Constructing and Clearing Combinatorial Exchanges Using Preference Elicitation

Trey Smith and **Tuomas Sandholm** and **Reid Simmons**
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{trey,sandholm,reids}@cs.cmu.edu

Abstract

Combinatorial exchanges arise naturally in multi-agent systems that execute hierarchically decomposed tasks, when the agents have uncertainty about each other's tasks and each other's capability of handling tasks. In a combinatorial exchange, the information is aggregated to one party who then decides the task allocation among the agents. Unfortunately, such exchanges can require that bidders calculate and communicate an exponential number of bids, each of which may involve solving a hard planning problem. We present a design for an auctioneer agent that can construct and clear a combinatorial exchange using preference elicitation. This design extends existing analyses of elicitation in the combinatorial auction to the combinatorial exchange. We also introduce the concept of item discovery that uses elicitation to construct the exchange when there is uncertainty about which items should be considered in the market. Our experimental results, in a multi-robot exploration domain, show that elicitation significantly reduces the number of bids that must be evaluated in order to clear the market. More important, the proportion of bids that must be evaluated decreases as we scale to larger problem instances. We also present experimental results for an anytime version of the elicitation algorithm.

Introduction

Combinatorial exchanges are markets with the property that a single bid can jointly express both supply and demand for combinations of items. Combinatorial exchanges arise naturally in multi-agent systems that execute hierarchically decomposed tasks, when the agents have uncertainty about each other's tasks and each other's capability of handling tasks. In a market framework, an agent can express its ability to lead a team by making a bid that supplies a high-level task, indicates the cost, and demands help in the form of subtasks from other robots. For example, a science rover with a coring drill could make a bid that supplies a core sample of a rock and demands an assisting rover to help it align the drill properly. By using a combinatorial exchange, we capture the important property that supply and demand are inter-dependent, which ensures the leader does not win the contract to supply the high-level task while losing a contract for one of the necessary sub-tasks.

Unfortunately, a combinatorial exchange with n items requires that each bidder calculate and communicate its preferences by submitting as many as 2^n bids. We present a design for a auctioneer agent that can construct and clear a combinatorial exchange using *preference elicitation*; that is, by eliciting a reduced number of bids, only enough to prove that a particular *allocation* (a selection of which bids to accept) is optimal in the sense of maximizing welfare over all bidders. Elicitation is especially important when each bid itself involves solving a hard planning problem.

This paper builds on work exploring ways to use structure inherent in bidder preferences to intelligently elicit only relevant bids, while still ensuring that the market finds a welfare maximizing outcome (Conen & Sandholm 2001). We extend these existing analyses of elicitation in the combinatorial auction (one seller and multiple buyers) to the combinatorial exchange (multiple sellers and buyers).

Our auctioneer agent interrogates each bidder intelligently regarding its preferences, and optimally assimilates the returned information as bounds on the possible welfare and feasibility of allocations. These bounds can be used both to guide interim search for good allocations and to determine when the preference elicitation is complete; that is, when there is enough information to prove an allocation optimal.

We also introduce the concept of *item discovery* that constructs the exchange during the process of preference elicitation. Most previous work on combinatorial exchanges has assumed that all of the items available to trade are known to the auctioneer in advance. We relax that assumption and think of the market as a demand-driven supply chain. Only top-level goal items are initially known, and the market maker discovers new items at successively earlier stages of the supply chain in the process of eliciting bids. For example, a scientist may submit a bid that demands composition analyses for two rocks. Each composition analysis is a goal item. The auctioneer can then query potential suppliers of composition analyses, learning that they demand spectrometer readings and core samples (thus discovering these new types of items). The process can continue to lower-level sub-tasks, for instance, when the auctioneer queries a core sample supplier and learns that it needs drill alignment assistance. In this way, the auctioneer only considers supply and demand for items that are relevant to producing the goal items.

