Computationally Limited Agents in Auctions

Kate Larson
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213 , USA
klarson@cs.cmu.edu

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

ABSTRACT

Auctions provide efficient and distributed ways of allocating goods and tasks among agents. In this paper we study optimal strategies for computationally limited agents, where agents must use their limited resources in order to compute valuations for (bundles of) the items being auctioned. Agents are free to compute on any valuation problems including their opponents'. The deliberation actions are incorporated into the agents' strategies and different auction settings (both single–item and combinatorial) are analyzed in order to determine equilibrium strategies. We show that is some auction mechanisms, but not others, in equilibrium the bidders compute on others' problems as well. It is shown that under our model of bounded rationality, the generalized Vickrey auction (GVA) looses its dominant strategy property. The model of bounded rationality impacts the agents' equilibrium strategies and so must be considered when designing mechanisms for computationally limited agents.

1. INTRODUCTION

Auctions provide efficient and distributed ways of allocating goods and tasks among agents and are becoming prevalent as electronic marketplaces on the Internet. In general, Internet auctions are aimed at two groups. Auction sites such as eBay, and Yahoo! Auctions host auctions where most participants are individual consumers. Whether bidding on baseball cards or digital cameras, the individual consumers' valuations for the items are often idiosyncratic. The items have some internal value for the bidder which may be difficult for others to determine. Auction sites such as AltraNet.com instead focus on business to business (B2B) transactions. Whether buying electricity or bidding on delivery jobs, the bidders' valuations of the items can often be determined in an objective manner. The items have some concrete use to the buyer. They will become part of supply chains or incorporated into schedules. Local plans and schedules can be computed, thus determining the worth of an item to a business.

Lately there has been interest in designing software agents that represent users when bidding in online auctions. A customer in a consumer–to–consumer auction can tell his agent his valuations. The agent can then visit the auction site and represent the customer, bidding optimally given the auction type and the specified valuations. In a B2B auction, computation often must be performed in order to determine the valuations of the items up for auction. The customer bidding in such an auction, can specify its current constraints to a software agent which computes in order to determine the valuation for an item, and then places a bid in the auction. However, the valuation determination problem will possibly involve computing on such NP–complete problems as scheduling and determining delivery routes. Optimal solutions may not be possible to determine. Instead, anytime algorithms, such as iterative improvement algorithms (see, for example, [Kirkpatrick *et al.*, 1983]), can be used where at any point in time the algorithms can return a solution, but as more time is spent on computing the solution to the problem, the solution quality increases. In auctions there are inherent deadlines. For example, a participant must submit a bid before the auction closes, possibly before computing a final solution to its optimization problem. These deadlines force software agents to make tradeoffs between solution quality and computing time.

Game theory is a useful tool to help in the design of bidding agents for auctions. There is a vast literature describing *rational* agents' optimal bidding strategies in different types of auctions [Milgrom, 1989]. However, economic models for *bounded rational* agents have often been more descriptive rather than prescriptive [Simon, 1955; Rubinstein, 1998].

We study settings where agents have to compute in order to determine their valuations for (bundles of) the items being auctioned. However, they have limitations on their computation. They may have to bid before they know their true valuation for the item, or may have to use items won in the auction, before the "best" value for the items can be determined. Therefore, what an agent bids will depend on what valuation it has computed for the item, as well as what valuations the other agents may have computed.

In the next two sections we give a brief overview of different auction mechanisms and provide a description of our model of bounded rationality and a fully normative deliberation control method. We formally define an agent's strategy, incorporating both deliberation actions and bidding actions and introduce the concept of strategic computation. We then analyze different auction settings, determining when strategic computation would or would not occur in equilibrium. We conclude by noting that agents' optimal strategies depend on the model of bounded rationality.

2. AUCTIONS

Auctions can be characterized based on whether there is one item being auctioned or multiple items. We discuss the setting where the items are desirable (goods, resources, services, etc.). In this setting the seller wants to maximize the revenue she gets while each bidder tries to minimize what he has to pay for the item(s). All of our results apply to the setting where the items are undesirable (e.g., tasks), the seller wants to minimize cost and the bidders want to maximize.

In this paper we investigate inherently private value auctions with risk neutral agents. This means that the value of an item depends only on an agent's own preferences.

2.1 Sale of a Single Item

There are many different auction mechanisms, but we shall discuss the standard ones.

