v(b)}{n(b)} \leq \frac{2}{3}$. Moreover, the inequality is strict unless the bidder receives exactly the three items that are endpoints of his nonzero edges. The reason is the following: $v(b)$ can be at most 2, and will be less unless the bidder receives at least the three items that are endpoints of his nonzero edges, so this is certainly true for $n(b) \geq 3$. If $n(b) = 2$, then $v(b)$ can be at most 1 and the fraction can be at most $\frac{1}{2} < \frac{2}{3}$; if $n(b) = 1$, then $v(b) = 0$. Because the value of any allocation is $\sum_{b \in W} n(b)\frac{v(b)}{n(b)}$ (where W is the set of bids that win at least one item), it follows that the target value can be achieved if and only if all items are allocated to bidders, and $\frac{v(b)}{n(b)} = \frac{2}{3}$ for all $b \in W$. But because this equality holds only if every $b_{S_j} \in W$ receives items $i_{s_j^1}, i_{s_j^2}, i_{s_j^3}$, it follows that the S_j corresponding to winning bids in an allocation achieving the target value constitute an exact cover by 3-sets. ∎

This contrasts, for example, with the case where bids are on bundles of at most size 2 (and any number of a bidder's bids can be accepted), which can be solved in polynomial time (Rothkopf, Pekeč, & Harstad 1998).

On the other hand, the following result shows that if the graph obtained by merging the G_2 graphs of the bidders displays certain structure, then the auction can be cleared in polynomial time. (Similar results have appeared for other graphical models of the bids (LaMura 1999; Sandholm & Suri 2003; Conitzer, Derryberry, & Sandholm 2004).)

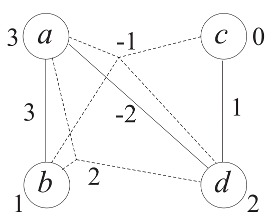
Theorem 7 *Consider the graph of all vertices (items), and all edges between items such that at least one bidder places a nonzero value on the edge. Suppose this graph has no cycles (it is a forest). Then the optimal allocation can be computed in* $O(nm)$ *time, where* n *is the number of bidders.*

Proof: The algorithm solves each tree in the forest separately. Fix a root r of the tree. For any vertex i in the tree, let $t(i, b)$ be the highest value that can be obtained in the auction from item i and its descendants alone (that is, if we throw away all other items), under the constraint that the bidder corresponding to bid b gets item i. Let $A(i, b)$ be the set of all allocations of the descendants that achieve this value. Then, for any clearing that assigns i to the bidder corresponding to b, without any loss we can change the allocation of the items in the subtree to be consistent with any element of $A(i, b)$ (assigning the descendants of i in the exact same manner); this will achieve at least as large total value from edges and vertices within the subtree; the value from all other vertices and all other edges that are disjoint from the subtree is clearly unaffected; and the only other edges that have one of the vertices of the subtree as an endpoint have i as that endpoint—and because i is still assigned to the bidder corresponding to b, they remain unaffected. Let $c_1, \ldots, c_{m_i}$ be the children of i. Then, we can conclude that $t(i, b) = v_b(i) + \sum_{k=1}^{m_i} \max\{v_b(i, c_k) + t(c_k, b), \max_{b' \neq b} t(c_k, b')\}$. This allows us to set up a simple dynamic program that will compute the $t(i, b)$ from the leaves upwards, and thus will eventually compute $t(r, b)$ for all b, and the highest such $t(r, b)$ is the optimal allocation value. We observe that for every bidder, for every edge (i, j), the value $v_b(i, j)$ is read exactly once; also, for any b and i, the expression $\max_{b' \neq b} t(i, b')$ takes only constant time to compute, because there are only two b''s for which we ever (for any b) need to look at $t(i, b')$: one that maximizes $t(i, b')$ (call it b_{i1}), and another one (b_{i2}) which maximizes $t(i, b')$ over all the remaining b' (which gives the second highest $t(i, b')$)—for the case where $b = b_{i1}$. It follows that the running time of the algorithm is $O(mn)$. The straightforward extension of the program to compute a best partial allocation $a(i, b)$ (with the restriction that i is allocated to the bidder corresponding to b) will allow for also computing an optimal allocation. ∎

Note that Theorem 7 defines a non-trivial class of costly disposal valuation functions which can be elicited using a polynomial number of value queries, and for which the winner determination problem can be solved in polynomial time. To the best of our knowledge, *this is the first class of computational and communication efficient costly disposal valuations in the literature*.

Generalization: k-wise dependency

The 2-wise dependency model can be easily extended to the case of k-wise dependency, for some $k \leq m$, by adding to the graph j-multiedges between subsets of the items of cardinality j, for any $j = 3, \ldots, k$. These multiedges account for up to k-wise dependencies between items. Given the k-wise dependency graph G_k, the valuation of a bundle S is obtained by considering the subgraph G^S induced by nodes in S, and summing up the weights of the nodes and of the edges (including multiedges) in G^S. An example of a G_4 graph, along with the corresponding valuation function, is shown in Figure 2. The class of valuations that can be expressed using a G_k graph is denoted $\mathbf{G_k}$. (We note that these are exactly the valuation functions whose hypercube representation has nonzero weights only on levels 1 through k.)