We have applied the proposed auctioneer design to a combinatorial exchange that models a multi-robot exploration task. Our results show that, for the domain in question, preference elicitation significantly reduces the number of bids that must be evaluated in order to clear the market. More important, the proportion of bids that must be evaluated decreases as we scale to larger problem instances. We also present experimental results for an anytime version of the elicitation algorithm.

Related Work

This work extends preference elicitation in the combinatorial auction (Conen & Sandholm 2001) to the combinatorial exchange.

(Walsh & Wellman 1999) examines the formation of demand-driven supply chains in a progressive distributed auction framework. That work inspired our novel concept of dynamic item discovery. Other research has modeled supply chains using distributed auctions (Babaioff & Nisan 2001) and combinatorial exchanges (Walsh, Wellman, & Ygge 2000).

Many researchers have designed algorithms to efficiently clear the combinatorial auction and generalizations, including exchanges (Sandholm *et al.* 2001a; Fujishima, Leyton-Brown, & Shoham 1999; Sandholm 2002). These algorithms rely on complete information (all bids known to the auctioneer up front). Our algorithm for clearing the market with incremental bidding repeatedly solves instances of the complete information clearing problem, so these existing algorithms can be used to great advantage as subroutines.

The application of market-based techniques to multi-robot exploration problems is a relatively new concept. In (Dias & Stentz 2000), multiple simultaneous exchanges concerning single items were used to distribute targets between a group of exploring robots.

Iterative combinatorial market techniques (Wurman & Wellman 2000; Parkes 1999; Parkes & Ungar 2000) can be viewed as another preference elicitation architecture, differing from our current work in their assumptions about query types and how auctioneer and bidder communication are interleaved. Some of their concepts, for instance using primal-dual techniques for query selection, may be applicable to our architecture.

Problem Formulation

In a combinatorial exchange, the auctioneer receives bids from both buyers and sellers concerning a set of indivisible, distinguishable items. Each subset of the set of items is called a *bundle* [1]. A bid consists of a supply bundle S, a demand bundle D, and a reward r. The semantics are that if a bid (S, D, r) is accepted, the bidder will supply the items in S, consume the items in D, and have a reward (increased happiness) of r.

There are submits two kinds of bids: *goal bids* and *production bids*. Goal bids are used to express top-level goals of the system. A goal bid has an empty supply bundle S, and a non-negative reward r. The items in the demand set, D, are top-level goals. For instance, an agent representing a scientist could announce a goal bid $(\emptyset, D, r)$ in which D is a set of compositional analyses of some rocks, and r is the importance of those analyses to the scientist.

Production bids are bids whose purpose is to support satisfying goal bids, either directly, or through supporting other production bids. A production bid (S, D, r) has a non-empty supply bundle S, and a non-positive reward r (negative rewards represent costs). The set of items in the demand bundle (possibly empty) represent what is needed by the bidder to produce the supply bundle. In our earlier example of a science rover with a drill submitting a bid (S, D, r), S would contain one item to supply (a core sample), D would contain one item that is demanded (assistance from another rover in aligning the drill bit), and the reward r would be a negative value representing the cost of the drilling operation.

Bidders can also place items in different *combination groups*. Whether or not a bidder supplies an item may affect the cost of supplying other items in the same combination group, but can *not* affect items in other combination groups. Consider a rover that is capable of performing a transport task A, a local survey task B, and communications relay task C. The transport and survey tasks each demand the rover's full attention, and force it to be present at a specific location. In this case, the cost of supplying $\{A, B\}$ may be different from the sum of the costs of supplying $\{A\}$ and $\{B\}$. This interdependence dictates that A and B must be placed in the same combination group. On the other hand, C is a background task whose cost is independent of what other items are supplied: it can be placed in a different combination group. Whenever items are known to be independent, bidders can drastically reduce the number of bids needed to completely specify their preferences. The combination group concept is similar to using OR-of-XOR bidding (Sandholm 2000; Nisan 2000), and specifying exclusions between bundles of items that share the same combination group.