In an *English auction* each bidder is free to raise its bid. When no bidder is willing to increase the bid further, the auction ends with the item being allocated to the agent with the highest bid. That agent pays the amount of its bid. For rational agents there is a dominant bidding strategy. Agents keep bidding some small amount ϵ more than the previous high bid until they reach their valuation. They stop bidding at that point.

In a *first–price sealed–bid auction* each agent submits one bid without knowing the other agents' bids. The highest bidder wins the item and pays the amount of her bid. There is no dominant bidding strategy for rational agents. An optimal strategy depends on the bids of the other agents. There is a symmetric Nash equilibrium for N risk–neutral bidders, if the valuation, v_i, for the bidders are drawn independently from a uniform distribution.

In a *Dutch auction* the auctioneer lowers the price until some bidder takes the item at the current price. Again, there is no dominant bidding strategy. In fact, the Dutch auction is strategically equivalent to the first–price sealed–bid auction.

The fourth commonly discussed auction type is the *Vickrey auction* or *second–price sealed–bid auction*. Each bidder submits one bid without knowing what the others' bid. The highest bidder wins the item but pays the amount of the second highest bid. For rational agents there is a dominant strategy which is for each agent to bid its true valuation.

2.2 Sale of Multiple Items

In auctions where multiple distinguishable items are sold, bidding strategies for agents can become much more complicated. A bidder's valuation for a combination of items might not be the sum of the individual items' valuations. It may be greater, smaller, or the same.

In traditional auction formats where items are auctioned separately, in order to decide how much to bid on an item, an agent needs to estimate which other items it will receive in the other auctions. This can lead to inefficient allocations where bidders do not get the combinations they want or else get combinations that they do not want [Sandholm, 2000].

Combinatorial auctions can be used to overcome these deficiencies. In a combinatorial auction, bidders may submit bids on combinations of items which allows the bidders to express complementarities between items. Based on the bids on the combinations of items, or *bundles*, the goods are *allocated* to the agents. Let $X = \{x_1, \ldots, x_n\}$ be a set of items. A bundle is a subset of the items, e.g., $\{x_1\}$ or $\{x_1, x_n\}$. An allocation of items among A agents is $Y = (y_1, \ldots, y_A)$ where $y_i \subseteq X$, $\cup_{i=1}^{A} y_i = X$ and $y_i \cap y_j = \emptyset$ for $i \neq j$. The *generalized Vickrey auction* (GVA) is a type of combinatorial auction where the payments are structured so as to motivate bidders to bid truthfully and has been suggested as a useful protocol for electronic auctions [Varian, 1995].

2.2.1 The GVA Protocol

Let the variable Y denote an allocation of goods. The GVA works in the following manner.

1. Each agent declares a valuation function. So $v_i(Y)$ is agent i's valuation for allocation Y.
2. The GVA chooses an optimal allocation Y^* that maximizes the sum of all the agents' declared valuations.
3. The GVA announces the winners and their payment p_i:

$$p_i = \sum_{j \neq i} v_j(Y^*_{\sim i}) - \sum_{j \neq i} v_j(Y^*)$$

   where $Y^*_{\sim i}$ is the allocation that maximizes the sum of all agents' valuations assuming that agent i did not participate.

Under the usual assumption that each agent has quasilinear preferences $u_i(Y) = v_i(Y) - p_i$, the utility of bidder i in the GVA is

$$u_i(Y^*) - p_i = v_i(Y^*) + \sum_{j \neq i} v_j(Y^*) - \sum_{j \neq i} v_j(Y^*_{\sim i}).$$

The GVA has several nice properties for rational agents. First, if the agents have quasilinear preferences, the GVA is incentive compatible. The dominant strategy for rational agents is to bid their true valuations for the bundles of items. Second, the GVA is Pareto efficient. There is no other way to allocate the items (and compute payments) that would make some agent better off without making some other agent worse off. Finally, it is individually rational for agents to participate. An agent's utility obtained from participating in the GVA is never lower than if it had not participated (i.e., the agent will never end up paying more for its bundle of items than its true valuation for the bundle).

