



Monte Carlo approximation in incomplete information, sequential auction games

Gangshu Cai, Peter R. Wurman*

Computer Science, North Carolina State University, Raleigh, NC 27695-7535, USA

Available online 29 November 2003

Abstract

We model sequential, possibly multiunit, sealed bid auctions as a sequential game with imperfect and incomplete information. We develop an agent that constructs a bidding policy by sampling the valuation space of its opponents, solving the resulting complete information game, and aggregating the samples into a policy. The constructed policy takes advantage of information learned in the early stages of the game and is flexible with respect to assumptions about the other bidders' valuations. Because the straightforward expansion of the complete information game is intractable, we develop a more concise representation that takes advantage of the sequential auctions' natural structure. We examine the performance of our agent versus agents that play perfectly, agents that also create policies using Monte Carlo, and other benchmarks. The technique performs quite well in these empirical studies, although the tractability of the problem is bounded by the ability to solve component games.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Sequential auctions; Game theory; Monte Carlo sampling; Incomplete information

1. Introduction

Online auctions have rapidly permeated both business-to-business negotiation and public marketplaces. The vast number of trading opportunities and the increasingly fluid markets bolsters the need for automated trading support in the form of *trading agents*—software programs that participate in electronic markets on behalf of a user. Simple bidding tools, like eSnipe¹ and AuctionBlitz² enable bidders to automate the submission of last-second bids on

eBay. However, these tools lack the sophistication that bidders require when faced with a plethora of auctions possibly hosted at multiple auction sites.

Recently, the design of more sophisticated trading agents has attracted the attention of researchers in artificial intelligence and other related fields [4,8,21,25,29]. In most of these studies, the agents are designed for a particular marketplace and lack the flexibility to adapt to other market configurations. In this paper, we develop a more general approach to constructing trading agents based on game theory and explore its computational limitations.

We develop our technique in the context of a sequence of (possibly multiunit) auctions with a small set of identified, risk-neutral participants, each of whom wants one unit of the item for which they have an independent private value. We assume that our

* Corresponding author.

E-mail addresses: gcai@unity.ncsu.edu (G. Cai), wurman@csc.ncsu.edu (P.R. Wurman).

¹ <http://www.esnipe.com>.

² <http://www.auctionblitz.com>.

agent knows the distribution of the other agents' valuations but not their actual values. This is meant to model common procurement scenarios and may fit some markets on eBay, in which it is apparently common for a small community of expert traders to recognize each other. In both situations, the relatively small number of significant opponents creates the opportunity to directly model one's competitors.

We cast the problem as an incomplete, imperfect information game. However, the straightforward expansion of a sequence of auctions creates a game that is intractable even for very small problems, and it is beyond the capability of the current game solution software to solve for the Bayes–Nash equilibria. Thus, we construct a bidding policy through Monte Carlo sampling. In particular, we sample the opponents' valuations, assume they play perfectly, and solve the resulting imperfect information game. We accumulate the results of the sampling into a heuristic strategy for the incomplete information game.

The resulting strategy implicitly captures the belief updating associated with observing the opponents' bids in earlier auctions. Underlying this work is the assumption that the information we gain about the other bidders can be used to improve play in later stages of the game. In particular, our observations of a bidder's actions in previous auctions should affect our belief about her valuation. For example, if we notice that Sue has placed bids at high values in previous auctions but not yet won anything, we are more likely to believe that Sue has a high valuation which may influence how we should bid in future auctions.

The primary motivation of this line of work is to explore the potential benefits and the practical limitations of this approach. We find that the straightforward expansion of the imperfect information game cannot be solved directly by current game solvers (e.g., GAMBIT³). Thus, we develop methods to take advantage of the sequential structure that greatly reduces the space required to represent the game. Although this decomposition enables us to solve larger games, GAMBIT's ability to solve the decomposed games remains a bottleneck.

³ The GAMBIT toolset is a software program and set of libraries that support the construction and analysis of finite extensive form and normal form multiplayer games. See <http://www.hss.caltech.edu/gambit/Gambit.html>.

In Section 2, we formalize our model of the sequential auction scenario and set up the game theoretic analysis. Section 3 describes how we leverage the substructure to significantly decrease the amount of computation necessary to solve the game. In Section 4, we use Monte Carlo sampling to generate a heuristic bidding policy for our agent. Section 5 presents our empirical results including comparisons between our heuristic policy and perfect play in markets that contain both single-unit and multiunit auctions. Section 6 develops the relationship between our approach and the mathematics underlying sequential equilibria. We present related work in Section 7 and then conclude.