The auctioneer's result is an *allocation*, a selection of accepted bids from each bidder. A *feasible allocation* is one in which the union of the accepted demand bundles is a subset of the union of accepted supply bundles (*i.e.*, demand does not exceed supply). The *welfare* of an allocation is the total reward for the accepted bids over all bidders. We want the auctioneer to find the welfare maximizing feasible allocation; this is called the *combinatorial exchange winner determination problem*, which we also refer to as the *clearing problem*.

The object of preference elicitation is to solve the clearing problem without complete information about the bidder preferences (*i.e.*, without knowing all of the bids). This is possible when there are *a priori* constraints on the preferences. In this paper, we assume the following constraints on preferences:

1. *Single method:* Each bidder b can submit at most one production bid (S, D_b, r_b) for each possible supply bundle S. This implies that a bidder can *not* express alternate ways to supply S, with different demands and rewards.

[1] Our current formulation is not a multi-unit exchange, so bundles cannot have multiple copies of the same item. However, supporting a multi-unit exchange would be an easy extension.

We make this important limitation on full expressibility in order to keep elicitation simple and tractable. We refer to the unique demand and reward corresponding to each supply bundle S as $S.D_b$ and $S.r_b$.

2. *Free disposal:* If a bidder supplies more items than are needed, it can throw away the extras at no cost. Thus supplying a smaller bundle is never harder than supplying a larger bundle. If we have bids $(S_1, S_1.D_b, S_1.r_b)$ and $(S_2, S_2.D_b, S_2.r_b)$ for two supply bundles $S_1 \subseteq S_2$ (the first supply bundle is smaller), then, correspondingly,
   (a) the first demand bundle is smaller ($S_1.D_b \subseteq S_2.D_b$) and
   (b) the first cost is smaller, equivalent to a higher reward ($S_1.r_b \geq S_2.r_b$).

For eliciting preferences, the auctioneer has the ability to make two types of queries:

1. *Value query:* ask a bidder b for its unique production bid (S, D_b, r_b) that contains a given supply bundle S. Our main emphasis is on reducing the number of value queries needed to solve the clearing problem.
2. *Type query:* ask a bidder b whether it is of the right type to produce item i, and if so, what combination group i belongs to. We assume that type queries are trivial to answer.

The auctioneer starts by knowing only the goal bids and the items they demand, which we view as the consumer end of a supply chain. All other items are discovered in the process of making value queries. When the auctioneer makes a value query for a supply bundle S, there may be unfamiliar items in $S.D_b$. In that case, the auctioneer makes a type query to each agent, concerning each unfamiliar item. Those agents capable of supplying a new item i may subsequently be queried about supply bundles containing i, discovering further new items that are needed in order to supply i. In this way, successively earlier stages of the supply chain are revealed, until we reach bids whose demand bundles are empty.

Because item discovery is initiated by value queries, and the market maker is generally able to avoid making some value queries, some items that are provably not part of the optimal allocation may never be discovered. This represents a big win in terms of minimizing the number of supply bundles that must be considered.

The primary goal of this work is preference elicitation: reducing the number of value queries needed to solve, or approximately solve, the clearing problem. The motivation is that, in many applications, appropriate bids are computationally difficult to generate. Our elicitation approach is covered in the next section.

Throughout the paper we assume that the bidders bid truthfully rather than strategically. This is a reasonable assumption, for example, in multi-robot systems, where the robots are owned by the same real-world party. Furthermore, it is well known that good strategy-proof exchanges cannot be constructed even in simple non-combinatorial exchanges. Even with only one seller, only one buyer and only one item to be traded, it is impossible to design a mechanism that (1) motivates the parties to participate in the sense that they can expect a nonnegative benefit from participating, (2) allocates the item to the party that values it the most, and (3) does not require an external benefactor (Myerson & Satterthwaite 1983).