2.2.2 Example

We now provide an example to illustrate how the GVA works. Let there be two agents, agent 1 and agent 2, and let there be two items, g_1 and g_2. Agents can bid on either item or on the bundle $\{g_1, g_2\}$. An agent's bid is represented by a tuple: (a bid for g_1, a bid for g_2, a bid for $\{g_1, g_2\}$ where the bids are XOR'ed together). Suppose the agents bid as follows

- Agent 1's bid: (20, 5, 25)
- Agent 2's bid: (10, 15, 30)

The GVA allocates g_1 to agent 1 and g_2 to agent 2 since this allocation maximizes the sum of the agents' valuations. The amount that each agent pays is computed as follows. If agent 1 did not bid, then $\{g_1, g_2\}$ would have been allocated to agent 2 whose valuation for this bundle is 30. When g_1 is allocated to agent 1, agent 2's valuation is only 15 since it receives g_2. Therefore, agent 1's payment is calculated as $30 - 15 = 15$ and its utility is $20 - 15 = 5$. Agent 2's payment is $25 - 20 = 5$ and its utility is $15 - 5 = 10$.

3. COMPUTATIONALLY LIMITED AGENTS

To participate in an auction, agents need to determine valuations for the items being auctioned. The question becomes: how are these valuations derived? In this paper we focus on situations where agents do not simply know their own valuations. Rather, they have to allocate their computational resources in order to compute the valuations. We present two models where agents use computation to determine their valuations.

3.1 Models of Computation

3.1.1 Model 1: Computation improves the valuations

In our first model, computation increases the agents' valuations. As the agent computes longer, the agent finds better and better ways of using the items. Therefore, the agent may be willing to place a better bid.

An example is a procurement auction where agents are carrier companies bidding on a delivery task which consists of delivering a parcel from one location to another. The agent who submits the lowest bid wins and its utility is the bid amount, b_i, that it gets paid minus its cost, c_i, of performing the delivery. As the agent computes on the problem of how to deliver the parcel, it can obtain better (less costly) vehicle routing solutions. It might then want to modify its bid, decreasing the bid so as to increase the likelihood of winning the auction.

This model takes the viewpoint that the agent has to have a solution ready for how to use the items by some time. For example, it may be that the trucks have to be dispatched right after the auction closes, so there is no time remaining to compute a better vehicle routing solution.

3.1.2 Model 2: Computation refines the agent's beliefs

Alternatively, computation can *refine* an agent's valuation, (see, for example [Sandholm, 1993]). The agent can maintain a distribution or bounds on the valuation. Additional computation can then refine the agent's beliefs, causing the distribution of the value to shift up *or down*. The support of the distribution can also shrink. This model takes the viewpoint that the agent can compute the actual solution of how it will use the items *after* winning the auction.

3.2 The Role of Deadlines

If agents have infinite computational resources (and complete algorithms) then they would be able to compute their true valuations and execute the equilibrium bidding strategies for rational agents. However, in real systems, this is not the case. Agents have limitations on their deliberation resources. We consider the setting where each agent has some fixed amount of time to compute and a computer of finite speed. Let agent i's deadline of when it has to stop computing be d_i. It can allocate its computation in any way it sees fit, either computing on some or all of its own valuation problems (one for each bundle), or using its resources to compute on other agents' problems in order to better tailor its own bids for the auction.

The second deadline in the model is the time, T, when the auction closes. Agents' personal deadlines, d_i, may occur before, after, or at time T.

3.3 Normative Control of Deliberation

Agents must decide how to allocate their limited deliberation resources. In single–item settings they must decide whether to compute only on their own valuation problem in order to obtain the best valuation possible, or whether to use some of their resources to compute on the other agents' valuation problems. In combinatorial auctions the agents' decisions become much more complex. Agents must decide whether to devote any computation to their opponents' problems. However, they also have multiple valuation problems of their own (possibly $2^m - 1$ if there are m items) on which they have to decide how to allocate their resources.

The agents have statistical performance profiles that describe how computation changes the valuations. Each agent uses this information to decide how to allocate its computation at every step in the process, based on the results of its computation so far.

There has been much work on performance profile based deliberation control [Zilberstein and Russell, 1996; Boddy and Dean, 1994; Horvitz, 1987; Hansen and Zilberstein, 2001]. To represent the performance profiles we use a tree structure [Larson and Sandholm, 2000]. The advantage of this approach is that it allows optimal conditioning on results of execution so far, and can condition on the actual problem instance.