$I_A = \{a,b,c,d\}$
$v(a) = 3$
$v(b) = 1$
$v(c) = 0$
$v(d) = 2$
$v(ab) = 7$
$v(ac) = 3$
$v(ad) = 3$
$v(bc) = 1$
$v(bd) = 3$
$v(cd) = 3$
$v(abc) = 7$
$v(abd) = 9$
$v(bcd) = 4$
$v(acd) = 4$
$v(abcd) = 9$

Figure 2: Example of 4-wise dependence graph, and the corresponding valuation function v. Multiedges are represented as dashed lines.

The following proposition shows that if a valuation function is included in $\mathbf{G}_k$ for some constant $k < m$, then it can be elicited in polynomial time.

Proposition 1 *Let k be an arbitrary constant, and assume that there exist only up to k-wise dependencies between items in the valuation function v. Then, v can be elicited asking $O(m^k)$ value queries.*

The following theorems generalize some of the results presented in the previous sections to the case of k-wise dependent valuations.

Theorem 8 *Assume that the valuation function v is a δ-approximation of v', for some $v' \in \mathbf{G_k}$, with k an arbitrary positive constant. Then, a function $g \in \mathbf{G_k}$ can be learned asking $O(m^k)$ value queries, such that $v(S) = g(S)$ for any bundle S with $|S| \leq k$, and*

$$|v(S) - g(S)| < \delta \left(1 + \binom{|S|}{k}\right)$$

for any bundle S with $|S| > k$.

Theorem 9 *Let v be an arbitrary strongly super modular valuation, and let v_k be the unique valuation function in $\mathbf{G_k}$ that coincides with v on the bundles of cardinality at most k. Let $c_i(v) = \max_{S,|S|=i}\{v(S) - v_k(S)\}$, and let $M(v) = \max_{i=k+1,\ldots,m} \frac{c_i(v)}{\sum_{j=1\ldots k}\binom{i}{j}}$. Then, there exists a function $v' \in \mathbf{G_k}$ such that $|v(S) - v'(S)| \leq \frac{M(v)}{2} \cdot \sum_{j=1\ldots k}\binom{|S|}{j}$ for any bundle S. Thus, we have $d(v, \mathbf{G_k}) \leq \frac{M(v)}{2} \cdot \sum_{j=1\ldots k}\binom{m}{j}$.*

It may appear that the bound stated in Theorem 9 is looser than the one reported in Theorem 3, which would be counterintuitive. However, we have to consider that, denoting with $M_2(v)$ and $M_k(v)$ the value of M as in the statement of theorems 3 and 9, respectively, typically $M_2(v) \gg M_k(v)$. In any case, denoting by $d_2, d_3, \ldots, d_k$ the distance between v and the $\mathbf{G_2}, \mathbf{G_3}, \ldots, \mathbf{G_k}$ classes, respectively, we have $d_2 \geq d_3 \geq \ldots d_k$ because the classes subsume each other.

The $\mathbf{G_k}$ hierarchy

Let $\mathbf{G}_k$ denote the class of k-wise dependent valuations. It is clear that these classes define a hierarchy, where $\mathbf{G}_i \subset \mathbf{G}_{i+1}$ and every inclusion is strict. The bottom class of the hierarchy is the $\mathbf{G_1}$ class, which corresponds to the class of linear valuations (i.e., the valuation of any bundle is simply the sum of the values of the singletons). These valuations are easy to elicit and to allocate, but are of no interest in the CA setting. Let us consider the second element of the hierarchy, $\mathbf{G_2}$. Theorem 6 shows that valuations in this class are hard to allocate. This means that even the most limited form of interdependency between items (2-wise dependency) is sufficient to render the problem of finding the optimal allocation hard. On the other hand, valuations that display up to k-wise item dependency (where k is an arbitrary constant) remain easy to elicit. The following proposition shows that when the interdependencies are between sets of $g(m)$ objects, where $g(m)$ is an arbitrary function such that $g(m) \to \infty$ as $m \to \infty$, preference elicitation with value queries becomes hard.

Proposition 2 *Let v be an arbitrary valuation in $\mathbf{G_{g(m)}}$, where $g(m)$ is an arbitrary function such that $g(m) \to \infty$ as $m \to \infty$. Then v is hard to elicit with value queries.*

Finally, it is easy to see that the class at the top of this hierarchy, $\mathbf{G}_m$, is fully expressive, i.e., it can express any valuation function. Thus, we can end with the following theorem.

Theorem 10 *Let valuations which display up to k-wise dependencies belong to the $\mathbf{G_k}$ class. Then we have the following hierarchy:*

$$\mathbf{G_1} \subset \mathbf{G_2} \subset \cdots \subset \mathbf{G_m} \ ,$$

where every inclusion is strict. Valuations in $\mathbf{G_1}$ are easy to elicit and allocate. Valuations in $\mathbf{G_k}$, where $k \geq 2$ is an arbitrary constant, are easy to elicit and hard to allocate. Valuations in $\mathbf{G_{g(m)}}$, where $g(m)$ is an arbitrary function such that $g(m) \to \infty$ as $m \to \infty$, are hard to elicit with value queries and hard to allocate. The class at the top of the hierarchy, $\mathbf{G_m}$, contains all possible valuations.