Approach

The standard approach to the clearing problem (without preference elicitation) effectively makes all possible value queries up front, and then clears the exchange using complete information. In order to do better, we need to answer the following questions *without* complete information (*i.e.*, without knowing all of the bids):

- *Bounding preferences:* How can we use the preference constraints to infer the tightest possible bounds on bids?
- *Best allocation search:* What is the best allocation according to some metric (*e.g.*, the allocation with the highest upper bound on welfare)?
- *Optimality detection:* Do we have enough information to prove that a given allocation is welfare maximizing?
- *Query selection:* What is the best next value query to make in order to reduce the total number of value queries needed to find a provably welfare maximizing allocation?

Bounding Preferences

Here we are looking for the tightest bounds we can infer on the welfare of a given allocation. First, we consider ways to use the free disposal constraint to bound the demand and reward in a bid. Given a production bid $(S, S.D_b, S.r_b)$ for a given supply bundle S, the free disposal constraint dictates that any superset $T \supseteq S$ must have a corresponding demand $T.D_b \supseteq S.D_b$. That is, $S.D_b$ is a lower bound on $T.D_b$. If $U \subseteq S$, then $S.D_b$ is an upper bound on $U.D_b$. The free disposal constraint on reward works similarly. Thus, free disposal induces a lattice structure on supply bundles in subset/superset relationships, such that when the auctioneer gets the answer to a value query about S, it can propagate bounds on demand and reward to subsets and supersets of S.

If a combination group contains n items, we need to consider all 2^n possible sets of these items as supply bundles of production bids that we might make value queries for. The auctioneer keeps one node in memory for each supply bundle, representing the tightest known bounds on the demand and reward of the bid with that supply bundle. We refer to the data structure of nodes as the *bid lattice* [2]. Each node contains the following information about a supply bundle S:

1. Demand interval: sets $S.D_{b,\min}$ and $S.D_{b,\max}$, such that the demand bundle for bidder b is guaranteed to satisfy $S.D_{b,\min} \subseteq S.D_b \subseteq S.D_{b,\max}$. $D_{b,\max}$ may take on the special value ALL, meaning that we have no upper bound information about demand.

[2] There can be multiple bid lattices for each bidder; there is one for each combination group.

2. Reward interval: $S.r_{b,\min}$ and $S.r_{b,\max}$. The reward is guaranteed to satisfy $S.r_b \in [S.r_{b,\min}, S.r_{b,\max}]$. If $r_{b,\min} = -\infty$ or $r_{b,\max} = +\infty$, we have no information about the lower or upper bound, respectively.

Because we discover items in the process of eliciting bids, each lattice starts empty, and new items are added as elicitation progresses. Thus we must support two basic operations on the bid lattice data structure:

1. Adding a new item i. For every node with supply bundle S already in the lattice, we add a new node with supply bundle $T = S \cup \{i\}$ and propagate bounds information from subsets of T, as explained below.
2. Propagating bounds from a value query. When a query is made for a supply bundle S, returning $S.D_b$ and $S.r_b$, we do the following:
   (a) Store the now exact bounds on S:
   - $S.D_{b,\min} \leftarrow S.D_b$; $S.D_{b,\max} \leftarrow S.D_b$
   - $S.r_{b,\min} \leftarrow S.r_b$; $S.r_{b,\max} \leftarrow S.r_b$

   (b) Propagate the bounds to supersets and subsets according to the constraints of free disposal:
   - For every superset T of S, set $T.D_{b,\min} \leftarrow T.D_{b,\min} \cup S.D_b$ and $T.r_{b,\max} \leftarrow \min(T.r_{b,\max}, S.r_b)$.
   - Similarly, for every subset U of S, set $U.D_{b,\max} \leftarrow U.D_{b,\max} \cap S.D_b$ and $U.r_{b,\min} \leftarrow \max(U.r_{b,\min}, S.r_b)$.