We index the problem by i and g where i is an agent and g is an item (or bundle of items) in the auction. For each g and i there is a performance profile tree, $\mathcal{T}_i^g$, representing the fact that the agent can use different algorithms on different problems as well as the fact that the performance profiles may be conditioned on (features of) the problem instance. Figure 1 exemplifies one such tree. We assume that all performance profile trees are common knowledge. Agent i can compute on agent j's valuation for item g, using the performance profile $\mathcal{T}_j^g$ to guide its computation. However, this computation would not change agent i's valuation for good g. The most practical model is the special case where the

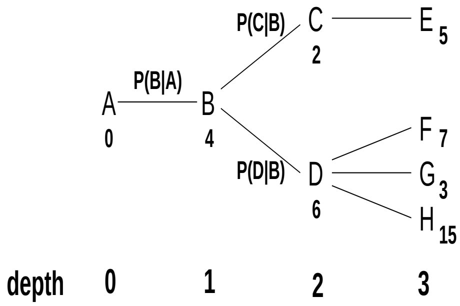

Figure 1: *One agent's performance profile tree for one valuation problem.*

agents' have the same performance profiles, i.e., $\mathcal{T}_i^g = \mathcal{T}_j^g$ for all agents i and j. This models a real world setting where the agents have been solving similar problems in the past, and have been improving their algorithms for those problems resulting in comparable algorithms among the agents.

Each depth of the tree corresponds to an amount of time t spent on running the algorithm on that valuation problem. Each node at depth t of the tree represents a possible valuation quality, $v_i^g(t)$, obtained by running t time steps on agent i's valuation problem for item g. There may be several nodes at a depth since the computation may result in different valuations depending on the problem instance or, in the case of stochastic algorithms, also on random numbers.

Each edge in the tree is associated with the probability that the child is reached in the next computation step given that the parent has been reached. This allows one to compute the probability of reaching any particular future node in a tree given any node, by multiplying the probabilities on the path between nodes. If there is no path, the probability is 0.

We denote by $\text{time}(n)$ the depth of the node n in the performance profile tree. In other words, $\text{time}(n)$ is the number of computational steps used to reach node n. We denote by $V(n)$ the value of node n.

In practice it is unlikely that an agent knows the valuation for every time allocation without actually doing the computation. Rather, there is uncertainty about how the valuation improves over time. A performance profile tree allows one to capture this uncertainty. The tree can be used to determine $P(v_i^g|t)$ denoting the probability that running the algorithm for t time steps produces a solution of value v_i^g.

We classify trees as either being deterministic or stochastic. In deterministic trees there is no uncertainty as to what value will be obtained after computing for t time steps. Deterministic trees are branches. For stochastic trees, as illustrated in Figure 1, there is uncertainty as to what value will be obtained after computing for t time steps.

The performance profile tree supports conditioning on the path of valuation quality so far. The performance profile tree that applies given a path of computation so far is simply the subtree rooted at the current node n. This subtree is denoted by $\mathcal{T}(n)$. If an agent is at a node n with value v, then when estimating how much additional deliberation would increase the valuation, the agent need only consider paths that emanate from node n. The probability, $P_n(n')$, of reaching a particular future node n' in $\mathcal{T}(n)$ is simply the product of the probabilities on the path from n to n'. The expected valuation after allocating t more time steps to the problem, if the current node is n, is

$$\sum P_n(n') \cdot V(n')$$

where the sum is over the set $\{n'|n'$ is a node in $\mathcal{T}(n)$ with depth $t\}$. This is specially useful for an agent, i, whose deadline, d_i is after the bidding deadline. It may want to estimate its final valuation and use this information in its bid formation.

Agents store the results of their deliberation actions at each time step in a state of deliberation.

DEFINITION 1. *The* **state of deliberation** *of agent i at time t is*

$$\theta_i(t) = \langle n_j^g \rangle_{j \in A}^{g \in G}$$

where A is the set of agents participating in the auction, G is the set of bundles being auctioned, and $\sum_{g,j} \text{time}(n_j^g) = t$.

Note that if there are M items then there are $2^M - 1$ bundles.

4. BIDDING STRATEGIES AND STRATEGIC COMPUTATION

A strategy for an agent in our model is composed of two interrelated components – the deliberation strategy and the bidding strategy. Let A be the set of agents participating in the auction and G be the set of bundles. Let $\text{Act}(A, G)$ be the set of deliberation actions where $\text{act}_i^g \in \text{Act}(A, G)$ is the action of taking one computation step on agent i's problem for bundle g.

