Chapter 10: Preference Elicitation in Combinatorial Auctions

Tuomas Sandholm and Craig Boutilier

1 Motivation and introduction

The key feature that makes combinatorial auctions (CAs) most appealing is the ability for bidders to express complex preferences over collections of items, involving complementarity and substitutability. It is this generality that makes providing the input to a CA extremely difficult for bidders. In effect, each bidder must provide her *valuation function* over the space of all bundles of items. More precisely, with m items for sale, there are $2^m - 1$ bundles over which a bidder may have to provide bids.

Requiring all of this information from all bidders is undesirable for several reasons. First, determining one's valuation for any specific bundle can be computationally demanding (Sandholm, 1993, 2000; Parkes, 1999b; Larson and Sandholm, 2001), thus requiring this computation for exponentially many bundles is impractical. Second, communicating exponentially many bids can be prohibitive (e.g., w.r.t. network traffic).[1] Finally, agents may prefer not to reveal their valuation information for reasons of privacy or long-term competitiveness (Rothkopf et al., 1990).

Several approaches have been proposed for addressing the problem. Ascending CAs—see, for example, Parkes (1999a), Wurman and Wellman (2000), Ausubel and Milgrom (2002), Parkes (Chapter 2), and Ausubel and Milgrom (Chapter 3)—provide one means of minimizing the information requirements on bidders by posting prices (sometimes implicitly) on all bundles and asking bidders to reveal their demands at the current prices.[2] A more general approach has recently been proposed (Conen and Sandholm, 2001), where the auctioneer (or *elicitor*), instead of requesting bids on all bundles, asks bidders for very limited, and ideally *relevant*, information about their valuations. Through *incremental* querying, the auctioneer gradually builds up a partial model of bidder valuations, one that becomes more refined with each query, until an optimal allocation can be determined. By adopting a query strategy in which previously revealed information guides the selection of subsequent queries, elicitation is focused on pertinent information. Ideally, an optimal allocation can be determined despite the fact that each bidder's valuation function has only been partially revealed. Ascending CAs can be viewed as a special case of this model.

The preference elicitation problem in CAs is in many ways the same as that faced in decision analysis and multiattribute utility theory (Keeney and Raiffa, 1976). Indeed, the preferences expressed by bids can be seen as a multiattribute utility function in which each item is an attribute. One way to deal with the exponential bid space is to assume some structure in utilities. For instance, it is common to assume utility can be expressed as the additive combination of independent local value functions for each attribute; much more flexible, yet compact representations have also been proposed and used (Keeney and Raiffa, 1976) including graphical models (Bacchus and Grove, 1995; Boutilier et al., 2001). The use of structured models that exploit the same intuitions has been developed in CAs under the guise of *bidding languages* (Nisan, Chapter 9). In such models, one takes advantage of the fact that a utility function over an exponential outcome space (or bundle space) can sometimes be expressed with far fewer parameters. Toward the end of this chapter we present methods for elicitation in CAs that exploit such structure in individual valuation functions.

Compact models still do not address the issue of the cost of precisely computing the parameters of the model in question. Again, the analogous problem in decision analysis—the fact that humans have a hard time precisely assessing utilities—has drawn considerable attention (Keeney and Raiffa, 1976; Kahneman and Tversky, 1979; Saaty, 1980; Dyer, 1972; White et al., 1984; Salo and Hämäläinen, 2001; Chajewska et al., 2000; Boutilier, 2002; Wang and Boutilier, 2003). Typically, in practical decision analysis, comparison queries of various sorts are asked (which require only yes/no responses) rather than direct evaluation queries. These impose bounds on utility parameters, and a number of these queries are asked until a single decision is proven to be optimal, or a manageable Pareto optimal set can be presented to the user for selection. In practice, optimal decisions can be found with very limited elicitation of the utility function.

A key feature that distinguishes the preference elicitation problem in CAs from traditional preference elicitation is the fact that certain information about the preferences of one bidder may be irrelevant given the preferences of others. For instance, suppose bidder b has expressed that she prefers bundle X to bundle Y, and that bundle X is worth no more than \$100. Should the auctioneer have information about other agents that ensures revenue greater than \$100 can be obtained for Y, asking for b's valuation for Y serves no useful purpose. Thus, careful interleaving of queries among different bidders can offer potential reductions in the amount of information that needs to be elicited (Conen and Sandholm, 2001). This is not always the case—as we discuss below, worst-case results exist that show, in general, the amount of communication required to realize an optimal allocation is exponential, equivalent to at least one bidder revealing her entire valuation for all bundles (Nisan and Segal, 2005). But we will see that careful elicitation, can, in practice, offer significant savings. Such multiagent considerations give most work on elicitation in CAs a rather different character than techniques in decision analysis.

The multiagent considerations naturally apply to the more standard single-item auctions as well. Recent work has focussed on the question of how to limit the amount of valuation information provided by bidders, for example, by (adaptively) limiting the precision of the bids that are specified (Grigorieva et al., 2002; Blumrosen and Nisan, 2002). The motivation for the work can be seen as largely the same as work on preference elicitation in CAs. Of course, the problem is much more acute in the case of CAs, due to the combinatorial nature of valuation space.

We begin in Section 2 with a discussion of a general framework for elicitation, and describe relevant concepts such as certificates and incentives. Section 3 deals with a class of elicitation algorithms that use the concept of a *rank lattice*. We discuss instantiations of the more general framework in Section 4, focusing on methods that make no assumptions about valuation structure, while Section 5 deals with methods that exploit structured valuations. We conclude with discussion of future directions.

2 A general elicitation framework

We begin by describing the basic CA setting, and propose a general model in which most forms of incremental elicitation can be cast. We also describe several concepts that have a bearing on most elicitation techniques.

2.1 The setting

Consider a setting with one benevolent seller (auctioneer or arbitrator) and n buyers (bidders). The seller has a set $M = \{1, \ldots, m\}$ of indivisible, distinguishable items to sell (we assume no reservation value). Any subset of the items is called a *bundle*. The set of bidders is $N = \{1, \ldots, n\}$

Each bidder has a valuation function $v_i : 2^M \to \mathbb{R}$ that states how valuable any given bundle is to that bidder. Let $v_i(\emptyset) = 0$ for all i. These valuations are private information. We make the standard quasilinearity assumption:

the utility of any bidder i is $u_i(X_i, p_i) = v_i(X_i) - p_i$, where $X_i \subseteq M$ is the bundle he receives and p_i is the amount that he has to pay.

A *collection* $(X_1, \ldots, X_n)$ states which bundle X_i each bidder i receives. If some bidders' bundles overlap, the collection is infeasible. An *allocation* is a feasible collection (i.e., each item is allocated to at most one bidder).

We will study elicitors that find a welfare maximizing allocation (or in certain settings, a Pareto efficient allocation). An allocation X is *welfare maximizing* if it maximizes $\sum_{i=1}^{n} v_i(X_i)$ among all allocations. An allocation X is *Pareto efficient* if there is no other allocation Y such that $v_i(X_i) \geq v_i(Y_i)$ for each bidder i and strictly greater for at least some bidder i.

2.2 Elicitors

By *preference elicitation* in CAs we refer to a process by which the auctioneer queries bidders for specific information about their valuations. If we think of elicitation as a distinct process, we can view the auctioneer as augmented with an *elicitor* (most practically embodied in software) that determines what queries to pose. Given any sequence of responses to previous queries, the elicitor may decide to ask further queries, or stop and (allow the auctioneer to) determine a feasible (possibly optimal) allocation and payments. Most models and algorithms of elicitation in CAs that have been studied to date can be cast as instantiations of the following general algorithm:

1. Let C_t denote information the elicitor has regarding bidder valuation functions after iteration t of the elicitation process. C_0 reflects any prior information available to the auctioneer.
2. Given C_t, either (a) terminate the process, and determine an allocation and payments; or (b) choose a set of (one or more) queries Q_t to ask (one or more) bidders.
3. Update C_t given response(s) to query set Q_t to form C_{t+1}, and repeat.

This framework is, of course, too general to be useful without addressing some key questions. All specific algorithms for elicitation that we survey take a concrete stance on each of the following issues.

First, what queries is the elicitor allowed to pose? Examples considered in the literature include: rank queries ("What is your second-most preferred bundle?"); order queries ("Is bundle a preferred to bundle b?"); bound queries ("Is bundle a worth at least p?"); value queries ("What is bundle a worth?"); and demand queries ("If the prices for—some or all—bundles were $\vec{p}$, which bundle would you buy?"). When evaluating the effectiveness of elicitation, we generally care about the number of queries required to determine an optimal allocation. This must be considered carefully, since powerful queries like "What is your valuation?" trivialize the problem. Thus it is natural to compare the number of queries asked by the elicitor on a specific problem instance to the number of the *same type* of queries needed to realize full revelation.

Second, how is information about bidder valuations represented? This is tied intimately to the permitted query types, since different queries impose different types of constraints on possible valuations. For instance, if only bound queries are used, then upper and lower bounds on valuations (and allocations) must be maintained. This question also depends on structural assumptions that the elicitor makes about valuations. The query types above exploit no structural information (since they ask only about bundles); but if one can assume, say, that a bidder's valuation function is linear, then queries can be directed toward parameters of the valuation function. Furthermore, the representation of the consistent valuation functions can be much more compact. Finally, one might use probabilistic representations to reflect priors over valuations, for instance, to decide which queries are most likely to be useful.

