Experiments on deliberation equilibria in auctions

Kate Larson and Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{klarson,sandholm}@cs.cmu.edu

Abstract

Auctions are useful mechanisms for allocating items (goods, tasks, resources, etc.) in multiagent systems. The bulk of auction theory assumes that the bidders know their own valuations for items a priori. *However, in many applications the bidders need to expend significant effort (computational or information gathering) to determine their valuations. This leads to the possibility of complex strategic behavior as agents may have incentive to not only use resources to determine their own valuations, but may also attempt to determine the valuations of competing bidders. It has been shown that given any commonly used auction protocol, it is theoretically possible to construct special instances such that this* strategic deliberation *occurs. We study the prevalence of strategic deliberating in order to determine whether it is merely of theoretical interest or if it is an issue which arises in practice. Using anytime algorithms and performance-profile-tree-based deliberation control in different real-world problem domains, and the deliberation equilibrium solution concept, we show that strategic deliberation does occur in practice whenever there is a certain amount of asymmetry between the agents.*

1. Introduction

In many AI and multiagent applications, computational limitations are simply a necessary evil that has to be dealt with. The realities of limited computational resources and time pressures caused by real-time environments mean that agents are not always able to optimally determine their best decisions and actions. The field of artificial intelligence has long searched for useful techniques for coping with this problem. Herbert Simon advocated that agents should forgo perfect rationality in favor of limited, economical reasoning. His thesis was that "the global optimization problem is to find the least–cost, or best–return decision, net of computational costs" [15].

Considerable work in AI has focused on developing normative models that prescribe how a computationally limited agent should behave [12]. This is a highly nontrivial undertaking, encompassing numerous fundamental and technical difficulties (see, for example [1, 4, 13]). While simplifications can be acceptable in single-agent settings, as long as the agent performs reasonably well, any deviation from full normativity can be catastrophic in multiagent systems. If a system designer can not guarantee that a strategy (including the computing actions) is the best strategy that an agent can use, there is a risk that the agent will be motivated to use a different strategy. Since the strategy choice that one agent makes can influence the strategic decisions of other agents, this may result in an outcome which is far from the predicted one.

Recently in the multiagent systems community, there has been interest in studying and understanding the strategic ramifications created by limited computing resources in different market mechanisms. Sandholm noted that if there is a cost associated with the act of determining the value for an item, then an agent may no longer have a dominant strategy in a Vickrey auction [14]. Auction design has also been presented as a way to simplify the meta-deliberation problems of the agents by providing incentives for the "right" agents to compute for the "right" amount of time [11]. In these papers it was assumed that agents may only compute on their own valuation problems. In recent work, we relaxed this assumption, allowing agents freedom to use their computing resources, coupled with fully normative deliberation controllers, on any valuation problem—including the problems of competing agents [7]. Using the deliberation equilibrium as the game-theoretic solution concept, where agents' computing actions are explicitly included in the strategies, we found that for all commonly used auction protocols, it was possible to construct instances such that agents had incentive to use some of their deliberation resources on other agents' valuation problems. We coined the term *strategic deliberation* to describe this phenomenon. However, this work was highly theoretical, with no experimental backing

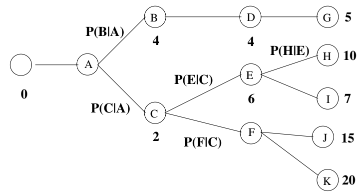

Figure 1. ***A performance profile tree.***

and it was unclear as to whether, in practice, this strategic behavior occurs. This paper addresses this criticism.

In this paper, using real-world data obtained from black-box domain solvers, we do a full game-theoretic analysis and classify the deliberation equilibria which occur when two agents participate in reverse Vickrey auctions. We examine different criteria which affect the strategic behavior of agents, including the role of reserve prices, and the importance of asymmetry.[1]

2. Decision theoretic deliberation control

We begin by providing a short overview of deliberation control methods. We assume that agents have algorithms that allow them to trade off computational resources for solution quality. In particular, we assume that agents have *anytime algorithms*, that is, algorithms that improve the solution over time and return the best solution available even if not allowed to run to completion [5, 1]. Most iterative improvement algorithms and many tree search algorithms (such as branch and bound) are anytime algorithms. The anytime algorithm can also be used as a paradigm for information gathering.

