Generalizing Preference Elicitation in Combinatorial Auctions

Benoît Hudson*
Carnegie Mellon University
Computer Science Department
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
bhudson@cs.cmu.edu

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

ABSTRACT

Combinatorial auctions where agents can bid on bundles of items are desirable because they allow the agents to express complementarity and substitutability between the items. However, expressing one's preferences can require bidding on all bundles. Selective incremental preference elicitation by the auctioneer was recently proposed to address this problem but the idea was not experimentally validated. This paper evaluates several approaches and finds that in many cases, automated elicitation is in fact beneficial: as the number of items for sale increases, the amount of information elicited is a small and diminishing fraction of the information collected in traditional "direct revelation mechanisms" where bidders reveal all their valuation information. In forward auctions, the benefit is slightly reduced by increasing the number of agents, while in reverse auctions the benefit increases with both agents and items. The exception is with rank-based elicitors, which limit themselves to eliciting the next-best bundle: these do not scale as the number of agents grows.

A full-length version of this paper is on-line at `http://www.cs.cmu.edu/~sandholm/generalizing03.pdf`

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*auctions*

General Terms

Algorithms, Experimentation, Performance

Keywords

Combinatorial auctions, electronic auctions, preference elicitation

1. INTRODUCTION

Combinatorial auctions, where agents can submit bids on bundles of items, are attractive when the bidders' valuations on bundles exhibit *complementarity* (a bundle of items is worth more than the sum of its parts) and/or *substitutability* (a bundle is worth less than the sum of its parts). Determining the winners in such auctions is a complex optimization problem that has recently received considerable attention. However, bidding in such an auction is itself a difficult problem: there are exponentially many bundles, and each agent may need to bid on all of them to fully express its preferences [5]. Appropriate bidding languages can reduce the communication overhead in some cases. We investigate the orthogonal approach of having the auctioneer propose to the agents what bundles they should bid on [1, 2]. That is, the auctioneer incrementally *elicits* information from the bidders, stopping when enough information has been elicited to determine the winners of the auction.

We ran experiments to evaluate both previously proposed and novel elicitation algorithms. In addition, we study the complexity of determining the bidders' payments. Along with combinatorial auctions we study combinatorial reverse auctions (procurement auctions). Our experiments show that only a very small fraction of the bidders' preference information needs to be elicited in order to determine the provably optimal allocation and the VCG payments. The full-length version of the paper also proves some propositions useful in designing practical elicitors.

2. ELICITOR'S CONSTRAINT NETWORK

The elicitor, as we implemented it, never asks a query whose answer could be inferred from the answers to previous queries. To support the storing and propagation of information received from the agents, we have the elicitor store its information in a constraint network. Specifically, the elicitor stores, for each agent, for each bundle, an interval $[LB_i(b), UB_i(b)]$. The lower bound $LB_i(b)$ is the highest lower bound the elicitor can prove on the true $v_i(b)$ given the answers received to queries so far. Analogously, $UB_i(b)$ is the lowest upper bound. We say a bound is *tight* when it is equal to the true value. A bound can be tightened by learning something about the value of that bundle, or by learning something about the value of a bundle related to it by the links in the constraint network.

3. RANK LATTICE BASED ELICITATION

In rank lattice based elicitation, each bidder ranks its bundles in order of decreasing valuation. The ranked lists of bundles induce a lattice of rank vectors (one entry per bidder). To find the optimal allocation, the elicitor searches the lattice in best-first order. This means that the elicitor only ever asks a bidder what its next-most-valuable bundle is and what its value is.

Figure 1 shows the result of running this elicitation policy on auctions with a varying number of agents and items. Each point in the graph is the average of 100 randomly generated experiments. As the number of items in the auction increases, the elicitation ratio decreases quickly.

Unfortunately, Figure 1 also shows that as the number of agents grows, the elicitation ratio quickly approaches 1. This phenomenon

*The material in this paper is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678. Some of the results in this paper appeared in the AMEC 2002 workshop.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

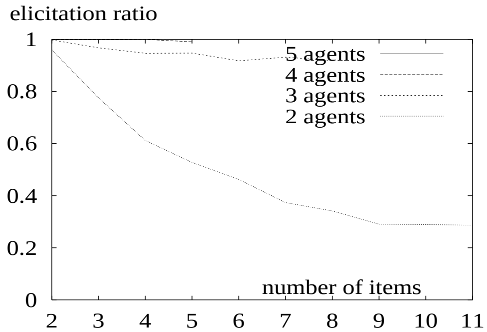

Figure 1: Performance of rank lattice based elicitation. The curves for 4 and 5 agents are barely visible, being at an elicitation ratio of almost 1.