Finally, the issue of termination is critical: when does the elicitor have enough information to terminate the process? Naturally, determination of an optimal or approximately optimal allocation (w.r.t. the responses offered) should be possible. However, incentive properties must also be accounted for (see below). Ideally, mechanisms should also account for the costs of elicitation (e.g., communication costs, or computational/cognitive costs imposed on the bidders).

2.3 Certificates

Since the aim of incremental elicitation is to determine optimal allocations without full valuation information, it is critical to know when *enough* information has been elicited. A *certificate* is a set of query-answer pairs that prove that an allocation is optimal.[3] The form of certificates naturally depends on the types of queries one is willing to entertain. For example, when the objective is to find a welfare-maximizing allocation, the elicitor clears the auction if, given the information it has received, it can infer that one allocation is worth at least as much as any other. A *minimal certificate* is a certificate that would cease to be a certificate if any query-answer pair were removed from it. A *shortest certificate* for a specific problem instance is a certificate that has the smallest number of query-answer pairs among all certificates.[4]

2.4 Handling incentives

Motivating the bidders to answer elicitor queries truthfully is a key issue; if the bidders lie, the resulting allocation may be suboptimal. It is well known that the *Vickrey-Clarke-Groves (VCG)* mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) makes truth-telling each bidder's dominant strategy.[5] (See Ausubel and Milgrom (Chapter 1) and Ronen (Chapter 15) for discussions of incentives.) However, the VCG mechanism, as generally defined, requires complete "up front" revelation of each bidder's valuation (in sealed-bid fashion).

When incremental elicitation is used, motivating bidders to answer queries truthfully is more difficult, since elicitor queries may leak information to the bidder about the answers that other bidders have given (depending on the elicitor's query policy). For instance, a bidder may condition her response on the precise sequence of queries asked by inferring that the current query is deemed necessary by the elicitor given responses to queries by other bidders. This makes the bidders' strategy spaces richer, and can provide incentive to reveal untruthfully.

Conen and Sandholm (2001) describe a methodology by which incremental elicitation mechanisms can be structured so that answering queries truthfully is an *ex post equilibrium*: bidding truthfully is each bidder's best strategy given that the other bidders bid truthfully. The elicitor asks enough queries to determine not only the welfare-maximizing allocation, but also VCG payments.[6] Using the welfare-maximizing allocation and VCG payments so constructed, the elicitor induces all bidders to answer their queries truthfully.[7]

A related approach to handling incentives in auctions involves proving that myopic best-responding is an *ex post* equilibrium in ascending auctions (Gul and Stacchetti, 2000). This approach has also been used in multi-unit (Ausubel, 2005) and combinatorial auctions (Ausubel, 2002). Another related approach uses proxy bidders in CAs: the proxies carry out myopic best response strategies in an ascending auction. In that case, revealing valuation information to the proxies truthfully on an as-needed basis is an *ex post* equilibrium (Parkes, 2001).

2.4.1 *Incentive-compatible* push-pull *mechanisms*

To improve revelation efficiency, the elicitor can, apart from querying (or *pulling* information), allow bidders to provide unsolicited information (i.e., *push* information), and treat it as if he had asked the corresponding query. Revelation through bidder push can be effective because the bidder has information (about her own valuation) that the elicitor does not. Revelation through elicitor pull can be effective because the elicitor has information that the bidder doesn't (about others' valuations). Because both modes have their strengths, the hybrid push-pull method can help reduce the amount of revelation compared to pure push or pull.

Bidders can also refuse to answer some of the elicitor's queries (e.g., if they are too hard to answer). As long as enough information is revealed to determine the optimal allocation and VCG payments, truth-telling is an *ex post* equilibrium. Thus, incentive properties remain intact despite the fact that the bidders can pass on queries and "answer" queries that the elicitor did not ask.[8] In the rest of this chapter, we only consider *pull* mechanisms.

2.5 Constraint network

While different elicitation algorithms may require different means of representing the information obtained by bidders, Conen and Sandholm (2001) describe a fairly general method for representing an incompletely specified valuation functions that supports update with respect to a wide variety of queries, and inference by the elicitor. A *constraint network* is a labeled directed graph consisting of one node for each bundle b representing the elicitor's knowledge of the preferences of a bidder. A *directed edge* (a, b) indicates that bundle a is (known to be) preferred to bundle b. Each node b is labeled with an interval $[LB_i(b), UB_i(b)]$, where $LB_i(b)$ is the greatest lower bound the elicitor can prove on the true $v_i(b)$ given the answers received to queries so far, and $UB_i(b)$ is the least upper bound. By transitivity, the elicitor knows that a is preferred to b, denoted $a \succeq b$, if there is a directed path from a to b or if $LB_i(a) \geq UB_i(b)$.

The free disposal assumption allows the elicitor to add the edges (a, b) to any constraint network for any $a \subseteq b$, as shown in Fig. 1. Responses to order queries can be encoded by adding edges between the pair of bundles compared, while value and bound queries can be encoded by updating the bounds. When bounds are updated, new lower bounds can readily be propagated "upstream" and new upper bounds "downstream."

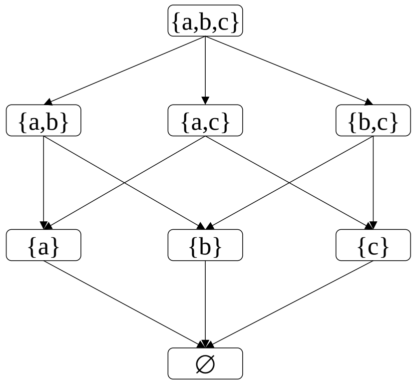

Figure 1: *A constraint network for a single bidder with 3 items a, b, and c encoding free disposal.*

The constraint network representation is useful conceptually, and can be represented explicitly for use in various elicitation algorithms. But its explicit representation is generally tractable only for small problems, since it contains 2^m nodes (one per bundle), and an average outdegree of $m/2$.

3 Rank lattice-based elicitors

We begin with discussion of a class of elicitors that use the notion of a *rank lattice* (Conen and Sandholm, 2001, 2002c). Rank-lattice elicitors adopt specific query policies that exploit the topological structure inherent in the problem to guide the elicitation of bidders' preferences.

Conceptually, bundles can be ranked for each bidder from most to least preferred. This gives a unique rank for each bundle for each bidder (assuming no indifference). Let $b_i(r_i)$ be the bundle that bidder i has at rank r_i. In other words, $b_i(1)$ is the bidder's most preferred bundle, $b_i(2)$ second-most, etc. A *rank vector* $r = [r_1, r_2, \ldots, r_n]$ represents the allocation of $b_i(r_i)$ to bidder i. Naturally, some rank vectors correspond to feasible allocations and some to infeasible collections. The value of a rank vector r is $v(b(r)) = \sum_i v_i(b_i(r_i))$. A rank vector $[r_1, r_2, \ldots, r_n]$ *dominates* rank vector $[r'_1, r'_2, \ldots, r'_n]$ if $r_i \leq r'_i$ for all bidders $i \in N$. The set of rank vectors together with the domination relation define the *rank lattice* as depicted in Figure 2.

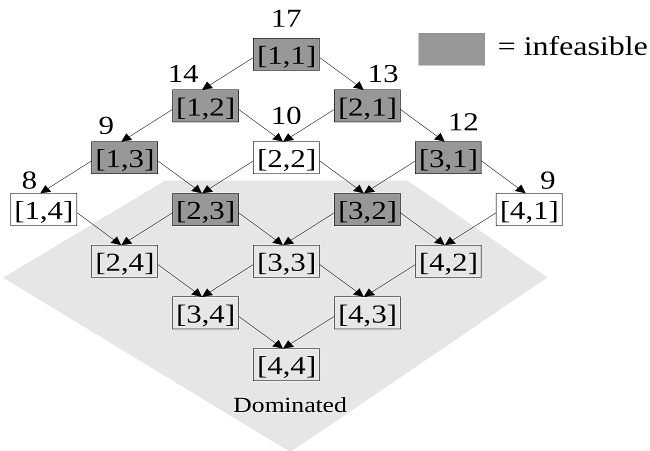

Figure 2: *Rank lattice for two bidders, 1 and 2, and two items, A and B, with the following value functions:* $v_1(AB) = 8$, $v_1(A) = 4$, $v_1(B) = 3$, $v_1(\emptyset) = 0$, $v_2(AB) = 9$, $v_2(B) = 6$, $v_2(A) = 1$, $v_2(\emptyset) = 0$. *Gray nodes are infeasible. The shaded area is the set of nodes dominated by feasible nodes. Above each node is its value.*

If a feasible collection (i.e., allocation) is not dominated by another allocation in the rank lattice, it is Pareto efficient (i.e., if the elicitor knew only the ranks of all bundles of all bidders, but had no valuation information, each such nondominated allocation is potentially optimal; see the three white nodes in Figure 2). Welfare-maximizing allocations (in this example, rank vector $[2, 2]$) can be determined using the values only of allocations in this set.