While anytime algorithms are models that allow for the trading off of computational resources, they do not provide a complete solution. Instead, anytime algorithms need to be paired with a meta-level deliberation controller that determines how long to run the anytime algorithm, that is, when to stop deliberating and act with the solution obtained. The deliberation controller's stopping policy is based on a *performance profile*: statistical information about the anytime algorithm's performance on prior problem instances. This helps the deliberation controller project how much (and how quickly) the solution quality would improve if further computation were allowed. Performance profiles are usually generated by prior runs of the anytime algorithm on different problem instances.

In order to capture all of the information available for making stopping decisions, the *performance profile tree (PPTree)* representation was introduced [7]. In a PPTree, the nodes represent solution types at given time points, and each edge carries the probability that the child node is reached given that the parent was reached. Figure 1 exemplifies one such tree. A PPTree can support conditioning on any and all features that are deemed to be of importance for making stopping decisions. The nodes can hold information about solution quality and any other solution feature that may be important.

A key aspect of the PPTree is that it automatically supports conditioning on the *path* (of the feature vector) so far. The performance profile tree that applies given a path of computation is the subtree rooted at the current node n. This subtree is denoted by $\mathcal{T}(n)$. If an agent is at node n with solution quality $V(n)$, then when estimating how much additional computing would increase the solution quality, 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 solution quality 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 $\mathcal{T}_i(n)$ which is reachable in t time steps$\}$.

3. Game theory and auctions

A game consists of a set of agents, I ($|I| = n$), a set of actions available to each agent, i, a set of histories for each agent i (sequences of actions taken by agent i), and a set of outcomes, O. Each agent is free to choose which strategy to use where a strategy is a contingency plan that determines what action the agent will take at any given point in the game. A *strategy profile*, $s = (s_1, \ldots, s_n)$, is a vector that specifies one strategy for each agent i in the game. We use the notation $s = (s_i, s_{-i})$ to denote a strategy profile where agent i's strategy is s_i and $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. The strategies in a strategy profile determine how the game is played out, and determine the outcome, $o(s) \in O$. Each agent i tries to choose a strategy s_i, so as to maximize its utility, which is given by a utility function $u_i : O \mapsto \mathbb{R}$.

A strategy is said to be dominant if it is an agent's strictly best strategy, against any strategy the other agents might play. If in strategy profile $s^* = (s^*_1, \ldots, s^*_n)$, each strategy, s^*_i, is a dominant strategy for agent i, then the strategy profile s^* is a *dominant strategy equilibrium*. Agents do not always have dominant strategies, and so dominant strategy equilibria do not always exist. Instead, the Nash equilibrium solution concept is used.

Definition 1 *A strategy profile s^* is a* Nash equilibrium *if no agent has incentive to deviate from its strategy given that the other players do not deviate. Formally, $\forall i \; \forall s'_i \; u_i(o(s^*_i, s^*_{-i})) \geq u_i(o(s'_i, s^*_{-i}))$.*

1 In the paper we use the terms "deliberate", "compute", and "gather information" interchangeably.

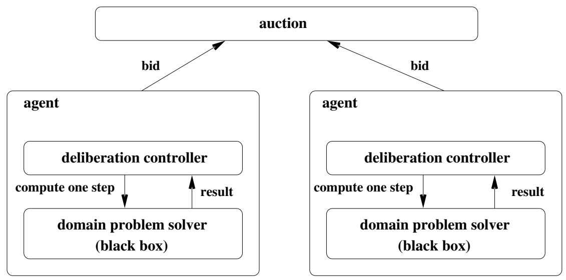

Figure 2. ***An auction with two bidding agents. In order to submit a bid, each agent needs to first (approximately) determine its valuation.***

Auctions are stylized markets and have been well studied in the game theory and economics literature [6]. While there are many types of auctions, in this paper we focus on the private-value (reverse) Vickrey auction. In a Vickrey auction, every agent submits a sealed-bid to an auctioneer. The highest bidder wins the item and must pay the amount of the second highest bid. The Vickrey auction has nice game-theoretic properties, since each agent has a dominant strategy which is to submit a bid equal to its own value of the item.