DEFINITION 2. *A* **deliberation strategy** *for agent i with deadline d_i is*

$$S_i^D = (\sigma_i(t))_{t=0}^{d_i}$$

where

$$\sigma_i(t) : \theta_i(t) \rightarrow \text{Act}(G, A).$$

That is, at each time step, the deliberation strategy specifies which valuation problem should be computed on.

Bidding strategies depend on the auction mechanism.

DEFINITION 3. *A* **bidding strategy** *for agent i in a* single–shot *auction which closes at time T is*

$$S_i^B : \theta_i(T) \rightarrow \Re^{2^M - 1}$$

where M is the number of items being auctioned. A bidding strategy for agent i in a sequential *auction which closes at time T is*

$$S_i^B = (s_i(t))_{t=0}^{T}$$

where

$$s_i(t) : \theta_i(t) \times B_i(t) \rightarrow \Re^{2^M - 1}.$$

The set, $B_i(t)$, contains all bids that agent i has observed up to time t.

Finally, it is possible to formally define a strategy.

DEFINITION 4. *A* **strategy** *for agent i is*

$$S_i = (S_i^D, S_i^B).$$

4.1 Roles of Deliberation

Deliberation plays several strategic roles in auction settings. Agents can use their computational resources in different ways. First, agents can deliberate on their own valuation problems in order to obtain better valuations. In a single item setting, where each agent has only one valuation problem for the single good being auctioned, this can be quite straightforward. However, in multi–item auctions, each agent can have several valuation problems and must decide how to spread its computational resources among the problems. Second, agents can also deliberate on their opponents' problems in an attempt to gather information about the bids the opponents will submit. This information can be used by the agent in its bid formation and in allocating its computational resources.

We make a distinction between these two types of deliberation. The first we simply call deliberation. The second we call *strategic deliberation*.

DEFINITION 5. *If an agent i uses part of its deliberation resources to compute on another agent's valuation problems, then agent i is performing* **strong strategic deliberation**.

DEFINITION 6. *If an agent i does not actually use its deliberation resources to compute on another agent's valuation problems, but does use information from the opponents performance profile to aid in counterspeculation, then agent i is performing* **weak strategic deliberation**.

In strong strategic deliberation the agent actually uses its own deliberation resources to compute on an opponent's problem, thus having less resources for its own valuation problems. In weak strategic deliberation, an agent does not actually use its computational resources to compute on an opponent's valuation problem, but instead will form its deliberation (and thus bidding) strategies based on information it obtains by just examining the opponent's performance profiles. Ideally, neither form of strategic deliberation would be present in an auction. However, strong strategic deliberation is the least desirable since agents not only counterspeculate on other agents strategies, but actually use their own limited resources in the process, leading to less computation time (and therefore worse valuations) on the actual problems that will end up as part of the solution that is executed. In economic terms, strong strategic deliberation generally decreases Pareto efficiency.

5. RESULTS

We divide the results into two sections. The first section discusses single–item auctions and the second section discusses multi–item auctions. *All the results hold for both models of deliberation (computing to increase the valuations and computing to remove uncertainty from valuations) unless otherwise specified.*

5.1 Single–Item Auctions

For single–item auctions, the four common auction types (first–price sealed–bid, English, Dutch, and Vickrey) separate into two groups based on whether in equilibrium agents will perform strong strategic computation, or whether no strategic computation, strong or weak, will occur.

We get the following two theorems about the possibility of strong strategic computation.

THEOREM 1. *If agents have free but limited computational resources, in Nash equilibrium strong strategic computation can occur in first–price sealed–bid auctions.*

THEOREM 2. *If agents have free but limited computational resources, in Nash equilibrium strong strategic computation can occur in Dutch auctions.*

The proofs are omitted due to space constraints.

On the other extreme, there are auction mechanisms where no strategic computation of any form occurs in equilibrium.

THEOREM 3. *If agents have free but limited computational resources, agents have dominant strategies with no strategic computation (weak or strong) for English auctions.*

THEOREM 4. *If agents have free but limited computational resources, agents have dominant strategies with no strategic computation (weak or strong) for Vickrey auctions.*

The proofs are omitted due to space constraints.