   Figure 1 shows progressive states of a lattice corresponding to a combination group with two items: A and B. Rectangles in the figure are nodes, and bounds on D_b and r_b are shown as intervals $\min \ldots \max$. In stage (1), we see the lattice before any value queries are made. Stages (2a) and (2b) show bound propagation after the auctioneer queries about $S = \{A\}$ receives the bid $(S = \{A\}, D_b = \{X\}, r_b = -50)$.

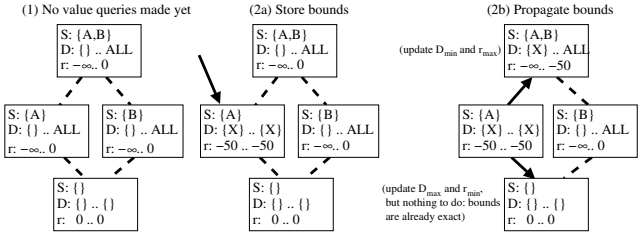

Figure 1: Propagating bounds from a value query.

Best Allocation Search

In an allocation, exactly one bid is accepted from each bid lattice. We use L to denote the set of all bid lattices, and $S_A(l)$ to denote the supply bundle of the bid that allocation A accepts from lattice $l \in L$. Further, noting that each lattice l belongs to a single bidder, we will use the notation $S_A(l).D$ and $S_A(l).r$ to mean $S_A(l).D_b$ and $S_A(l).r_b$, where b is the bidder for lattice l. An allocation A is feasible if demand does not exceed supply; that is, if $\bigcup_{l \in L} S_A(l).D \subseteq \bigcup_{l \in L} S_A(l)$. We define the welfare as

$$W(A) = \begin{cases} \sum_{l \in L} S_A(l).r & \text{if } A \text{ is feasible} \\ -\infty & \text{otherwise} \end{cases}$$

We also define the *best case welfare* $W_{\text{best}}(A)$ and *worst case welfare* $W_{\text{worst}}(A)$ of an allocation. To calculate these measures, we apply a *selection function* that fills in the missing information in a bid to form a complete virtual bid, according to rules that we define:

1. The *best case* selection function sets the virtual bid from bidder b for S to be $(S, D_b, r_b) \leftarrow (S, S.D_{b,\min}, S.r_{b,\max})$.
2. The *worst case* selection function sets the virtual bid from bidder b for S to be $(S, D_b, r_b) \leftarrow (S, S.D_{b,\max}, S.r_{b,\min})$.

Once a selection function has generated complete virtual bids, we calculate the welfare $W_{\text{worst}}(A)$ or $W_{\text{best}}(A)$ according to the definition already given. Note that the auctioneer can calculate these measures based on the information it has, and they satisfy $W_{\text{worst}}(A) \leq W(A) \leq W_{\text{best}}(A)$.

Each selection function transforms the partial information in the bid lattice to complete information, representing a best case or worst case scenario. Finding the allocation with the highest best case or worst case welfare is then an instance of the combinatorial exchange winner determination problem with complete information. This is an NP-hard and inapproximable problem. There are good existing algorithms for the related problem of clearing the combinatorial auction (Sandholm *et al.* 2001a; Fujishima, Leyton-Brown, & Shoham 1999). The combinatorial exchange is substantially harder, with the best algorithms clearing an exchange of a few hundred bids in a few seconds (Sandholm *et al.* 2001b).

Our current approach searches depth-first, branching on which bid is accepted from each bid lattice, with branch-and-bound pruning. This is structurally similar to some of the advanced algorithms; however, we did not invest the effort to duplicate their sophisticated heuristics, because our experiments do not include timing results.

Optimality Detection

Optimality detection is the problem of determining when we have enough information to prove that an allocation A is optimal. To do so, we need to show that $W(A) \geq W(B)$ for every allocation B. Assuming A and B are feasible, this is equivalent to:

$$\sum_{l \in L} S_B(l).r - S_A(l).r \quad \leq \quad 0$$

We call the term for each l in the above sum the *advantage* of A over B in lattice l. What upper bound can we calculate on the advantage in order to prove $W(A) \geq W(B)$?