4. Deliberation equilibria

The bulk of auction theory assumes that bidders know their own valuations for items *a priori*. However, in many applications the bidders need to expend significant effort (computational or information gathering) to determine their valuations. In this section we describe how to incorporate bidding agents' deliberation actions into their strategies and present the *deliberation equilibrium* solution concept.

A strategy for an agent is composed of two interrelated components — the deliberating component and the bidding component. What an agent bids depends on the solutions it has obtained for the valuation problems, and the problem on which an agent deliberates depends partially on how it is planning on bidding, and how other agents bid. Figure 2 illustrates this relation.

In our model the game is divided into time periods. In each time period, an agent, i, is allowed to either execute a deliberation action, $\mathrm{comp}_i(j)$, or a deliberation action followed by a bidding action, $(\mathrm{comp}_i(j), b_i)$. A deliberation action is the act of deliberating for one time step on some agent j's valuation problem. An agent may also decide to take a null deliberating action, $\emptyset^C$, by not deliberating on any problem. In our (reverse) Vickrey auction, each agent, i, submits a single bid, $b_i \in \mathbb{R}$.

In general a history for agent i at time period t, $h_i(t)$, is a list of all actions agent i has taken. We augment this definition of a history to include the cost incurred by the agent at time t, $\mathrm{cost}_i(t)$, and a state of deliberating at time t, $\theta_i(t)$, *i.e.*, the deepest nodes in the PPTrees reached by the agent. We represent this augmented history by $H_i(t) = (h_i(t), \mathrm{cost}_i(t), \theta_i(t))$ and define $\mathcal{H}_i(t) = \{H_i(t)\}$.

A strategy is a mapping from history to action. In the (reverse) Vickrey auction the agents can take deliberation actions until a specified time T, when they have to submit a bid to the auctioneer. A (deliberation) strategy for an agent is defined as follows.

Definition 2 *A strategy for agent i in a (reverse) Vickrey auction with closing time T is $S_i(\sigma_i(t))_{t=0}^T$ where*

$$\sigma_i(t+1) : \begin{cases} \mathcal{H}_i(t) \mapsto \{\mathrm{comp}_i(j) | j \in I\} & \textit{if}\, t < T \\ \mathcal{H}_i(t) \mapsto \mathbb{R} & \textit{if}\, t = T \end{cases}$$

If agents' deliberation and bidding strategies are in (Nash) equilibrium then we say there exists a (Nash) *deliberation equilibrium.*

Agents can use their deliberation resources in different ways. They can deliberate on their own problems in order to obtain better valuations. They can also deliberate on their opponents' problems in an attempt to gather information about the bids that the opponents may be submitting.

Definition 3 (Strategic Deliberation) *If an agent i uses part of its deliberation resources to deliberate on another agent's valuation problem, then agent i is* strategically deliberating. *That is, a strategy, $S_i = (\sigma_i(t))_{t=0}^T$, consists of strategic deliberation if there exists a time step t and a history $x \in \mathcal{H}_i(t)$ such that $\sigma_i(t)(x) = \mathrm{comp}_i(j)$ where $j \neq i$.*

We believe that strategic deliberation is undesirable. First, it may lead to a decrease in social welfare as agents compute on problems which do not directly improve their utility. Second, deliberation control is difficult when there is only a single problem. Having to make deliberation control decisions across multiple problems may be overwhelming for the agents.

5. Experiments

Using performance profile trees as our deliberation control procedure, we conducted a series of experiments to explore the effect that limited deliberation resources has on agents' strategies. After generating performance profiles using data from different real-world application domains, we used Gambit [10], a popular solver for finding game-theoretic equilibria, to find and categorize all Nash deliberation equilibria.

5.1. Application scenarios

We conducted our experiments using data from two different scenarios; vehicle routing and single-machine manufacturing scheduling. We treated the domain problem solvers as black boxes.