In auction mechanisms where there were dominant bidding strategies for rational agents (English and Vickrey auctions) there is no strategic computation of any form. However, in equilibrium strong strategic computation occurs in auctions where there were no dominant strategies for rational agents (first–price sealed–bid and Dutch auctions). It is tempting to conclude that in auctions where there are dominant strategies for rational agents, no strategic computation occurs in equilibrium for bounded rational agents. However, that conclusion is premature, as will be illustrated in the next section.

5.2 Combinatorial Auctions

In combinatorial auctions, agents have a complicated decision task. They have several possible valuation problems of their own, across which they have to decide how to split their computational resources on. Ideally, the agents could simply ignore the other bidders and focus solely on their own valuation problems. However, we show that agents can gain both by counterspeculating their opponents and even by computing on some of their opponents' valuation problems.

THEOREM 5. *Assume that the agents have free but limited computational resources and the performance profiles are stochastic. In a generalized Vickrey auction, an agent may or may not have a dominant strategy. Even if the agent has a dominant strategy, that strategy can involve strong strategic deliberation.*

Proof: By example. Let there be two agents, α and β, and three items, $g1$, $g2$, and $g3$. The performance profiles for the agents' valuation problems are in Figure 2. Valuation problems that remain zero no matter how much computation is allocated to them are not shown.

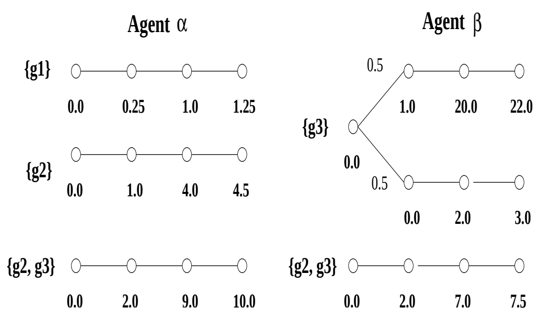

Figure 2: *Performance profiles for agents α and β. There is uncertainty in agent β's valuation for bundle $\{g3\}$.*

Assume that both agents' deadlines, d_α and d_β, occur at $t = 3$. Assume, also, that the auction closes at $T = 3$. Agent β has a dominant strategy. In the first time step it computes on the valuation for $\{g3\}$. If $v_\beta^{g3}(1) = 1.0$ then it computes two more time step on $\{g3\}$ to obtain a valuation of 22.0. At time T it would bid its true valuation for all bundles. If $v_\beta^{g3}(1) = 0.0$ then it performs two computational steps on $\{g2, g3\}$ and obtains a valuation $v_\beta^{\{g2,g3\}}(2) = 7.0$. At time T it would bid its true valuation.

Agent α's best response is to compute the first time step on agent β's valuation problem for $\{g3\}$ (i.e. perform strong strategic computation). If after one time step, agent α determines that $v_\beta^{\{g3\}}(1) = 1.0$ then agent α computes two time steps on its own valuation problem for $\{g1\}$. Otherwise it computes two steps on its own valuation problem for $\{g2, g3\}$.

Agent α realizes that if after one computation step, $v_\beta^{\{g3\}}(1) = 1.0$ then agent β will continue computing on the problem, obtain a valuation of 22.0, and include it in its bid. Since in any optimal allocation, the item $g3$ will be awarded to agent β, agent α could never be awarded any bundle that contains $g3$. Therefore it is better off computing on the valuation problem for $\{g1\}$ and bidding its true valuation. However, if $v_\beta^{\{g3\}}(1) = 0.0$ then agent α knows that it can win the bundle $\{g2, g3\}$ and so computes two steps on the valuation problem.

The expected utility for each agent can be determined:

$$u_\alpha = \frac{1}{2}(1.0 - 0.0) + \frac{1}{2}(9.0 - 7.0) = 1.5$$
$$u_\beta = \frac{1}{2}(22.0 - 0.0) + \frac{1}{2}(0) = 11.0$$

□

THEOREM 6. *Assume that the agents have free but limited computational resources and the performance profiles are all deterministic (each performance profile tree is a branch). In a generalized Vickrey auction an agent may or may not have a dominant strategy. Only weak strategic computation can occur in Nash equilibrium.*

Proof:

If a performance profile for agent i's valuation problem of bundle g is deterministic, there is no need for other agents to use their computational resources on the problem (i.e., strongly strategically compute). They need merely to check the performance profile itself to see the value for each time allocation, and then use this information to counterspeculate agent i.