Conclusions

We introduced the degree of interdependency between items as a key notion in combinatorial auctions, and showed that when this degree is bounded by a constant, polynomial elicitation is sufficient. Additionally, we showed how the auctioneer can approximate bidders' preferences by preferences with a bounded degree of interdependency, using only polynomially many queries. We showed that the winner determination problem is already NP-complete for preferences with degree of interdependency 2 (this worst-case hardness may not be a problem for winner determination algorithms in practice), and we also demonstrated a special case in which the winner determination problem can be solved in polynomial time.

One path for future research is to experimentally study the hardness of the communication and winner determination problems when bidders' valuations are drawn according to a model of bounded interdependency. Another path is to find and study extensions of this model that allow for richer valuation functions but nevertheless maintain (at least some of) the desirable communication and winner determination properties of valuations with bounded interdependency. Finally, we can attempt to generalize to variants and generalizations such as combinatorial reverse auctions and exchanges.

References

Ausubel, L., and Milgrom, P. 2002. Ascending auctions with package bidding. *Frontiers of Theoretical Economics* 1. No. 1, Article 1.

Ausubel, L. M., and Milgrom, P. 2006. The lovely but lonely Vickrey auction. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. chapter 1.

Blum, A.; Jackson, J.; Sandholm, T.; and Zinkevich, M. 2004. Preference elicitation and query learning. *Journal of Machine Learning Research* 5:649–667.

Chang, Y.-C.; Li, C.-S.; and Smith, J. R. 2003. Searching dynamically bundled goods with pairwise relations. In *ACM-EC*, 135–143.

Chevaleyre, Y.; Endriss, U.; Estivie, S.; and Maudet, N. 2004. Multiagent resource allocation with k-additive utility functions. In *Workshop on Computer Science and Decision Theory*.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* 11:17–33.

Conen, W., and Sandholm, T. 2001. Preference elicitation in combinatorial auctions: Extended abstract. In *ACM-EC*, 256–259.

Conen, W., and Sandholm, T. 2002. Partial-revelation VCG mechanism for combinatorial auctions. In *AAAI*, 367–372.

Conitzer, V., and Sandholm, T. 2004. Revenue failures and collusion in combinatorial auctions and exchanges with VCG payments. In *Agent-Mediated Electronic Commerce (AMEC) workshop*.

Conitzer, V.; Derryberry, J.; and Sandholm, T. 2004. Combinatorial auctions with structured item graphs. In *AAAI*, 212–218.

de Vries, S.; Schummer, J.; and Vohra, R. V. 2003. On ascending auctions for heterogeneous objects. Draft, Nov.

Groves, T. 1973. Incentives in teams. *Econometrica* 41:617–631.

Hudson, B., and Sandholm, T. 2004. Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *AAMAS*, 386–393.

Lahaie, S., and Parkes, D. 2004. Applying learning algorithms to preference elicitation. In *ACM-EC*.

LaMura, P. 1999. *Foundations of Multi-Agent Systems*. Ph.D. Dissertation, Stanford University.

Larson, K., and Sandholm, T. 2001. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK)*, 169–182.

Larson, K., and Sandholm, T. 2005. Mechanism design for deliberative agents. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Lehmann, D.; O'Callaghan, L. I.; and Shoham, Y. 2002. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM* 49(5):577–602.

Nisan, N., and Segal, I. 2005. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*. Forthcoming.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *ACM-EC*, 1–12.

Parkes, D. 1999a. iBundle: An efficient ascending price bundle auction. In *ACM-EC*, 148–157.

Parkes, D. 1999b. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*.

Rothkopf, M.; Pekeč, A.; and Harstad, R. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.

Rothkopf, M.; Teisberg, T.; and Kahn, E. 1990. Why are Vickrey auctions rare? *Journal of Political Economy* 98(1):94–109.

Sandholm, T., and Suri, S. 2003. BOB: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence* 145:33–58.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2002. Winner determination in combinatorial auction generalizations. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 69–76.

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *AAAI*, 256–262.

Sandholm, T. 2000. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce* 4(3):107–129.

Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence* 135:1–54.

Sandholm, T. 2006. Winner determination algorithms. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press.

Santi, P.; Conitzer, V.; and Sandholm, T. 2004. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Conference on Learning Theory (COLT)*, 1–16.

Tennenholtz, M. 2000. Some tractable combinatorial auctions. In *AAAI*.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16:8–37.

Wurman, P., and Wellman, M. 2000. AkBA: A progressive, anonymous-price combinatorial auction. In *ACM-EC*, 21–29.

Zinkevich, M.; Blum, A.; and Sandholm, T. 2003. On polynomial-time preference elicitation with value queries. In *ACM-EC*, 176–185.