5.1.1. Vehicle routing In the real-world vehicle routing problem (VRP) in question, a dispatch center is responsible for a certain set of tasks (deliveries) and has a certain set of resources (trucks) to take care of them. Each truck has a depot, and each delivery has a pickup location and a drop-off location. The dispatch center's problem is to minimize transportation cost (driven distance) while still making all of its deliveries and honoring the following constraints: 1) each vehicle has to begin and end its tour at its depot, and 2) each vehicle has a maximum load weight and maximum load volume constraint.

To generate data for our experiments, an iterative improvement algorithm was used for solving the VRP. The problem instances were generated using real-world data collected from a dispatch center that was responsible for 15 trucks and 300 deliveries. We generated two independent sets by randomly dividing the deliveries in half. To generate 1000 instances for each set used to build PPTrees, we randomly selected (with replacement) 60 deliveries.

5.1.2. Manufacturing scheduling The second domain is a single-machine manufacturing scheduling problem with sequence-dependent setup times on the machines, where the agent's objective is to minimize weighted tardiness, $\sum_{j \in J} w_j T_j = \sum_{j \in J} w_j \max(f_j - d_j, 0)$, where T_j is the tardiness of job j, and w_j, f_j, d_j are the weight, finish time, and due-date of job j.

In our experiments, we used a state-of-the-art scheduler developed by others as the domain problem solver [3]. It is an iterative improvement algorithm that uses a scheduling algorithm called Heuristic Biased Stochastic Sampling [2]. We treated the domain problem solver as a black box without any modifications.

The problem instances were generated according to a standard benchmark [9]. The due-date tightness factor was set to 0.3 and the due-date range factor was set to 0.25. The setup time severity was set to 0.25. These parameter values are the ones used in standard benchmarks [9]. Each instance consisted of 100 jobs to be scheduled. We generated two independent sets of instances by using different random number seeds.

5.2. Generating performance profiles

We used independent sets of 1000 instances per set from both application domains to generate performance profile trees for the agents. Due to limitations in Gambit [10], the software used for the game-theoretic analysis, we were required to use coarse discretizations on both solution quality and time steps. Each tree had depth two (an agent could compute for two time steps on a single problem). The solution quality was coarsely discretized in order to reduce the branching factor of the PPTrees so that the problems were feasible for Gambit. Figure 3 shows sample performance profile trees produced when the solution quality was uniformly discretized into buckets of size 25000 for scheduling and size 10000000 for the trucking domain.[2]

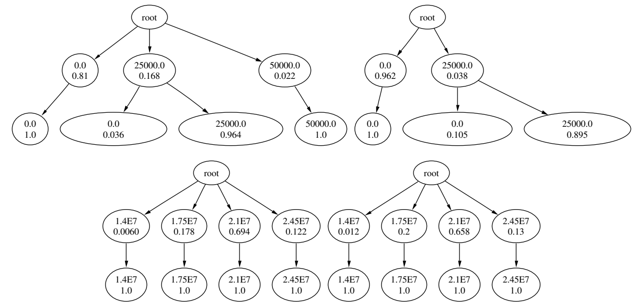

Figure 3. ***Performance profile trees for the scheduling domain (top two trees) and trucking domain (bottom two trees). Each node in a tree contains two numbers. The first number is the solution quality and the second number is the probability of reaching the node, given that its parent was reached.***

5.3. Cost functions

In our model there are costs associated with deliberating. The costs for agent i are represented by a function $c_i(t_1, t_2) = K_i^1 t_1 + K_i^2 t_2$ where t_j is the amount of time agent i deliberated on the valuation problem of agent j, and $K_i^1, K_i^2 \geq 0$ are predefined constants. The cost functions of the agents are either *symmetric* or *asymmetric*.

Definition 4 *A cost function* $c_i(t_1, t_2) = K_i^1 t_1 + K_i^2 t_2$ *is* symmetric *if* $K_i^1 = K_i^2$. *A cost function* $c_i(t_1, t_2) = K_i^1 t_1 + K_i^2 t_2$ *is* asymmetric *if* $K_i^1 \neq K_i^2$.

Symmetric costs naturally model situations where agents *compute* on problems in order to determine valuations. For example, an agent may pay some amount K for each CPU cycle used running an algorithm. Asymmetric cost functions naturally model information gathering scenarios as it is not unreasonable that there are different costs associated with gathering information from different sources.