Bundles	Agent α's Valuations	Agent β's Valuations
$\{g1\}$	4	0
$\{g2\}$	1	0
$\{g3\}$	0	0
$\{g1, g2\}$	0	0
$\{g1, g3\}$	6	0
$\{g2, g3\}$	0.5	12
$\{g1, g2, g3\}$	2	0

Table 1: *Values obtained if one step of computation is spent on each valuation problem for agents α and β.*

Weak strategic computation can occur. Assume there are two agents, α and β, and three goods, $g1$, $g2$, and $g3$. Each agent is allowed only one computation step and then they must bid, i.e., d_α, d_β, $T = 1$. Initially the valuations for both agents for all bundles is zero. The valuations obtained after one computation step is allocated to each problem are listed in Table 1.

The agents must decide how to use their single computation step. Agent β's dominant strategy is to compute on $\{g2, g3\}$ and to bid its true valuation. If agent α did not perform weak strategic computation, and thus did not counterspeculate agent β, it would want to compute one step on bundle $\{g1, g3\}$ and bid its true valuation. However, agent α would not be awarded $\{g1, g3\}$ since it shares item $g3$ with the bundle $\{g2, g3\}$ which will be allocated to agent β. Therefore agent α's utility would be 0 and its single computation step wasted. If agent α performs weak strategic computation then its best response is to compute one step on the valuation problem for $\{g1\}$ and bid its true valuation. Agent α's utility is 4 and agent β's utility is 12. □

Theorems 5 and 6 show that the generalized Vickrey auction loses its dominant strategy property when agents are computationally limited.

6. OTHER MODELS OF BOUNDED RATIONALITY

In this paper we use a model of bounded rationality where agents have free but limited deliberation resources. This model is key to the results. If a different model is used then the results may change. For example, under a model where there is unlimited computation but each computation step costs the agent some amount c, Theorem 4 no longer holds. Instead, Sandholm showed the following:

THEOREM 7. *Let computation actions be costly. Then, in a single–item Vickrey auction with uncertainty about an agent's valuation, a risk neutral agent's best deliberation action can depend on the other agents (i.e. weak strategic computation may occur) [Sandholm, 2000].*

In fact, even the English auction loses its dominant strategy property. In recent work the following has been shown:

THEOREM 8. *Let computation actions be costly. Then, in an English auction, if more than one bidding agent has a stochastic performance profile for its valuation problem, then strong strategic deliberation can occur in Nash equilibrium [Larson and Sandholm, 2001].*

This has repercussions for mechanism design for bounded rational agents in general. How the agents rationality is bounded should be incorporated into mechanism design in order to guarantee desirable properties.

7. RELATED RESEARCH

In this section we give a brief overview of some related research. Game theory is a useful tool to help in the design of bidding agents for auctions. There is a vast literature describing *rational* agents' optimal bidding strategies in different types of auctions [Milgrom, 1989]. However, economic models for *bounded rational* agents have often been more descriptive rather than prescriptive [Simon, 1955; Rubinstein, 1998]. In order to provide a prescriptive model, Larson and Sandholm proposed incorporating deliberation actions into agents' strategies in order to analyze, game theoretically, bounded rational agents in a 2–agent bargaining game [Larson and Sandholm, 2000].

In auctions there has been work on both bounded rational bidding agents and mechanisms. For bounded rational bidding agents, Sandholm noted that under a model of costly computation, the dominant strategy property of Vickrey auctions fails to hold [Sandholm, 2000]. Instead, an agent's best deliberation action can depend on the other agents. In recent work auction settings where agents have hard valuation problems have been studied [Parkes, 1999]. Auction design is presented as a way to simplify the meta–deliberation problems of the agents', providing incentives for the "right" agents to deliberate for the "right" amount of time. A costly computation model where agents compute to refine their valuations is used. However, situations where agents' may compute on each others' problems in order to refine their bids are not considered, nor are combinatorial auctions, where agents' have to select which of their own valuation problems and which of their opponents' valuation problems to compute on, studied.

There has also been recent work on computationally limited mechanisms. In particular research has focused on the generalized Vickrey auction and investigates ways of introducing approximate algorithms to compute outcomes without loosing the incentive compatibility property [Nisan and Ronen, 2000; Kfir-Dahav *et al.*, 2000; Lehmann *et al.*, 1999]. These methods still require that the bidding agents compute and submit their valuations.