Since no allocation that lies below another in the rank lattice can be a better solution to the allocation problem, Conen and Sandholm (2001, 2002b,c) propose a series of search algorithms to find optimal allocations that exploit this fact to guide the elicitation process. Intuitively, rank-based elicitors construct relevant parts of the rank lattice by traversing the lattice in a top-down fashion, asking queries in a "natural" order. Since the rank lattice has 2^{mn} nodes, careful enumeration is critical, since the entire rank lattice cannot be examined in any but the smallest problems. Within the general elicitation model, these methods rely on rank, value, and "relative" bound queries; and assume no structure in valuations.

3.1 Using rank and value queries

The PAR (Pareto optimal) algorithm (Conen and Sandholm, 2002c) is a top-down search algorithm that uses rank queries to find all Pareto-efficient allocations. It initially asks every bidder for her most preferred bundle, constructing the collection $(1, \ldots, 1)$ that sits atop the rank lattice. As the search progresses, the elicitor asks one new query for every successor that it "constructs" in the rank lattice by asking one of the bidders for her next most preferred bundle. Specifically, starting with $(1, \ldots, 1)$ in the *fringe*, PAR chooses a node from the fringe, adds it to the Pareto-efficient set if it is feasible; if not, its children are added to the fringe. At each stage, all nodes in the fringe that are dominated by a Pareto optimal node are removed. The algorithm terminates when the fringe is empty. At termination, all Pareto optimal solutions have been identified.[9] PAR can be augmented to produce a welfare-maximizing outcome by asking value queries of all bundles that occur in the Pareto efficient set. This variant is called MPAR (maximizing PAR).

The EBF (efficient best first) algorithm (Conen and Sandholm, 2002c) is designed to find a welfare-maximizing allocation under the standard assumption of transferable and quasi-linear utility. Like PAR, it asks for each bidder's bundles in most-to-least-preferred order. However, EBF also asks for bundle *values*. These values give the search algorithm additional guidance as to which nodes to expand. The algorithm starts from the root and always expands the fringe node of highest value, while pruning provably dominated nodes. The first feasible node reached is optimal.

Unlike typical best-first search, EBF cannot always determine which node on the fringe has highest value (and thus should be expanded) given its current information; thus further elicitation is generally required. Conen and Sandholm define a simple (nondeterministic) subroutine for node expansion that defines an elicitation policy where the elicitor picks an arbitrary node from the fringe and elicits just enough information to determine its value, until it can prove which node on the fringe has highest value. Interestingly, since the elicitor uses constraint network inference to propagate value bounds, the best node in the fringe can be determined without knowing its precise value or the identity of the bundles that make it up. Determining feasibility then requires that rank queries be asked to determine each unknown $b_i(r_i)$.

Conen and Sandholm show that MPAR and EBF are, in a sense, as effective as possible in their uses of information, within the restricted class of *admissibly equipped* elicitors. An elicitor is *admissible* if it always finds a welfare-maximizing allocation; both EBF and MPAR are admissible. An elicitor is *admissibly equipped* if it can perform *only* the following operations: (a) determine the welfare of a given collection (by asking bidders for their valuations for relevant bundles); (b) determine whether a given collection is feasible or not (by asking bidders for the bundles at that rank vector); and (c) determine the next unvisited direct successors of a collection in the lattice.

Theorem 3.1 *(Conen and Sandholm, 2002c) No admissible, admissibly equipped, deterministic elicitor requires fewer feasibility checks on every problem instance than* EBF.

This result does not depend on the specific instantiation of the nondeterministic node-expansion strategy used by EBF.

Theorem 3.2 *(Conen and Sandholm, 2002c) No admissible, admissibly equipped, deterministic algorithm that only calls the valuation function for feasible collections requires fewer calls than* MPAR.

This result restricts elicitors to asking for the valuation of feasible collections. It also counts valuation calls to collections, as opposed to individual bundle-bidder pairs. In practice, we care about the latter and have no reason to restrict the elicitor to queries about feasible collections.

While EBF and MPAR are as effective as any admissible, admissibly equipped elicitor, Conen and Sandholm (2002c) show that worst-case behavior for both algorithms (hence for any algorithm in this class) is quite severe. Specifically, MPAR needs to query the value of n^m allocations in the worst case, while EBF must call the valuation routine on $(2^{mn} - n^m)/2 + 1$ collections.

The practical effectiveness of these algorithms is strongly dependent on the number of items and agents. Maintaining an explicit constraint network for each bidder and rank lattice can be problematic, since the former (i.e., number of bundles) grows exponentially with the number of items, and the latter grows exponentially with the number of relevant bundles. The hope is that by clever elicitation, EBF will obviate the need to construct anything but a very small portion of the rank lattice. Unfortunately, EBF shows rather poor empirical performance in this regard (Hudson and Sandholm, 2004): the *elicitation ratio*—the ratio of the number of queries asked by an elicitor to the total number of queries required by full revelation—is rather poor. On small problems (from two to four bidders, two to eleven items), the ratio drops with the number of items, but quickly approaches 1 as the number of bidders increases (and is close to one with as few as four bidders). This is not surprising, since bidders tend to win smaller, low-ranked bundles when the number of participants is large, forcing enumeration of large parts of the lattice.

One benefit of examining the lattice top-down occurs when considering VCG payments. Once EBF terminates, no additional queries are needed to determine VCG payments.

Theorem 3.3 *(Conen and Sandholm, 2002c) No information in addition to the information already obtained by* EBF *is necessary to determine the VCG payments.*

3.2 Differential elicitation

Conen and Sandholm (2002b) propose variants of the EBF algorithm in which the elicitor asks bidders for the *differences between valuations* rather than absolute valuations. Such *differential elicitation* methods require bidders to reveal less about their valuations. The elicitor asks rank queries (i.e., what bundle has rank k), and either *differential value queries* of the form "What is the difference between the value of bundle b and your most preferred bundle?" or *differential bound queries* of the form "Is the difference between the value of bundle b and your most preferred bundle greater than δ?"

The general differential elicitation algorithm is a modification of EBF. The key observation is that the optimal allocation minimizes the aggregated loss in utility of each bidder relative to her most preferred bundle. The algorithm therefore iteratively elicits differences between valuations relative to the (unspecified, maximum) valuation for the highest ranking bundle. We focus on the use of differential bound queries, and suppose that there is some small "accounting unit" ϵ such that all valuations must differ by some integral multiple of ϵ.

The algorithm EBF-DE proceeds like EBF differing only in its use of differential bound queries rather than value queries, and its strategy for expanding nodes. For any specific bundle-bidder pair, EBF-DE asks queries for that bundle in increasing order of difference (e.g., is the difference between the value of b and your most preferred bundle greater than 0? ϵ? 2ϵ? etc.). If the bidder responds yes, this establishes a lower bound on the difference; if she responds no, this establishes the precise difference. These lower bounds can be used to reason about domination in the rank lattice, since the lower bounds on loss (relative to the optimal bundle) for each bidder can be summed to obtain a lower bound on the aggregate loss.[10] EBF-DE ensures a welfare-maximizing allocation is found. Furthermore, like EBF, we have:

Proposition 3.4 *(Conen and Sandholm, 2002b) No information in addition to that obtained by* EBF-DE *is necessary to determine VCG payments.*

4 Elicitation with unstructured valuations

Elicitors that exploit rank lattices have a very restricted, inflexible form of query policy, intimately intertwined with the elicitor's lattice representation of the information gleaned from bidders. In this section, we describe work that offers more flexible approaches to elicitation along the lines of the general elicitation framework described in Section 2.2. We focus here on the case of unstructured valuations, deferring discussion of structured valuations to Section 5.

Conen and Sandholm (2001) describe a general framework for unstructured preferences in which the elicitor's knowledge is organized as a set of *candidates*. A candidate $c = \langle c_1, c_2, \ldots, c_n \rangle$ is any allocation (i.e., feasible collection) that, given the responses to queries so far, is potentially optimal. Given the nature of the queries considered below, one can generally impose upper and lower bounds on the value associated with each allocation in the candidate set.

One can obviously view the problem of elicitation as a game against nature, in which an optimal elicitor constructs an optimal strategy (or contingency plan) where nature chooses valuations for the bidders, thus dictating responses to the queries prescribed by the optimal policy.[11] This can be solved using tree search, but this is clearly impractical except for tiny problems (Hudson and Sandholm, 2004). This has led the development of heuristic methods for elicitation, as we now describe.

4.1 Value queries

Hudson and Sandholm (2004) consider various instantiations of the general elicitation framework in which the elicitor is restricted to asking *value queries*, in which a bidder is asked to reveal her valuation $v_i(b)$ of a specific bundle b. This sets the value of b for bidder i and can be used to set upper and lower bounds on other bundles using constraint network inference (Section 2.5). Note that without edges in the constraint network (e.g., due to free disposal), information about the value of one bundle provides no information on the value of others. All of the value-query policies considered ask each bidder for the value of the *grand bundle* M consisting of all items, since it imposes an upper bound on the value of all bundles—formal justification for eliciting the grand bundle is given by Hudson and Sandholm (2004). They also investigate the potential savings of several elicitation policies relative to the $Q = n(2^m - 1)$ value queries required by full elicitation. We let q_{min} denote the shortest certificate for a specific problem instance (here, the fewest value queries an omniscient elicitor could ask).

The *random elicitation policy* simply asks random value queries (whose answers cannot yet be inferred given the answers so far) until a optimal allocation can be determined.