2 We experimented with other feasible uniform discretizations. The results were all similar to the results reported in this paper. Therefore, due to space considerations we do not include them.

5.4. The reverse Vickrey auction

We provide a motivating example. A company wishes to contract out a set of 100 tasks as a sole-source contract, i.e., the entire set is allocated to one manufacturer. The company runs a reverse Vickrey auction to allocate the set to one of two possible manufacturers. Each task in the set has a deadline, and a task-specific penalty for each unit of time that the task is late. The manufacturer has to pay the total weighted tardiness to the company as a penalty. If agent i wins the auction, has obtained a valuation, v_i, by deliberating, and has incurred cost c_i while doing so, then its utility is $u_i = x - v_i - c_i$ where $x = \min(b_j, R)$ where b_j is the second lowest bid (if any) and R is the company's reserve price, i.e., the company's maximum willingness to pay to get the task set manufactured. If agent i does not win, its utility is $u_i = -c_i$.

In our experiments there are two agents, agent 1 and agent 2. Each agent has different valuation problems and thus different performance profiles. The performance profile trees are common knowledge. At each time step an agent has a choice of actions available to it: it can deliberate on its own valuation problem, on its competitor's problem, or it can choose not to deliberate. Once both agents stop deliberating (after at most two time steps), they submit bids to the auctioneer.

5.5. Representing equilibria

We denote a (deliberation) strategy by A_{iB_i} where A_i is the deliberation action taken by agent i in the first step, and B_i is the *set* of deliberation actions taken by agent i in the second step, *conditional on the results observed after the first step*. We do not denote the bidding actions since agents are motivated to always submit their value obtained by deliberating.

We represent the deliberation actions as follows: N is the action of not deliberating, O is the action of deliberating for one step on its own problem, and, Z is the action of deliberating for one step on its opponent's problem. Agents may play mixed strategies, that is, they can randomize between multiple actions. For example, at a given time step, an agent may decide to randomize between not deliberating (N) and deliberating on its own problem (O). We denote this by $m(O, N)$.

Consider the following example. Assume that there exists an equilibrium where agent 1 randomizes between deliberating on its own problem and the problem of agent 2, and then either stops deliberating or deliberates for one step on its own problem in the second stage. Agent 2 randomizes between all three actions, and then stops. This is denoted by $(m(O, Z)_{(O,N)}, m(O, Z, N)_N)$.

5.6. Symmetric cost functions

We first investigate what happens if agents have symmetric, but potentially different, cost functions. For both application domains we ran a series of reverse Vickrey auctions, varying the reserve prices and cost functions of the agents. Table 1 presents a complete taxonomy of all Nash (deliberation) equilibria in the scheduling domain, when the reserve price was taken from the set $\{25000, 50000, 100000\}$. These price choices span the interesting range for the scheduling domain. Reserve prices cap the potential utility of the agents. At the bottom end the agents' utilities are capped so that there is a potential that even if both agents deliberate on their own problems, neither agent will win the auction. At the high end the reserve price has little influence, since the cap is set high enough such that as long as agents deliberate, an agent will win the auction. The cost functions of the agents were of the form $c_i(t_1, t_2) = K_i(t_1 + t_2)$ for $K_i \in \{10, 100, 12500, 25000, 50000\}$. These value choices span the interesting range for the scheduling domain: at the bottom end the values are low enough so that deliberation on problems is a potential strategy, while at the top end there is the potential that the deliberating cost is higher than any possible utility that could be achieved.[3]

There are several observations. First, we did not observe any equilibria where strategic deliberation occurred. Second, the cost functions of the agents influenced their strategic behavior. For example, when $K_1 = 100$ then agent 1 always spent the first time step deliberating on its own problem. However, the action it took in the second time step depended on the actions of agent 2. Third, multiple and mixed equilibria appeared when the costs were high enough (12500), and disappeared when the costs became too high (compared to the reserve price). Finally, reserve prices influenced the deliberation equilibria. For example, the equilibria that occurred when $c_1(t_1, t_2) = 50000 \cdot (t_1 + t_2)$ and $c_2(t_1, t_2) = 25000 \cdot (t_1 + t_2)$ changed according to the reserve price. When the reserve price was $R = 25000$ then there was a single equilibrium: neither agent deliberated on any problem. When $R = 50000$, there was a different, single equilibrium. Finally, when $R = 100000$ there were multiple equilibria: two pure equilibria where one agent deliberated on its own problem for one time step while the other did not deliberate at all, and a mixed equilibrium where both agents randomized between not deliberating and deliberating on their own problem in the first step.[4]

3 We experimented with other reserve prices and cost functions. The results of these experiments were similar to the ones presented in the paper.