8. CONCLUSION

Auctions provide efficient and distributed ways of allocating goods and tasks among agents, and have become a prevalent form of electronic marketplaces online. For rational agents, bidding strategies for auctions have been well studied in the game theory literature. However, software agents participating in auctions are rarely fully rational. Instead they may have computational limitations which curtail their ability to compute valuations for the items (and bundles) being auctioned. This adds another dimension to the agents' strategies as they have to determine not only the best bid to submit, but how to use their computational resources in order to determine their valuations and also to gain information about the valuations of the other agents participating in the auction.

We introduced the concepts strong and weak strategic deliberation. In strong strategic deliberation agents use part of their deliberation resources to compute on other agents' valuation problems. In weak strategic deliberation agents do not actually compute on others' valuation problems, but do use information from the other agents' performance profiles to aid in counterspeculation. We categorize single–item auctions based on whether agents' strategies will include computing on opponents' valuation problems or not, given our model of bounded rationality. The results are summarized in Table 2. In combinatorial auctions the computation of valuations becomes very complex. Agents must decide on which subset of valuations to compute on. This decision may depend on how other agents are using their own computational resources. Theorems 5 and 6 show that the generalized Vickrey auction loses its dominant strategy property when agents are computationally limited. Finally, we note that the model of bounded rationality used is important when talking about resource bounded agents' strategies. A different model may change the strategies substantially.

The results for combinatorial auctions are disheartening for resource bounded agents, as they are faced with a complicated decision problem of how to allocate their limited resources. Future work should focus on designing protocols which would simplify this decision problem while maintaining other nice properties of the generalized Vickrey auction.

	Auction mechanism	Counterspeculation by rational agents?	Strategic deliberation?	
			Weak deliberation	Strong deliberation
Single item	Vickrey	no	no	no
	English	no	no	no
	First Price	yes	yes	yes
	Dutch	yes	yes	yes
Multiple items				
	GVA (deterministic performance profiles)	no	yes	no
	GVA (stochastic performance profiles)	no	yes	yes

Table 2: ***A summary of when strategic computation does and does not occur when agents have free but limited deliberation.***

Further study should also include a full analysis of bidding and deliberation strategies under other models of bounded rationality, followed by the design of different auction protocols which take into account agents limitations, and lead to as Pareto efficient as possible outcomes.

9. REFERENCES

[Boddy and Dean, 1994] Mark Boddy and Thomas Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67:245–285, 1994.

[Hansen and Zilberstein, 2001] Eric Hansen and Shlomo Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126:139–157, 2001.

[Horvitz, 1987] Eric J. Horvitz. Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of Third Workshop on Uncertainty in Artificial Intelligence*, pages 429–444, Seattle, Washington, July 1987. American Association for Artificial Intelligence.

[Kfir-Dahav *et al.*, 2000] Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Mechanism design for resource bounded agents. In *ICMAS 2001*, 2000.

[Kirkpatrick *et al.*, 1983] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[Larson and Sandholm, 2000] Kate Larson and Tuomas W Sandholm. Deliberation in equilibrium: Bargaining in computationally complex problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 48–55, Austin, TX, 2000.

[Larson and Sandholm, 2001] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *To appear in Theoretical Aspects of Rationality and Knowledge (TARK VII)*, Siena, Italy, July 2001.

[Lehmann *et al.*, 1999] Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 96–102, Denver, CO, November 1999.

[Milgrom, 1989] Paul Milgrom. Auctions and bidding: A primer. *Journal of Economic Perspectives*, 3(3):3–22, 1989.

[Nisan and Ronen, 2000] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 242–252, Minneapolis, MN, 2000.

[Parkes, 1999] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[Rubinstein, 1998] Ariel Rubinstein. *Modeling Bounded Rationality*. MIT Press, 1998.

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

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

[Simon, 1955] Herbert A Simon. A behavorial model of rational choice. *Quarterly Journal of Economics*, 69:99–118, 1955.

[Varian, 1995] Hal Varian. Mechanism design for computerized agents. In *Usenix Workshop on Electronic Commerce*, New York, July 1995.

[Zilberstein and Russell, 1996] Shlomo Zilberstein and Stuart Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82(1–2):181–213, 1996.