Theorem 4.1 *(Hudson and Sandholm, 2004) For any given problem instance, the expected number of value queries q that the random elicitation policy asks is at most $\frac{q_{min}}{q_{min}+1}(Q+1)$.*

This upper bound guarantees relatively minor savings in elicitation since q_{min} increases with the number of agents and items. However, the pessimistic nature of the proof—that there is one minimal certificate—gives hope that in practice random elicitation may perform better than suggested. Experiments (see Fig. 3) show that, while the elicitation ratio q/Q is less than 1, and slowly decreases with the number of items, it generally offers little savings over full elicitation (Hudson and Sandholm, 2004).

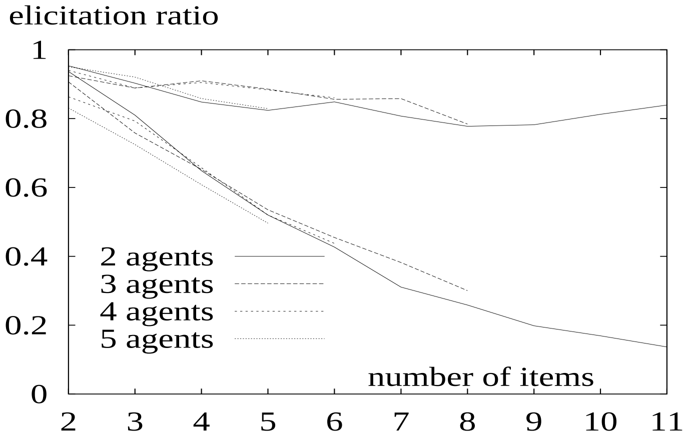

Figure 3: *Top lines: Random elicitation policy. Bottom lines: Random allocatable policy.*

The *random allocatable policy* improves on the random policy by restricting the set from which a query will be randomly selected. Note that the elicitor might know that a bundle b will not be allocated to bidder i before he knows the bidder's precise valuation for the bundle. This occurs when the elicitor knows of a different allocation which it can prove will generate at least as much value as any allocation that allocates b to agent i. If the elicitor cannot (yet) determine this (i.e., if there is a candidate in which $c_i = b$), then the bundle-bidder pair is deemed *allocatable*. The random allocatable policy is identical to the random policy with a restriction to allocatable bundle-bidder pairs. Hudson and Sandholm (2002, 2004) provide a characterization of situations in which restricting queries to allocatable pairs can and cannot lessen the effectiveness of the random policy. This restriction can never lessen the effectiveness by more than a factor of 2. Empirically, they show that the random allocatable policy performs dramatically better than the random policy, that the proportion of queries asked drops quickly with the number of items, and is unaffected by the number of bidders (see Figure 3).

The *high-value candidate* policy (Hudson and Sandholm, 2004) relies on the following intuition: to prove an allocation is optimal, we require a sufficiently high lower bound on it, and sufficiently low upper bounds on all other allocations. By only picking from high-value candidates, the high-value candidate elicitor is biased towards queries that (we expect) need to be asked anyway. In addition, by picking from those queries that will reduce as many values as possible, it is also biased toward reducing upper bounds.

More precisely, let $C_{\max}$ be the set of *high-value candidates*, those with the greatest upper bound. For each $(b, i) \in C_{\max}$, let *sub-bundles*(b, i) be the number of other bundles in $C_{\max}$ whose value might be affected upon eliciting $v_i(b)$, that is, those $(b', i) \in C_{\max}$ for which $b \supset b'$ and $LB_i(b) < UB_i(b')$. The elicitor asks value queries by choosing uniformly at random among the (b, i) with the most sub-bundles. Empirically, Hudson and Sandholm (2004) show that the high-value candidate elicitor performs better than the random allocatable elicitor (see Fig. 4 for illustrative results); it achieves an elicitation ratio of only 24% with 8 items and 3 agents, as opposed to 30% for the random allocatable policy and 78% for the random policy.

Hudson and Sandholm (2004) also evaluate the performance of an *omniscient* elicitor, one that knows each bidder's valuation function, but must ask a set of queries whose responses constitute a minimal certificate for the instance in question.[12] The performance of the omniscient elicitor provides

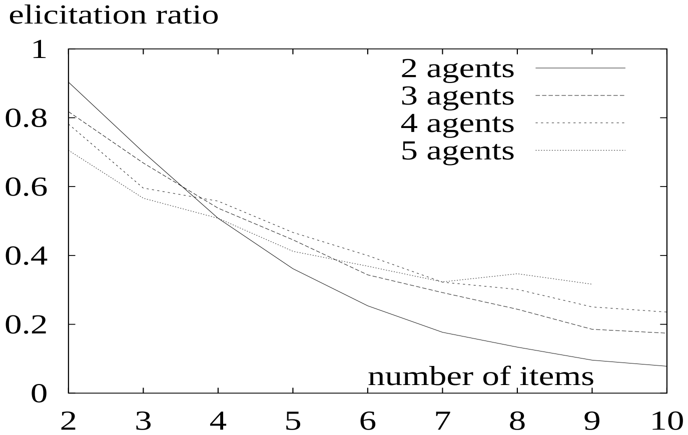

Figure 4: *High-value candidate elicitation policy. The legend is in the order of the plot lines at 2 items. The elicitation ratio falls with increasing number of items, but grows with increasing number of agents, when there are more items than agents (see (Hudson and Sandholm, 2004) for details).*

an instance-specific lower bound on that of any real elicitor.

The policies described above require intensive computation, especially if the candidate set is represented explicitly, since this scales poorly with the number of agents. Pruning of dominated candidates after any query response requires time quadratic in the number of candidates, while determining the best query (e.g., in the high-value candidate policy) and termination requires linear time, a tremendous burden since there may be as many as n^m candidates.[13]

Candidates need not be represented explicitly. Query selection can be accomplished by repeatedly solving an efficient integer program (IP) to compute the value of the highest-valued candidate (Hudson and Sandholm, 2004). With reasonable caching, the implicit approach can be several orders of magnitude faster than the explicit candidate representation.

Finally, Hudson and Sandholm (2004) address the question of whether there exist *universal* elicitors, that is, elicitors that save revelation on all instances (excluding those where even the omniscient elicitor must reveal everything).

Definition 1 *A* universal revelation reducer *is an elicitor with the following property: given any problem instance, it guarantees (always in the deterministic case; in expectation over the random choices in the randomized case) saving some elicitation over full revelation—provided the shortest certificate is shorter than full revelation. Formally: if $q_{min} < Q$, the elicitor makes $q < Q$ queries.*

By Theorem 4.1, the unrestricted random elicitor is a universal revelation reducer. In contrast:

Theorem 4.2 *(Hudson and Sandholm, 2004) No deterministic value query policy is a universal revelation reducer.*

4.2 Order queries

In some applications, bidders might need to expend great (say, cognitive or computational) effort to determine their valuations precisely (Sandholm, 1993; Parkes, 1999b; Sandholm, 2000; Larson and Sandholm, 2001), but might easily be able to see that one bundle is preferable to another. In such settings, it is natural to ask bidders *order queries*, that is, which of two bundles b or b' is preferred. A response to this query induces a new (direct) domination relation in the constraint network for the bidder. Naturally, by asking only order queries, the elicitor cannot compare the valuations of one agent with those of another, so it generally cannot determine a welfare-maximizing allocation. However, order queries can be helpful when interleaved with value queries.

Hudson and Sandholm (2004) propose an elicitation policy that combines value and order queries by simply alternating between the two, starting with an order query. Whenever an order query is to be asked, the elicitor computes all tuples (b, b', i) where bundles b and b' are each allocated to bidder i in some candidate, and where the elicitor knows neither $b' \succeq b$ nor $b \succeq b'$, and asks the order query corresponding to a random such tuple. Value queries are chosen randomly from among allocatable bundle-bidder pairs (as in the random allocatable policy above).

To compare the effectiveness of this policy against value-query-based policies, one needs to assess the relative "costs" of order and value queries. If order queries are cheap, it is worth mixing the two; otherwise using value queries alone is better. An experiment by Hudson and Sandholm suggests that (when the random allocatable policy is used to select value queries) the break-even point occurs when an order query costs about 10% of the cost of a value query.

An advantage of the mixed value-order policy is that it does not depend as critically on free disposal. Without free disposal, any policy that uses value queries only would have to elicit all values. Order queries, on the other hand, can create useful edges in the constraint network which the elicitor can use to prune candidates.

Conen and Sandholm (2001) present several other algorithms within the general elicitation framework that use combinations of value, order, and rank queries. Effectiveness of elicitation has also been demonstrated in combinatorial reverse auctions (Hudson and Sandholm, 2003) and in combinatorial exchanges (Smith et al., 2002).

4.3 Bound-approximation queries

In many settings, the bidders can more readily estimate their valuations than accurately compute them; and often the more accurate the estimate, the more costly it is to determine. This might be the case, for example, if bidders determine their valuations using anytime algorithms (Larson and Sandholm, 2001). To account for this, Hudson and Sandholm (2004) introduce the use of *bound-approximation queries* for CAs, where the elicitor asks a bidder i to tighten it's upper bound $UB_i(b)$ or lower bound $LB_i(b)$ on the value of a given bundle b.[14] This allows for elicitation of a more incremental form than that provided by value queries. The model studied by Hudson and Sandholm is one in which the elicitor can provide a hint as to how much time to spend refining this bound.