4 We conducted similar experiments for the vehicle routing domain. The same properties were observed. Due to space considerations we do not

R=25000	100	1000	12500	25000	50000
100	$O_N, O_{(O,N)}$	O_N, O_N	$O_{(O,N)}, N$	$O_{(O,N)}, N$	$O_{(O,N)}, N$
1000 and 12500	$N, O_{(O,N)}$	$N, O_{(O,N)}$	N, O_N $m(O,N)_N, m(N,O)_N$ O_N, N	O_N, N	O_N, N
25000 and 50000	$N, O_{(O,N)}$	$N, O_{(O,N)}$	N, O_N	N, N	N, N
R=50000					
100	$O_N, O_{(O,N)}$	O_N, O_N	$O_{(O,N)}, N$	$O_{(O,N)}, N$	$O_{(O,N)}, N$
1000 12500 and 25000	$N, O_{(O,N)}$	$N, O_{(O,N)}$	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	O_N, N
50000	$N, O_{(O,N)}$	$N, O_{(O,N)}$	N, O_N	N, O_N	N, N
R=100000					
100	$O_N, O_{(O,N)}$	O_N, O_N	$O_{(O,N)}, N$	$O_{(O,N)}, N$	$O_{(O,N)}, N$
1000 12500 25000 and 50000	$N, O_{(O,N)}$	$N, O_{(O,N)}$	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N

Table 1: ***All Nash equilibria for the scheduling domain with symmetric cost functions. The rows are different cost functions for agent 1 and the columns are different cost functions for agent 2. Each cell contains all Nash equilibria given the agents' cost functions and the reserve price R. To conserve space, we have combined rows which have identical equilibria.***

5.7. Asymmetric cost functions

We conducted experiments with asymmetric cost functions. Tables 2 and 3 present results from the vehicle routing domain, where the reserve price, R, was set to $5 \cdot 10^7$, *i.e.*, high enough so that it did not introduce additional strategic considerations, and where $K_i^j \in \{1000, 2.5 \cdot 10^6, 5 \cdot 10^6, 1 \cdot 10^7\}$. These values were chosen so as to span the interesting range for the routing domain.

We only present results where cost functions had the form $c_i(t_1, t_2) = \sum_{j=1}^{2} K_i^j \cdot t_j$ where $K_i^i \geq K_i^k$. If the cost of deliberating on a competitor's valuation problem is higher than the cost of deliberating on the agent's own problem, then clearly no strategic deliberation will occur in equilibrium.

Several interesting phenomena were observed. First, when the cost of deliberating on the other agent's valuation problem was low, strategic deliberation occurred in equilibrium. This is illustrated by Table 2. Almost every mixed Nash equilibrium involved agent 1 deliberating on the problem of agent 2 with positive probability. However, when $K_1^2 = 2.5 \cdot 10^6$, then agent 1 never deliberated on the valuation problem of agent 2. The same behavior was observed for agent 2 (see Table 3). Second, if K_i^j was high enough then no strategic deliberation occurred, irrespective of the value of K_i^i (Table 3). Third, the other agent's actions also play an important role. That is, agents' do not have dominant strategies which are parameterized by their own cost functions, instead, complex strategic behavior occurs. This can be observed from Table 3.[5]