One obvious answer is that

$$S_B(l).r - S_A(l).r \leq S_B(l).r_{\max} - S_A(l).r_{\min}$$

Summing over all lattices, this upper bound gives us the sufficient condition for superiority of $W_{\text{worst}}(B) \leq W_{\text{best}}(A)$, which is not surprising. But we can do better: if $S_A(l) \subseteq S_B(l)$, free disposal dictates that $S_B(l).r \leq S_A(l).r$, so that we get the tightest possible bound

$$S_B(l).r - S_A(l).r \leq \min(0, S_B(l).r_{\max} - S_A(l).r_{\min})$$

A simple way to check that A is optimal is to use this bound to compare A to every other allocation B. Unfortunately, there are exponentially many allocations to compare against. We can save some of this effort by defining a new selection function called the *upper bound advantage over A (adv(A))*, as follows: set the virtual bid for a supply bundle S in lattice l belonging to bidder b to be $(S, D_b, r_b) \leftarrow (S, S.D_{b,\min}, r_{\text{adv}})$, where

$$r_{\text{adv}} = \begin{cases} \min(0, S.r_{b,\max} - S_A(l).r_{b,\min}) & S_A(l) \subseteq S \\ S.r_{b,\max} - S_A(l).r_{b,\min} & \text{otherwise} \end{cases}$$

This selection has the property that $W(B) \leq W(A)$ is provable given our current information if and only if $W_{\text{adv}(A)}(B) \leq 0$. We can use our existing techniques to quickly find the allocation B^* which maximizes $W_{\text{adv}(A)}(B^*)$. If $W_{\text{adv}(A)}(B^*) \leq 0$, A is optimal. This optimality test is not only sound but also *complete*, meaning that it flags optimality of an allocation as soon as we have enough information to prove it.

We should also mention a much simpler optimality condition: find the allocation A with highest best case welfare. If the auctioneer has tight bounds on the welfare of A ($W_{\text{worst}}(A) = W_{\text{best}}(A)$), then A is optimal. This condition is sound, but not complete. However, for our domain, it essentially always becomes true at the same time as the complete condition explained above, and is much faster to compute.

Query Selection

Query selection is the problem of finding the best next value query to make in order to reduce the total number of value queries needed to find a provably welfare maximizing allocation. At this stage in our research, we are focusing on heuristic rules for query selection, and we do not yet have a general theory for how well the auctioneer can do.

Currently, we refine the allocation A with the highest best case welfare $W_{\text{best}}(A)$. Refining A means making a value query for one of the bids accepted by A that we do not have complete information about. The intuition is that the auctioneer should continue working on A until A is either provably optimal or no longer the best candidate. Among the bids accepted by A, we currently query about the one closest to the consumer end of the supply chain.

Overall Elicitation Algorithm

We now have the building blocks for a full elicitation algorithm:

1. Find the allocation A that maximizes $W_{\text{best}}(A)$.
2. If A is optimal, we are done. Check using either of the two optimality conditions described above.
3. If desired, return an anytime result as follows:
   (a) The anytime answer is the allocation B that maximizes $W_{\text{worst}}(B)$.
   (b) Let C be the allocation that maximizes $W_{\text{adv}(B)}(C)$. The *solution quality* of B, defined as $W(B)$/OPT, has a lower bound of $\frac{W_{\text{worst}}(B)}{W_{\text{worst}}(B)+W_{\text{adv}(B)}(C)}$. If desired, we can use this bound to halt the anytime algorithm when the quality is provably above $1 - \epsilon$ for a given ϵ.
4. Make a value query concerning one of the bids accepted by A that the auctioneer does not yet have complete information about. Propagate bounds in the corresponding bid lattice.
5. Return to step 1.