The *bound-approximation query policy* considered is one in which the elicitor determines the best query as follows: for each (b, i) pair, it (optimistically) assumes that the answer $z = UB_i(b)$ will be provided to a lower bound query (thus moving the lower bound maximally); conversely, it (optimistically) assumes that the answer $z = LB_i(b)$ will be provided to an upper bound query. The best query is that whose sum of (assumed) changes over all candidates is maximal. The policy was tested using a model of bound refinement that allowed for diminishing returns of computational effort: the marginal rate at which the bound is tightened reduces with the amount of computational effort expended. Experiments in (Hudson and Sandholm, 2002) show on small problems that this policy can reduce overall computation cost as the number of items grows (and is independent of the number of agents).

The question of VCG payments in the general elicitation framework has been studied empirically for value, order and bound-approximation queries (Hudson and Sandholm, 2003). Unlike the EBF elicitor, in which VCG payments are obtained as a side effect due to the rigid query order, additional queries will generally be required in the general model. Experimentally, once a specific elicitor found the optimal allocation (and its value), each bidder i was removed from consideration in turn, and the elicitation algorithm was continued without candidates that allocated items to i, thus allowing the optimal allocation opt_{-i} (and value) without i, hence VCG payments, to be computed.

In the 2-agent case, almost no additional elicitation is required: opt_{-i} simply allocates the grand bundle M to the agent that was not removed. Thus at most 4 additional values are needed over what is necessary to compute the optimal allocation: $v_1(opt_1)$, $v_1(M)$, $v_2(opt_2)$, and $v_2(M)$. While this argument does not generalize to more than 2 agents, in practice, relatively little extra information is needed. For example, the elicitation ratio of the bound-approximation policy is 60% at 3 agents and 5 items, and determining VCG payments only increases the elicitation ratio to 71%. Similarly, that of the value and order policy only increases from 48% to 56%.

4.4 Demand queries

Demand queries form another interesting, naturally occurring class of queries. In a demand query the elicitor asks bidder i: "If the prices on bundles $S \subseteq M$ were p_i (where $p_i : 2^M \rightarrow \mathbb{R}$), which bundle S would you buy?" In practice, prices are explicitly quoted only on a small subset of the bundles; the price of any other bundle is implicitly defined based on the prices of its subsets, for example, $p_i(S) = \max_{S' \subseteq S} p_i(S')$. The bidder would answer with the profit-maximizing bundle $\max_{S \subseteq M} v_i(S) - p_i(S)$.

Demand query elicitors can be distinguished along several dimensions:

1. Does the elicitor adopt *anonymous pricing*, by offering the same prices $p_i(S) = p(S)$ to all bidders, or *discriminatory pricing*, offering different prices $p_i(S)$ to different bidders i?
2. Are *bundle prices* used, or *item prices*? Item prices associate a price $p_i(k)$ with each item k, with $p_i(S) = \sum_{k \in S} p_i(j)$.[15]
3. How does the elicitor chooses the next query to ask?

Ascending (combinatorial) auctions—in which prices increase during the auction—can be seen as a special case of the general elicitation framework in which demand queries are used in a very specific way. There has been considerable research on on ascending CAs, much of it focused on structured valuations of various types. Indeed, most work on elicitation using demand queries arises in the context of ascending CAs. We briefly overview here results dealing with general, unstructured valuations, and return to the question of structured valuations in the next section. Ascending CAs are discussed in further detail by Parkes (Chapter 2) and Ausubel and Milgrom (Chapter 3). Elicitation using demand queries is also addressed by Segal (Chapter 11).

An important strand of research is the design of ascending CAs for general valuations (e.g., Parkes, 1999a; DeMartini et al., 1999; Wurman and Wellman, 2000). There are ascending discriminatory bundle-price auctions that yield optimal allocations in this general setting. They can be understood as linear programming algorithms for the winner determination problem: primal-dual algorithms (de Vries et al., 2003) or subgradient algorithms (Parkes and Ungar, 2000; Ausubel and Milgrom, 2002). Additional conditions on the valuation functions need to be satisfied for the auctions to also yield VCG prices (de Vries et al., 2003) (see also Parkes (Chapter 2) and Ausubel and Milgrom (Chapter 3)). It is unknown whether any *anonymous* bundle-price auction guarantees optimality (Nisan, 2003), though for restrictive definitions of "auction", the insufficiency of anonymous prices has been shown (Parkes Chapter 2). (See Bikhchandani and Ostroy (2002) for a discussion of competitive equilibria with anonymous bundle-prices.) Item-price demand queries can be used to efficiently simulate any value query (the converse is not true) (Nisan, 2003). (However, to answer a demand query, a bidder may need to solve his planning problems for a large number of bundles, if not all of them; a value query can be answered based on one plan.) Therefore, the optimal allocation can always be found using item-price demand queries, but the number of queries needed (shortest certificate) is exponential in some instances. Furthermore, even in some instances where a polynomial number of item-price demand queries constitutes a certificate, no *ascending* item-price auction (even a discriminatory one) can find the optimal allocation (Nisan, 2003).

Taken together, Blum et al. (2004) and Lahaie and Parkes (2004) show that bundle-price queries have drastically more power than item-price queries. As pointed out by Nisan (2003), any reasonable ascending bundle-price auction will terminate within a pseudopolynomial number of queries:

$$s \cdot n \cdot (\text{highest-bid/minimum-bid-increment}).$$

Here s is the number of maximum number of terms required to express any bidder's valuation in the *XOR* language (introduced in Sandholm (2002), see also Nisan (Chapter 9)). Furthermore, using bundle-price demand queries (where only polynomially many bundles are explicitly priced) and value queries together, the optimal allocation can be found in a number of queries that is polynomial in m, n, and s (Lahaie and Parkes, 2004), using techniques from computational learning theory (see Section 5.3). On the other hand, there is a nontrivial lower bound for *item-price* demand queries:

Theorem 4.3 *(Blum et al., 2004) If the elicitor can only use value queries and item-price demand queries, then $2^{\Omega(\sqrt{m})}$ queries are needed in the worst case to determine the optimal allocation. This holds even if each bidder's valuation, represented in the XOR-language, has at most $O(\sqrt{m})$ terms.*

Nisan and Segal (2005) address the communication complexity of CAs, in particular, the possibility that clever elicitation can reduce the need for a bidder to specify a valuation function over all bundles. Their main results are negative: reduction in communication complexity is not (generally) possible. While they do not rely on specific query types, these results demonstrate the importance of supporting, "personalized" prices for all bundles.[16] In particular, Nisan and Segal demonstrate that any communication protocol must reveal supporting prices; stated very roughly:

Proposition 4.4 *(Nisan and Segal, 2005) A communication protocol with message space M realizes an efficient allocation rule iff there exists a (per-*

sonalized) price vector (over bundles) for each $m \in M$ realizing a price equilibrium.

This generalizes an earlier result of Parkes (2002), who considers a more restricted query language (and explores the relationship to ascending auctions and demand queries). The notion of *dual utilities* (under mild normalization and quasi-linearity assumptions) is then used to provide a lower bound on communication. More precisely, assume a two-bidder problem and let V_1 be space of valuations of the first bidder and V_2 the second. For any $v_1 \in V_1$, we say $v_2 \in V_2$ is the *dual* of v_1 iff the social welfare of every allocation is identical given $\langle v_1, v_2 \rangle$.

Theorem 4.5 *(Nisan and Segal, 2005) If for each $v_1 \in V_1$, there is a dual $v_2 \in V_2$, then any efficient communication protocol requires at least $\log |V_1|$ bits.*

For a CA, the dual notion can be applied to show that:

Theorem 4.6 *(Nisan and Segal, 2005) The dimension of the message space in any efficient protocol for a CA with m items (with general valuations) is at least $2^m - 1$.*

This means that the communication required is at least that of one agent revealing her full valuation. This bound is tight for two agents, but an upper bound for more than $n > 2$ agents of $(n-1)(2^m - 1)$ is known not to be tight. Nisan and Segal also describe a wide variety of approximation results (see Segal (Chapter 11) for further details).

We have seen that in practice, full revelation is sometimes prevented by careful incremental elicitation. These results show, however, that in general, this cannot be the case (relative to the entire valuation of one agent). We will see the impact of these results in the case of structured valuations below.

5 Elicitation with structured valuations

Despite the worst-case complexity of preference elicitation for general valuations, for certain restricted classes, the elicitor can learn the function v_i (even in the worst case) using a number of queries that is polynomial in the number of items. Many of these results build upon work in computational learning theory (COLT) (Angluin, 1988; Angluin et al., 1993; Bshouty et al., 1994, 1995).

Many of these classes are rich enough to exhibit both complementarity and substitutability. When valuations can elicited with polynomially many (reasonable) queries, a natural approach to elicitation is to first determine each bidder's valuation completely, and then determine an allocation (and payments), rather than attempt to save elicitation effort (relative to full revelation). We will, however, see approaches that attempt to save elicitation effort even when valuations are themselves compactly specifiable.