How much asymmetry is required in order for strategic deliberation to occur? The experiments reported in Tables 2 and 3 involved gross asymmetries. It was not clear whether large differences in deliberating costs are required in order to force strategic deliberation, or whether just an ϵ-difference in the cost constants could produce interesting strategic behavior. Using the scheduling domain as our sample application, we fixed the cost function of agent 2 as $c_2(t_1, t_2) = 100 \cdot (t_1 + t_2)$ and set $c_1(t_1, t_2) = 100 \cdot t_1 + K_1^2 \cdot t_2$. We decreased K_1^2 by increments of 0.5. When $K_1^2 \geq 97.0$ the unique Nash (deliberation) equilibrium was (O_N, O_N). However, for $K_1^2 < 97.0$ the unique Nash (deliberation) equilibrium was $(Z_{(O,N)}, O_N)$. It appears as though we can conclude that gross asymmetries are not required in order for strategic deliberation to occur, however there should be more than an ϵ-difference between the costs associated with deliberating on different problems.

6. Discussion

From studying the results from both the symmetric and asymmetric cost function experiments, there are some interesting points that arise. First, in order for strategic delib-

present the results here.

5 Using PPTrees generated from data from the scheduling domain, we repeated the experiments and observed similar equilibria. Due to space limitations we do not present the results in this paper.

(1000,1000)	1000	2.5E6	5E6	1E7
1000	O_N, O_N	O_N, N	O_N, N	O_N, N
2.5E6 and 5E6	N, O_N	N, O_N $*\mathbf{m(O,Z)_{(O,N)}}, \mathbf{m(O,Z)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,Z)_{(O,N)}}, \mathbf{m(O,Z)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(Z,N)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N
1E7	N, O_N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ $*\mathbf{m(O,Z,N)_{(O,N)}}, \mathbf{m(O,Z,N)_{(O,N)}}$ $*\mathbf{m(Z,N)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N
(1000,2.5E6)				
1000		O_N, N	O_N, N	O_N, N
2.5E6		N, O_N $*\mathbf{m(O,Z)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N	N, O_N $*\mathbf{m(Z,N)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N	N, O_N $*\mathbf{m(Z,N)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N
5E6 and 1E7		N, O_N $*\mathbf{m(O,Z)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N	N, O_N $*\mathbf{m(O,Z)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N	N, O_N $*\mathbf{m(Z,N)_{(O,N)}}, \mathbf{m(O,N)_N}$ O_N, N

Table 2: ***Nash (deliberation) equilibria for the vehicle routing domain with asymmetric cost functions. The reserve price is*** $R = 5 \cdot 10^7$***. The cost function of agent 1 is*** $c_1 = K_1^1 \cdot t_1 + 1000 \cdot t_2$***. The entry in the upper left hand corner cell of each subtable specifies*** (K_1^2, K_2^1) ***The values for*** K_1^1 ***are listed in the rows, and the values for*** K_2^2 ***are listed in the columns. The subtables*** $(1000, 5E6)$ ***and*** $(1000, 1E7)$ ***were identical to subtable*** $(1000, 2.5E6)$***. To conserve space, we have combined rows which have identical equilibria.***

(2.5E6, 1000)	1000	2.5E6	5E6	1E7
2.5E6 and 5E6	N, O_N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N
1E7	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N	N, O_N $*\mathbf{m(O,N)_N}, \mathbf{m(Z,N)_{(O,N)}}$ O_N, N
(2.5E6,2.5E6)				
2.5E6 5E6 and 1E7		N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N	N, O_N $m(O,N)_N, m(O,N)_N$ O_N, N

Table 3: ***Nash (deliberation) equilibria for the vehicle routing domain with asymmetric cost functions. The reserve price is*** $R = 5 \cdot 10^7$***. The cost function of agent 1 is*** $c_1 = K_1^1 \cdot t_1 + 2.5 \cdot 10^6 \cdot t_2$***. he entry in the upper left hand corner cell of each subtable specifies*** (K_1^2, K_2^1) ***The values for*** K_1^1 ***are listed in the rows, and the values for*** K_2^2 ***are listed in the columns. The subtables*** $(2.5E6, 5E6)$ ***and*** $(2.5E6, 1E7)$ ***were identical to subtable*** $(2.5E6, 2.5E6)$***. To conserve space, we have combined rows which have identical equilibria.***

eration to occur, there must be a certain amount of asymmetry between the problems of the agents. This asymmetry may arise due to differences in cost functions, or the relative difficulty of different agents' valuation problems. For example, in a vehicle routing problem, it might be very easy to find a short route in one problem instance, yet not be so easy in a different problem instance. Even if there is no strategic deliberation (as in the symmetric cost function setting) agents' optimal strategies depend on what strategies the other agents are following: dominant strategy equilibria do not always exist. For example, there may exist situations where an agent may decide to not compute on any problem in order to avoid incurring a cost, given that a competitor intends to solve its own valuation problem.