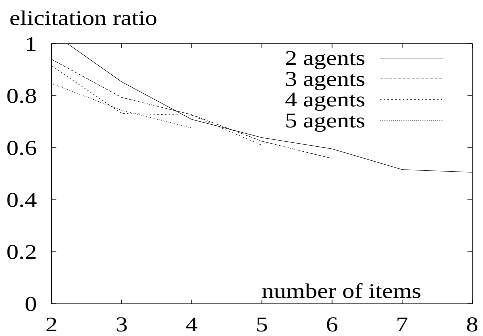

Figure 2: Elicitation using bound-approximation queries. The 2-agent, 2-item instances average an elicitation ratio greater than 1 because the method can incur up to cost 2 per bundle (1 each for tight upper and lower bounds)

can be explained as follows: as the number of agents increases, we expect agents to win, lower-ranked bundles. Because rank lattice based elicitors require the agents to reveal all high-rank bundles before any low-rank bundles, as the number of agents increases, each agent reveals a greater number of bundle values. Indeed, if there are more agents than items, at least one agent will fail to win any items and thus will reveal its entire set of preferences.

4. BOUND-APPROXIMATION QUERIES

In many settings, the bidders can easily compute rough estimates of the valuations, but the more accurate the estimate, the more costly it is to determine. In this sense, the bidders determine their valuations using anytime algorithms [4, 6]. For this reason, we introduce a new query type: a *bound-approximation query*. In such a query, the elicitor asks an agent i to spend some amount of time t tightening the agent's upper bound $UB_i(b)$ or lower bound $LB_i(b)$ on the value of a given bundle b. This query type leads to more incremental elicitation in that queries are not answered with exact information, and the information is only refined as needed.

We evaluated bound-approximation queries using an elicitation policy that seeks to maximize the change in bounds occasioned by a query, and using a reasonable model for agents' computation. Figure 2 shows that as the number of items increases, the fraction of the overall computation cost actually incurred diminishes: the optimal allocation is determined while querying only very approximate valuations on most bundle-agent pairs. The method also maintains its benefit as the number of agents increases.

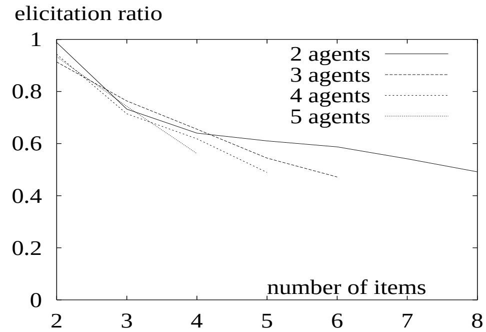

Figure 3: Bound-approximation queries in reverse auctions.

5. DETERMINING VCG PAYMENTS

Having allocated the items to the agents, the auctioneer needs to specify how much each agent should pay for its bundle. Requiring an agent to pay the amount it revealed during the elicitation algorithm has the disadvantage that agents will be motivated to lie about their preferences (and may need to spend additional computational resources to compute what preferences they should reveal). In the Vickrey-Clarke-Groves (VCG) mechanism [3] applied to a combinatorial auction (this mechanism is also known as the *Generalized Vickrey Auction*), the auctioneer charges each agent an amount equal to the negative externality that agent imposed on the other bidders; this removes agents' incentive to lie.

In general, computing the VCG payments requires more information than is elicited to compute the winners of the auction. However, the qualitative results remain the same: the elicitation ratio falls with the number of items and increases slightly with the number of agents. For example, determining winners using the bound-approximation policy on auctions with 3 agents and 5 items requires an elicitation ratio 60%, while computing VCG payments increases the ratio to 71%.

6. REVERSE AUCTIONS

While earlier work on preference elicitation has primarily focused on combinatorial forward auctions, the methodology can be adapted with very slight modifications for combinatorial *reverse* auctions as well, where there is one buyer and multiple sellers (bidders). Figure 3 shows some of the results of running our elicitors on reverse auctions with bound-approximation queries. As happens with forward auctions, the elicitation ratio in reverse auctions falls as the number of items increases. Interestingly, the experiments indicate that adding more agents to a reverse auction tends to decrease the elicitation ratio, unlike in forward auctions where the ratio slightly increases with agents.

7. REFERENCES

[1] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. *ACM-EC*, pp. 256–259, 2001. A more detailed version appeared in the IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms, pp. 71–80.

[2] Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. *AAAI*, pp. 367–372, 2002.

[3] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[4] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. *TARK*, pp. 169–182, 2001.

[5] Noam Nisan and Ilya Segal. The communication complexity of effi cient allocation problems. Draft. Second version March 5th, 2002.

[6] David C Parkes. Optimal auction design for agents with hard valuation problems. *AMEC Workshop, IJCAI*, 1999.