5.1 General results

Nisan and Segal (2005) provide some very powerful worst-case results for general valuations, as discussed above. They also provide results for a number of restricted valuation classes. Significantly, using the notion of a dual valuation, they are able to show that any restricted valuation class that includes its dual valuations requires a message space that allows the specification of a full valuation in the restricted space. Thus specific results are derived for homogeneous valuations, valuations, submodular valuations (using a modified notion of dual) and substitute valuations (a lower bound based on additive valuations).

5.2 Value queries

Zinkevich et al. (2003) define the class of *read-once valuations* and show that elicitation of such a valuation is effective with value queries. A read-once valuation, analogous to a read-once formula, is a function that can be represented as a tree, where the items being auctioned label the leaves, together with the bidder's valuations for the individual items. The function's output value is obtained by feeding in a bundle S of items to the leaves and reading the valuation $v_i(S)$ from the root. A leaf sends the item's valuation up the tree if the item is included in S, otherwise the leaf sends 0. Different types of gates can be used in the nodes of the tree. A SUM node sums the values of its inputs; a MAX node takes the maximum value of its inputs; an ALL node sums its inputs *unless* one of the inputs is zero, in which case the output is 0. Read-once valuations can capture many natural preference, but are not fully expressive.

Theorem 5.1 *(Zinkevich et al., 2003) If a bidder has a read-once valuation with* SUM*,* MAX*, and* ALL *gates only, it can be learned using $O(m^2)$ value queries.*

Consider also the setting where a bidder's valuation function is a δ-approximation of a read-once function: given that the valuation function is v, there exists a read-once function v' such that for all sets S, we have $|v(S) - v'(S)| < \delta$.

Theorem 5.2 *(Zinkevich et al., 2003) Let v be a δ-approximation of a read-once function consisting of* MAX *and* SUM *gates only. Then a function v' can be learned in $m(m-1)/2$ value queries such that for any set of items S', we have $|v'(S') - v(S')| < 6\delta|S'| + \delta$.*

Zinkevich et al. (2003) consider more general gates from which to build valuation circuits: MAX_l gates, which output the sum of the l highest inputs; ATLEAST_k gates, which output the sum of its inputs if there are at least k positive inputs, and 0 otherwise; and $\text{GENERAL}_{k,l}$ gates, which generalize all of these gates by returning the sum of its highest l inputs if at least k of its inputs are positive, and returns 0 otherwise (we assume $k \leq l$.)

Theorem 5.3 *(Zinkevich et al., 2003) If a bidder has a read-once valuation with* $\text{GENERAL}_{k,l}$ *gates only, it can be learned using a polynomial number of value queries.*

While read-once valuations can be exactly learned in a polynomial number of queries, finding the optimal allocation of items to just two bidders with known read-once valuations is $\mathcal{NP}$-hard (Zinkevich et al., 2003). It is polynomially solvable if one of the two bidders has an additive valuation function.

Another restricted, but polynomially elicitable valuation class is that of *toolbox valuations*. Here, each bidder has an explicit list of k bundles $S_1, \ldots, S_k$, with values $\nu_1, \ldots, \nu_k$ respectively. The value given to a generic set S' is assumed to be the *sum* of values of the S_i contained in S', that is, $v(S') = \sum_{S_i \subseteq S'} \nu_i$. These valuations are natural if, for example, the items are tools or capabilities and there are k tasks to perform that each require some subset of tools. The value of a set of items to the agent is the sum of the values of the tasks that the agent can accomplish with those items.

Theorem 5.4 *(Zinkevich et al., 2003) If a bidder has a toolbox valuation, it can be learned using* $O(mk)$ *value queries.*

Other valuation classes learnable in a polynomial number of value queries were introduced by Conitzer et al. (2003) and Santi et al. (2004). These include valuations where items have at most k-wise dependencies, and certain other valuations. Furthermore, if two classes of valuations are each learnable in a polynomial number of queries, then so is their union—even though the elicitor does not know in advance in which of the two classes (or both) the bidder's valuation belongs. Santi et al. (2004) also present severely restricted valuation classes where learning nevertheless requires an exponential number of value queries. First steps toward a characterization of polynomial learnability of valuation functions are also given.

5.2.1 *Power of interleaving queries among agents*

Since read-once and tool-box valuations can be elicited efficiently, we may be inclined to simply elicit the entire valuation of each bidder and then optimize. However, as with the elicitation algorithms discussed above, we might also consider how to exploit the existence of multiple agents to obviate the need for full elicitation. Blum et al. (2004) consider a two-bidder setting in which each bidder desires some subset of bundles, where each of the desired bundles contains at most $\log_2 m$ items. Each bidder's valuation is 1 if he gets at least one of the desired bundles, and 0 otherwise. Observe that there are $\binom{m}{\log m}$ bundles of size $\log m$, so some members of this class cannot be represented in *poly*(m) bits. So, a valuation function in this class can require a super-polynomial number of value queries to learn. However, a polynomial number of value queries suffices for finding the optimal allocation:

Theorem 5.5 *(Blum et al., 2004) In the setting described above, the optimal allocation can be determined in a number of value queries polynomial in m.*

The approach involves randomly proposing even splits of the items between the two bidders. Because they prefer small bundles, an allocation where both bidders are satisfied is found in a small number of proposals, with high probability. Blum et al. (2004) also present a way to derandomize this protocol.

Another example by Blum et al. (2004) shows that for some valuation classes, learning the bidders' valuations is hard while eliciting enough to find the optimal allocation is easy—*even though the valuation functions have short descriptions.* Consider an "almost threshold" valuation function. It is defined by specifying a bundle S'. This bundle in turn defines a valuation function that is 1 for any bundle of size greater than or equal to $|S'|$, except for S' itself, and is 0 otherwise.

Proposition 5.6 *(Blum et al., 2004) If a bidder has an almost threshold valuation function, it can take at least* $\binom{m}{\lceil m/2 \rceil - 1}$ *value queries to learn it.*

Theorem 5.7 *(Blum et al., 2004) In a setting with two bidders, if both of them have almost threshold valuation functions, then the optimal allocation can be determined in* $4 + \log_2 m$ *value queries.*

This demonstrates that there is super-exponential power in deciding what to elicit from a bidder based on the answers received from other bidders so far.[17] On the other hand, Blum et al. also present a sufficient condition on valuation classes under which the ability to allocate the items optimally using a polynomial number of value queries implies the ability to learn the bidders' valuation functions exactly using a polynomial number of value queries.

5.3 Demand queries

As mentioned, most research on demand queries takes place in the context of ascending CAs. In this setting, considerable research has focused on valuation classes under which ascending item-price auctions yield an optimal allocation. For example, suppose items are *substitutes*: increasing the price on one item does not decrease the demand on any other item. It is well known that in this setting, some vector of anonymous item prices (i.e., *Walrasian prices*) constitutes a certificate. Several ascending CAs have been developed for (subclasses of) substitutes (Kelso and Crawford, 1982; Gul and Stacchetti, 2000; Ausubel, 2002). Furthermore, a novel mechanism for substitutes exists that requires polynomial number of item-price demand queries in the worst case (Nisan and Segal, 2005), it asks queries in an unintuitive way that does not resemble ascending auctions.

Lahaie and Parkes (2004) provide a novel use of demand queries for elicitation of concisely representable valuations. Again drawing parallels with work in COLT, they show how any query learning model can be used to elicit preferences effectively. Specifically, they use the results of Nisan and Segal (2005) to argue that complexity of preference elicitation should exhibit dependence on the size of the representation of a bidder's valuation function in a suitable representation language. By drawing a strong analogy between value queries and *membership queries* in COLT (as do Blum et al. (2004)) and demand queries and *equivalence queries*, Lahaie and Parkes show that any class of valuations can be elicited effectively if they can be learned efficiently:

Definition 2 *(Lahaie and Parkes, 2004) Valuation classes $V_1, \ldots V_n$ can be* efficiently elicited *using value and demand queries iff there is an algorithm L and polynomial p such that, for any $\langle v_1, \ldots, v_n \rangle \in V_1 \times \ldots \times V_n$, L outputs an optimal allocation for $\langle v_1, \ldots, v_n \rangle$ after no more than $p(size(v_1, \ldots, v_n), n, m)$ value and demand queries.*

Theorem 5.8 *(Lahaie and Parkes, 2004) Valuation classes $V_1, \ldots V_n$ can be efficiently elicited using value and demand queries if each V_i can be efficiently learned using membership and equivalence queries.*

Lahaie and Parkes describe an algorithm that basically simulates an efficient concept learning algorithm on each of the classes V_i until a point is reached at which an equivalence is required of each bidder by the algorithm. Then a demand query is posed using (individualized) bundle prices using current hypothesized valuations. This continues until all bidders accept the proposed bundles at the proposed prices. As a result, this algorithm does not necessarily determine each agent's valuation fully, and thus genuinely relies on the joint valuations of all bidders to guide the interaction process. Polynomial communication complexity is also shown for efficiently elicitable valuation classes. Lahaie and Parkes apply these results to show that various classes of valuation functions can be efficiently elicited.

Among the classes of functions considered by Lahaie and Parkes (2004) are *t-sparse polynomials* where each bidder i has a valuation that can be expressed as a polynomial (over variables corresponding to items) with at most t_i terms. This representation is fully expressive, but is especially suitable for valuations that are "nearly additive," and can thus be represented with few terms. Drawing on existing algorithms from learning theory, they show that only polynomially many queries are needed to learn each valuation.