Second, the (reverse) Vickrey auction is a very simple auction mechanism and when agents are fully rational, they have dominant strategies. In spite of its simplicity, in recent work it has been shown that the information structures which appear in the Vickrey auction and drive the theoretical strategic deliberation results, also appear in many multistage auctions such as the English auction and variants [8]. Thus, while our experiments focus only on one auction type, we believe that the conclusions are applicable across a wide range of auction mechanisms.

7. Conclusions and future research

In many auction settings it is unreasonable to assume that agents know their valuations for items that they will bid on, *a priori*. Instead, agents may need to expend considerable computational resources in order to determine their valuations before bidding. Recently there has been a body of work which has studied the strategic implications which arise when the deliberation actions (computing or information gathering actions) of agents are explicitly included in the strategies of agents. The analysis of *deliberation equilibria* in various auction settings has led to the discovery of strategic deliberation, equilibrium strategies where agents use some of their deliberation resources in order to (partially) determine the valuations of competitor agents.

This paper presents the first experimental study of deliberation equilibria in auctions. We ran reverse Vickrey auctions with bidding agents who had limited deliberation resources and were provided with fully normative deliberation control methods. We observed no strategic deliberation when agents had *symmetric* cost functions.[6] However, if the cost functions of the agents were *asymmetric* then strategic deliberation occurred in equilibrium. This supports our position that when designing electronic markets and other multiagent systems, it is of both theoretical and practical importance to consider the deliberation actions of agents.

There are several research directions that should be pursued. First, in our experiments we were limited in the number of deliberation steps agents could take. As the technology becomes available, we plan to run additional experiments where agents are allowed to deliberate more. The second research direction is to game-theoretically *design* auctions so that strategic deliberation will provably not occur in equilibrium. This would allow agents to focus solely on their own deliberation control problems, making it easier for deliberative agents to participate.

8. Acknowledgements

We would like to thank Vincent Cicirello and Stephen Smith for providing us with both the scheduler and the scheduling instance generator. 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.

References

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

[2] J. L. Bresina. Heuristic-biased stochastic search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, (AAAI'96)*, pages 271–278, 1996.

[3] V. Cicirello and S. Smith. Amplification of search performance through randomization of heuristics. In *Principles and Practice of Constraint Programming – CP 2002*, pages 124–138, 2002.

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

[5] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 111–116, St. Paul, MN, August 1988. Morgan Kaufmann, San Mateo, CA.

[6] V. Krishna. *Auction Theory*. Academic Press, 2002.

[7] K. Larson and T. Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Siena, Italy, July 2001.

[8] K. Larson and T. Sandholm. Strategic deliberation and truthful revelation: An impossibility result. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, New York, NY, May 2004. Short paper.

[9] Y. H. Lee, K. Bhaskaran, and M. Pinedo. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions*, 29:45–52, 1997.

[10] R. McKelvey, A. McLennan, and T. Turcoy. *Gambit: Software Tools for Game Theory*. The Gambit Project, 2002.

[11] D. 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.

[12] S. Russell. Rationality and intelligence. *Artificial Intelligence*, 94(1):57–77, 1997.

[13] S. Russell and E. Wefald. *Do the right thing: Studies in Limited Rationality*. The MIT Press, 1991.

[14] T. Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. Special Issue on Applying Intelligent Agents for Electronic Commerce. A short, early version appeared at the Second International Conference on Multi–Agent Systems (ICMAS), pages 299–306, 1996.

[15] H. A. Simon. *Models of bounded rationality*, volume 2. MIT Press, 1982.

6 This is not proof that it never occurs in practice. If more deliberation actions were allowed, then we may observe it experimentally.