Experiments

We applied the elicitation algorithm described above to a multi-robot exploration domain. This domain is a greatly simplified version of a detailed multi-robot Mars rover simulation we are currently developing. In particular, there are n_t science targets (in our experiments, $n_t = 5$), whose positions are independently drawn from a uniform distribution over the unit square. There is one goal bid for each target: this bid demands a set of tasks to be performed at the target location. All goal bids have the same reward, r_0. The tasks are distributed among targets as evenly as possible. The number of items in the exchange is the same as the number of tasks. A single bidder submits all of the goal bids on behalf of scientists.

There are n_r rovers (in our experiments, $n_r = 4$) with differing capabilities. The initial position of each rover is random on the unit square. For every task, there is at least one rover with the capability of performing that task. Capabilities are distributed randomly among rovers, as evenly as possible. The number of lattice nodes in the exchange is determined by the number of capabilities distributed to the rovers. When reporting the number of nodes in the exchange, we count only non-trivial nodes (those with non-empty supply bundles).

If a rover performs a task, it must visit the target referenced by that task, incurring a motion cost equal to the distance traveled. Thus, to find the total cost of performing a set of tasks, the rover calculates the distance traveled along the minimum length path from its starting point that visits all its targets. Finding this TSP-like optimal path for a large number of targets is an example of the kind of hard planning problem that we wish to avoid through reducing the number of value queries. Since visiting one target affects the cost of visiting any other target, each rover has one combination group that contains all the tasks it is capable of.

The uniform reward per goal bid, r_0, is chosen so that the minimum cost allocation with all goal bids satisfied has total cost and total reward equal. With this reward setting, about half of the goal bids are accepted in the optimal allocation for a typical problem instance.

Scaling Experiment

In this experiment, we looked at the effect of exchange size on the *elicitation ratio:* the proportion of non-trivial nodes that the auctioneer queried about before finding a provably optimal allocation. Average results over five runs are shown in Figure 2. We see that when the number of nodes increases, the elicitation ratio generally drops. When the number of items increases, the elicitation ratio increases slightly (for

instance, the elicitation ratio line for 12 items is generally above that for 10 items).

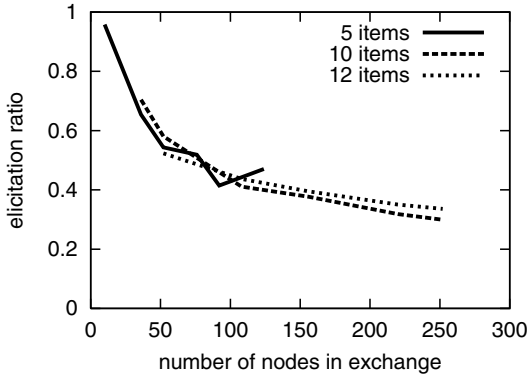

Figure 2: Scaling experiment: elicitation vs. exchange size.

For the domain studied, elicitation allows significant and increasing savings in bidder effort as the exchange scales. Competing factors influence the ratio:

1. When there are a larger number of nodes in each bid lattice, each query has the ability to affect more nodes through bounds propagation; this leads to a lower elicitation ratio as the number of bids increases.
2. On the other hand, increasing numbers of items and nodes mean that there are more similar allocations whose cost is difficult to differentiate, so that more queries are needed to prove any one allocation optimal.

Of these factors, (1) appears to be more important in our domain when scaling the number of nodes. An apparent exception is seen near the right end of the 5 items line, where the elicitation ratio rises as the problems approach node saturation (*i.e.*, approach the maximum possible number of nodes given the number of items and the other fixed parameters of the domain). More research is needed to determine how domain parameters and query selection influence scaling.

Anytime Experiment

This experiment shows how the solution quality of the best allocation so far varies as a function of the number of value queries. After each query, the auctioneer reports the allocation A with the highest worst case welfare. Two measures of A were taken: the actual quality $W(A)$/OPT, and the auctioneer's provable lower bound on quality given its information so far. Median results over five runs are shown in Figure 3. Tests over a wide range of exchange sizes gave qualitatively similar results. An alternative anytime scheme of reporting the allocation with highest best case welfare performed much worse.