The power of bundle-price demand queries is exhibited in their treatment of XOR valuations. In contrast to the results of (Blum et al., 2004) which use only item-price demand queries (and value queries), we have:

Theorem 5.9 *(Lahaie and Parkes, 2004) The class of XOR valuations can be elicited with polynomially many value and (bundle-price) demand queries.*

Finally, linear-threshold representations are considered, involving the *r-of-S* expressions: a bundle has value 1 if it contains at least r of the items specified in S, and 0 otherwise (majority valuations are a special case). These can also be learned efficiently using demand queries only (no value queries)

6 Conclusion

Preference elicitation for CAs is a nascent research area, but one of critical importance to the practical application of CAs. Though strong worst-case results are known, preliminary evidence suggests that, in practice, incremental elicitation can sometimes reduce the amount of revelation significantly compared to direct mechanisms. In certain natural valuation classes, even the worst-case number of queries is polynomial. Furthermore, as we presented, in some settings there is super-exponential power in deciding what to ask a bidder based on what other bidders have expressed so far. This offers considerable promise for research in the area, and new developments that will push the use of incremental elicitation into practice.

Future research should study concise, yet powerful (and potentially application-specific) query types and new elicitation policies. Most importantly, elicitors must be designed that address the tradeoffs among a number of different problem dimensions. Among these are:

(1) Bidder's evaluation complexity. In many CAs, the main bottleneck is to have the bidders determine (through information acquisition or computation of plans) their valuations. The mechanism should be frugal in requiring such effort. This requires a model of how hard alternative queries are to answer (see, e.g., bound-approximation queries, Section 4.3). More sophisticated and *strategic* models are presented, for example, by Sandholm (2000) and Larson and Sandholm (2001).

(2) Privacy A: Amount of information revealed to the auctioneer (less seems better). This gets addressed to some extent by optimizing (1). However, when there is a tradeoff between (1) and (2), one might be able to tune the elicitor for (1), and obtain privacy using complementary techniques. For example, the elicitor could be run as a trusted third party that only reveals the optimal allocation (and payments) to the auctioneer. Alternatively, it is sometimes possible to avoid the auctioneer's learning unnecessary information by using cryptographic techniques such as secure function evaluation (e.g., Naor et al., 1999).

(3) Privacy B: Amount of information revealed to other bidders. There is a tradeoff between revelation to the elicitor and revelation to other bidders. If the elicitor decides which query to ask based on what other bidders have stated so far, less information needs to be revealed to the elicitor. Yet it is exactly that conditioning that causes the elicitor to leak information across bidders (Conitzer and Sandholm, 2002b). An elicitor could control that tradeoff by controlling the extent of the conditioning.

    A different idea is to determine the outcome without an auctioneer, that is, via the agents communicating with each other directly. When relying on computational intractability assumptions, it is possible to devise protocols that accomplish this fully privately: no coalition of agents can learn anything about preferences of the remaining agents (except which outcome is chosen overall) (Brandt, 2003). Without intractability assumptions, fully private protocols only exist for a restricted class of mechanisms. There are protocols for first-price sealed-bid auctions, but none can exist for second-price (Vickrey) auctions (Brandt and Sandholm, 2004). It is also possible for agents to *emulate* a preference elicitor fully privately (Brandt and Sandholm, 2004b).

(4) Communication complexity. Reducing the number of bits transmitted (Nisan and Segal, 2005) is useful, and is usually roughly in line with (1). However, when there is a tradeoff between (1) and (4), it seems that in practice it should be struck in favor of (1): if a bidder can afford to evaluate a bundle, he will also be able to communicate the value.

(5) Elicitor's computational complexity. There are two potentially complex activities an elicitor will necessarily face: deciding what query to ask next, and deciding when to terminate (and which allocation(s) to return). In the elicitor designs presented in the general elicitation framework of this chapter, these operations required determining the undominated candidate allocations (complexity can creep in at the stage of assimilating a new answer into the elicitor's data structures, or in using those data structures to determine domination, or both). In some elicitation policies, determining the next query to ask can be complex beyond that. For example, even for the omniscient elicitor, finding a shortest certificate (which, in turn, allows one to ask an optimal query) seems prohibitively complex. For some elicitors, determining when to terminate may also be hard beyond the hardness of maintaining the candidates.

(6) Elicitor's memory usage. For example, maintaining the list of candidates explicitly becomes prohibitive at around 10 items for sale. The implicit candidate representation technique that we discussed exemplifies how memory usage can be traded off against time in order to move to larger auctions.

(7) Mechanism designer's objective. The designer could settle for a second-best solution in order to benefit along (1)–(6). For example, one might terminate before the optimal allocation is found, when the cost of further elicitation is likely to outweigh the benefit. (Handling the incentives in that approach seems tricky because the VCG scheme requires optimal allocation.) Furthermore, the mechanism could even be designed for the specific prior at hand, and it could be designed automatically using a computer, (e.g., Conitzer and Sandholm, 2002a). While research on automated mechanism design has focused on direct-revelation mechanisms so far, the approach could be extended to sequential mechanisms with an explicit elicitation cost. When communication and/or computational complexity becomes prohibitive, the revelation principle ceases to apply, and there can even be benefit in moving to mechanisms with insincere equilibrium play (Conitzer and Sandholm, 2004).

Acknowledgments

Parts of this research were funded by CombineNet, Inc., parts by NSF CAREER Award IRI-9703122 and NSF grants IIS-9800994, ITR IIS-0081246, ITR IIS-0121678, ITR IIS-0427858, and parts by NSERC.

Notes

[1]See Blumrosen and Nisan (2002) for a discussion of this point even in the case of single valuations.

[2]Ascending CAs are detailed in Parkes (Chapter 2) and Ausubel and Milgrom (Chapter 3).

[3]This differs slightly from the notion of certificate defined by Parkes (2002), where a certificate is a subspace of valuation functions that are consistent with the queries and their answers.

[4]With multiple query types, we could account for the "cost" of different queries.

[5]This ceases to hold if bidders can decide how much effort to invest in determining their own valuations (e.g., via information acquisition or computing) (Sandholm, 2000).

[6]The elicitor has to ask enough queries to be able to determine the welfare-maximizing allocations (among the remaining bidders) with each bidder removed. This imposes an additional elicitation burden (see the discussion of certificates above).

[7]Reichelstein (1984) already studied implications of incentive compatibility constraints on certificate complexity, and used the VCG mechanism. That work was very different because 1) it only studied single-shot communication, so the issue of the elicitor leaking information across bidders did not arise, 2) it assumed a nondeterministic communication complexity model where, in effect, the elicitor is omniscient: it knows all the bidders' preferences in advance, 3) there was only one good to be traded, and 4) that good was divisible.

[8]However, if agents can endogenously decide whether to use costly deliberation to determine their own and others' valuations, then, in a sense, there exists no truth-promoting mechanism that avoids counterspeculation unless the mechanism computes valuations for the agents, or is trivial i.e., ignores what agents reveal (Larson and Sandholm, 2004).

[9]This assumes no indifference; otherwise, a subset of Pareto optimal allocations is found.

[10]Other variants of EBF-DE are also considered by Conen and Sandholm (2002b), e.g., using differential value queries to request the precise difference or using bisection search on difference size. Some variants of EBF-DE are very similar to the iBundle(3) iterative auction (Parkes, 1999a).

[11]We contrast an optimal elicitor, which does not know the responses *a priori*, with an *omniscient* elicitor, which does and simply needs to produce a certificate.

[12]In the worst case exponential communication is required to find an (even approximately) optimal allocation in a CAs—no matter what query types and elicitation policy is used (Nisan and Segal, 2005).

[13]Techniques for speeding up the determination of domination are presented in (Hudson and Sandholm, 2003).

[14]Such queries were proposed for 1-object auctions in Parkes (1999b).

[15]One idea is to use *coherent prices*, where the price for a bundle may not exceed the sum of prices of the bundles in any partition of the bundle, and the prices for super-bundles of the bundles in the optimal allocation are additive (Conen and Sandholm, 2002a).

[16]Because of this connection, we discuss these general results in the context of demand queries, though the results do not rely on the use of demand queries for elicitation.

[17]An exponential communication *and computation* gap has been demonstrated in settings other than CAs (Conitzer and Sandholm, 2004).

References

Angluin, Dana (1988), Queries and concept learning. *Machine Learning*, 2 (4):319–342.

Angluin, Dana, Lisa Hellerstein and Marek Karpinski (1993), Learning read-once formulas with queries. *Journal of the ACM*, 40:185–210.

Ausubel, Lawrence (2005), An efficient ascending-bid auction for multiple objects. *American Economic Review*, 2005. To appear.

Ausubel, Lawrence (2002), An efficient dynamic auction for heterogenous objects. Technical report, University of Maryland, Department of Economics.

Ausubel, Lawrence and Paul Milgrom (2002), Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1. No. 1, Article 1.

Bacchus, Fahiem and Adam Grove (1995), Graphical models for preference and utility. In *Conference on Uncertainty in Artificial Intelligence*, pages 3–10, Montreal.

Bikhchandani, Sushil and Joseph Ostroy (2002), The package assignment model. *Journal of Economic Theory*, 107:377–406.

Blum, Avrim, Jeffrey Jackson, Tuomas Sandholm and Martin Zinkevich (2004), Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004. Early version in COLT-03.

Blumrosen, Liad and Noam Nisan (2002), Auctions with severely bounded communication. In *FOCS*, pages 406–415, 2002.

Boutilier, Craig (2002), A POMDP formulation of preference elicitation problems. In *National Conference on Artificial Intelligence*, pages 239–246, Edmonton.

Boutilier, Craig, Fahiem Bacchus and Ronen Brafman (2001). UCP-Networks: A directed graphical representation of conditional utilities. In *Conference on Uncertainty in Artificial Intelligence*, pages 56–64, Seattle.

Brandt, Felix (2003), Fully private auctions in a constant number of rounds. In R. N. Wright, editor, *Conference on Financial Cryptography*, volume 2742 of *Lecture Notes in Computer Science*, pages 223–238. Springer.

Brandt, Felix and Tuomas Sandholm (2004b), On correctness and privacy in distributed mechanisms. In *Agent-Mediated Electronic Commerce workshop*, pages 1–14, NYC, New York.

Brandt, Felix and Tuomas Sandholm (2004), (Im)Possibility of unconditionally privacy-preserving auctions. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 810–817, New York, NY, USA.

Bshouty, Nader, Thomas Hancock and Lisa Hellerstein (1995), Learning arithmetic read-once formulas. *SIAM Journal on Computing*, 24(4):706–735.

Bshouty, Nader, Thomas Hancock, Lisa Hellerstein and Marek Karpinski (1994), An algorithm to learn read-once threshold formulas, and transformations between learning models. *Computational Complexity*, 4:37–61.

Chajewska, Urszula, Daphne Koller and Ronald Parr (2002), Making rational decisions using adaptive utility elicitation. In *National Conference on Artificial Intelligence*, pages 363–369, Austin, TX.

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

Conen, Wolfram and Tuomas Sandholm (2001), Preference elicitation in combinatorial auctions: Extended abstract. In *ACM Conference on Electronic Commerce*, pages 256–259, Tampa, FL. A more detailed description of the algorithmic aspects appeared in the IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms, pp. 71–80.

Conen, Wolfram and Tuomas Sandholm (2002a), Coherent pricing of efficient allocations in combinatorial economies. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 254–260, NYC, NY,. Early version: AAAI-02 workshop on Game-Theoretic and Decision-Theoretic Agents.

Conen, Wolfram and Tuomas Sandholm (2002b), Differential-revelation VCG mechanisms for combinatorial auctions. In *Agent-Mediated Electronic Commerce workshop*, Bologna, Italy. Springer Lecture Notes in Computer Science LNCS 2531.

Conen, Wolfram and Tuomas Sandholm (2002c), Partial-revelation VCG mechanism for combinatorial auctions. In *National Conference on Artificial Intelligence*, pages 367–372, Edmonton.

Conitzer, Vincent and Tuomas Sandholm (2002a), Complexity of mechanism design. In *Conference on Uncertainty in Artificial Intelligence*, pages 103–110, Edmonton.

Conitzer, Vincent and Tuomas Sandholm (2002b), Vote elicitation: Complexity and strategy-proofness. In *National Conference on Artificial Intelligence*, pages 392–397, Edmonton.

Conitzer, Vincent and Tuomas Sandholm (2004), Computational criticisms of the revelation principle. In *Conference on Logic and the Foundations of Game and Decision Theory*, Leipzig, Germany. Earlier versions: short paper at ACM-EC-04; AMEC-03 workshop.

Conitzer, Vincent, Tuomas Sandholm, and Paolo Santi (2003), Combinatorial auctions with k-wise dependent valuations. Mimeo, Oct.

de Vries, Sven, James Schummer and Rakesh Vohra (2003), On ascending auctions for heterogeneous objects. Mimeo, Nov.

DeMartini, Christine, Anthony Kwasnica, John Ledyard and David Porter (1999), A new and improved design for multi-object iterative auctions. Technical Report 1054, CalTech, Social Science.

Dyer, J.S (1972), Interactive goal programming. *Management Science*, 19: 62–70.

Grigorieva, Elena, Jean-Jacques Herings, Rudolf Müller and Dries Vermeulen (2002), The private value single item bisection auction. Research Memorandum 51, METEOR, Maastricht.

Groves, Theodore (1973), Incentives in teams. *Econometrica*, 41:617–631.

Gul, Faruk and Ennio Stacchetti (2000), The English auction with differentiated commodities. *Journal of Economic Theory*, 92(1):66–95.

Hudson, Benoit and Tuomas Sandholm (2002), Effectiveness of preference elicitation in combinatorial auctions. In *Agent-Mediated Electronic Commerce workshop*, Bologna, Italy. Extended version: Carnegie Mellon University CMU-CS-02-124, March.

Hudson, Benoit and Tuomas Sandholm (2003), Generalizing preference elicitation in combinatorial auctions. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1014–1015, Melbourne, Australia. Extended version available.

Hudson, Benoit and Tuomas Sandholm (2004), Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 386–393, NYC, NY. Early versions: CMU-CS-02-124, AMEC-02, SITE-02.

Kahneman, David and Amos Tversky (1979), Prospect theory: An analysis of decision under risk. *Econometrica*, 21:263–291.

Keeney, Ralph and Howard Raiffa (1976), *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, New York.

Kelso, Alexander and Vincent Crawford (1982), Job matching, coalition formation, and gross substitues. *Econometrica*, 50:1483–1504.

Lahaie, Sebastién and David Parkes (2004), Applying learning algorithms to preference elicitation. In *ACM Conference on Electronic Commerce*, NYC, NY.

Larson, Kate and Tuomas Sandholm (2001), Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge VIII*, pages 169–182, Siena, Italy.

Larson, Kate and Tuomas Sandholm (2004), Designing auctions for deliberative agents. In *Agent-Mediated Electronic Commerce workshop*, pages 225–238, NYC, NY. A short newer version: *Strategic Deliberation and Truthful Revelation: An Impossibility Result* in ACM-EC-04.

Naor, Moni, Benny Pinkas and Reuben Sumner (1999), Privacy preserving auctions and mechanism design. In *ACM Conference on Electronic Commerce*, Denver, CO.

Nisan, Noam (2003), The power and limitations of item price combinatorial auctions. Slides from the FCC Combinatorial Bidding Conference, Queenstown, MD, Nov. 21–23.

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

Parkes, David (2001), *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania.

Parkes, David (2002), Price-based information certificates for minimal-revelation combinatorial auctions. In *Agent-Mediated Electronic Commerce workshop*, pages 103–122. Springer-Verlag, Heidelberg.

Parkes, David (1999a), iBundle: An efficient ascending price bundle auction. In *ACM Conference on Electronic Commerce*, pages 148–157, Denver, CO.

Parkes, David (1999b), Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce workshop*, Stockholm, Sweden.

Parkes, David and Lyle Ungar (2000), Iterative combinatorial auctions: Theory and practice. In *National Conference on Artificial Intelligence*, pages 74–81, Austin, TX.

Reichelstein, Stefan (1984), Incentive Compatibility and Informational Requirements. *Journal of Economic Theory*, 34:32–51.

Rothkopf, Michael, Thomas Teisberg and Edward Kahn (1990), Why are Vickrey auctions rare? *Journal of Political Economy*, 98(1):94–109.

Saaty, Thomas L (1980), *The Analytic Hierarchy Process.* McGraw-Hill, New York.

Salo, Ahti and Raimo Hämäläinen (2001), Preference ratios in multiattribute evaluation (PRIME)–elicitation and decision procedures under incomplete information. *IEEE Trans. on Systems, Man and Cybernetics*, 31(6):533–545.

Sandholm, Tuomas (1993), An implementation of the contract net protocol based on marginal cost calculations. In *National Conference on Artificial Intelligence*, pages 256–262, Washington, D.C..

Sandholm, Tuomas (2000), Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129. Special Issue on Applying Intelligent Agents for Electronic Commerce. Short version: International Conference on Multi-Agent Systems (ICMAS), pages 299–306, 1996.

Sandholm, Tuomas (2002), Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54. Earlier versions: WU-CS-99-01 1/28/99, IJCAI-99.

Santi, Paolo, Vincent Conitzer, and Tuomas Sandholm (2004), Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Conference on Learning Theory*, pages 1–16, Banff, Alberta, Canada.

Smith, Trey, Tuomas Sandholm, and Reid Simmons (2002), Constructing and clearing combinatorial exchanges using preference elicitation. In *AAAI-02 workshop on Preferences in AI and CP: Symbolic Approaches*, pages 87–93.

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

Wang, Tianhan and Craig Boutilier (2003), Incremental utility elicitation with the minimax regret decision criterion. In *International Joint Conference on Artificial Intelligence*, pages 309–316, Acapulco.

White III, Chelsea, Andrew Sage and Shigeru Dozono (1984), A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man and Cybernetics*, 14(2): 223–229.

Wurman, Peter and Michael Wellman (2000), AkBA: A progressive, anonymous-price combinatorial auction. In *ACM Conference on Electronic Commerce*, pages 21–29, Minneapolis, MN.

Zinkevich, Martin, Avrim Blum, and Tuomas Sandholm (2003), On polynomial-time preference elicitation with value queries. In *ACM Conference on Electronic Commerce*, pages 176–185, San Diego, CA.