Most of the improvement in the lower bound on solution quality comes very near the end of the run, so approximation schemes that stop early based on the lower bound reaching some high quality $1 - \epsilon$ will save only marginal effort.

On the other hand, the actual quality typically reaches 1 (optimality) about 20-50% more quickly than the lower bound. Thus, if we are willing to give up a provable quality

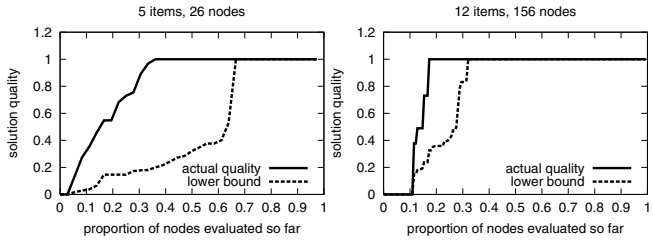

Figure 3: Anytime experiment: solution quality vs. queries.

bound, the auctioneer could do better. On repeated instances of similar problems, it could save significant effort by learning how many queries are needed before it has a probably approximately optimal solution.

Conclusions and Future Research

We have presented a design for a auctioneer agent that can construct and clear a combinatorial exchange using preference elicitation. Our design extends existing analyses of elicitation in the combinatorial auction to the combinatorial exchange. We also introduced the concept of item discovery during the process of preference elicitation. Our experimental results, in a multi-robot exploration domain, show that elicitation allows significant and increasing savings in bidder effort as the exchange scales. Tests of an anytime version of the elicitation algorithm show that we could save further effort by stopping early, if we are willing to accept a probably approximately optimal solution.

In future work, we hope to relax the single method of production constraint, allowing robots to have multiple plans for supplying the same items, with different corresponding demands and costs. We are also working on an architecture for coordinating multiple combinatorial exchanges. Each bidder communicates with one local exchange, and the local exchanges are themselves bidders in a larger exchange. In a distributed system, this architecture can minimize bottlenecks and delays due to communication latency.

References

Babaioff, M., and Nisan, N. 2001. Concurrent auctions across the supply chain. In *ACM-EC*.

Conen, W., and Sandholm, T. 2001. Minimal preference elicitation in combinatorial auctions. In *Proc. of the IJCAI Workshop on Economic Agents, Models, and Mechanisms*.

Dias, M. B., and Stentz, A. 2000. A free market architecture for distributed control of a multirobot system. In *Proc. of IAS*.

Fujishima, Y.; Leyton-Brown, K.; and Shoham, Y. 1999. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. of IJCAI*, 548–553.

Myerson, R., and Satterthwaite, M. 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 29.

Nisan, N., and Segal, I. 2001. The communication complexity of efficient allocation problems. In *Proc. of the*

DIMACS Workshop on on Computational Issues in Game Theory and Mechanism Design.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *Proc. of ACM-EC*.

Parkes, D., and Ungar, L. 2000. Iterative combinatorial auctions: Theory and practice. In *Proc. of AAAI*, 74–81.

Parkes, D. 1999. Optimal auction design for agents with hard valuation problems. In *Proc. of the IJCAI Workshop on Agent-Mediated Electronic Commerce*.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2001a. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. of IJCAI*, 1102–1108.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2001b. Winner determination in combinatorial auction generalizations. In *Proc. of AGENTS Workshop on Agent-Based Approaches to B2B*, 35–41.

Sandholm, T. 2000. eMediator: A next generation electronic commerce server. In *Proc. of AGENTS*, 73–96.

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

Walsh, W. E., and Wellman, M. P. 1999. Efficiency and equilibrium in task allocation economies with hierarchical dependencies. In *Proc. of IJCAI*, 520–526.

Walsh, B.; Wellman, M.; and Ygge, F. 2000. Combinatorial auctions for supply chain formation. In *Proc. of ACM-EC*.

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