2. Model

Consider an agent, i , that has the task of purchasing one item from a sequence of auctions, K . Let c be the number of auctions, and k be an individual auction. We refer to the collection of auctions as the *market-place*. Individual auctions may offer multiple units and differ in the manner in which they form prices. The specification of the order and rules of the collection of auctions is the *market configuration*.

Let $q(k)$ be the number of units offered in auction k , and the total number of objects be $q = \sum_k q(k)$. The auctions close in a fixed, known order, and in this model, all are treated as sealed bid auctions. The sealed bid assumption may not be as restrictive as it seems. In fact, the sniping strategy used by many bidders on eBay [20,23] reduces the open-outcry auction to the equivalent of a sealed bid auction.

Let J denote the other bidders in the market, and $A = J \cup i$. The total number of bidders, including i , is $n = |J| + 1$. In a particular auction, a subset, $\mathcal{A} \subseteq A$, of the agents will place bids. Let the bid of bidder j in auction k be denoted b_j^k .

Naturally, the rules of the auctions will affect the bidders' choices of actions. A multiunit auction must have a policy for setting prices.⁴ In this study, we consider only two such policies. The *Mth-price* policy sets the price paid by all winners to the value of the lowest winning bid (this is the policy used in eBay's

⁴ See Ref. [30] for a survey of some pricing policies.

Dutch Auction format).⁵ Under the *pay-your-bid* policy, each winner pays the price she offered. Pay-your-bid is the policy used on Yahoo’s multiunit auctions. In the case of a single unit for sale, the two policies are equivalent.

Given a sequence of sealed bid auctions, the agent must select a bid to place in each auction. Let W^k be the set of bid choices that are acceptable in auction k . Typically, we assume that W^k is the set of integers in some range and is identical across all of the auctions. However, the techniques we develop admit different bid choices in each auction. The number of bid choices is $m = |W^k|$. We assume that ties are broken randomly.

Our agent has a value $v_i(k)$ for an item in k , and bidder $j \in J$ has valuation $v_j(k)$. In this study, we assume that the items available in K are identical and that all participants are interested in only a single unit. We anticipate that the techniques we develop in this paper can be extended to auctions of heterogeneous items if an agent’s valuations for the items are correlated, that is, if learning about an agent’s valuation of one item helps predict its valuation of another item.

Agent i does not know bidder j ’s true value for the items but knows that it is drawn from a distribution D_j . In this model, we assume that valuations are independent and private, but we do not make any particular assumptions about the functional form of the distributions, nor do we assume that the distributions are identical for all of the bidders. We will make various assumptions about whether the bidders in J know each other’s valuations or agent i ’s valuation.

We assume that each participant is present for the first auction and continues to participate in each auction until either she wins or the sequence ends. Thus, a buyer that does not win in auction k will participate in auction $k+1$. We assume that the auctioneer makes public a list of all of the bids once the auction is complete. This is consistent, for instance, with eBay’s policy. Let h_j^k be the sequence of bids that agent j placed in the auctions up to, but not including, k . That is, $h_j^k = \{b_j^1, \dots, b_j^{k-1}\}$. We call h_j^k bidder j ’s *history* up to auction k . The history of all J bidders leading to auction k is denoted as H_j^k .

2.1. Sequential game representation

We model the sequential auction scenario as an extensive form game, $\Gamma(\mathcal{A}, V_{\mathcal{A}}, K, W^K)$, where $\mathcal{A} = J \cup i$ and W^K denotes the bid choices for all of the auctions. A *subgame* has the same structure, except that part of the game has already been played. For example, the subgame that results when bidder j wins the first item is $\Gamma(\mathcal{A}', V_{\mathcal{A}'}, K', W^{K'})$, where $\mathcal{A}' = i \cup J \setminus j$ and $K' = K \setminus \{1\}$.

It is also useful to identify the game structure of individual auctions. Denote a *component* auction game, $\gamma(\mathcal{A}, V_{\mathcal{A}}, W^k)$, in which agents, \mathcal{A} , with valuations, $V_{\mathcal{A}}$, for the items in auction, k , choose bids from the domain W^k . Note that a game (or subgame) is a sequence of component games. In game theoretic terms, γ is the game in which \mathcal{A} is the set of players, W^k are the actions, and the payoff is $v_j(k) - b_j^k$ for the bidder with the highest bid, and zero for everyone else. Because the auction is a sealed bid, all of the bidders’ actions are simultaneous, and the game involves imperfect information.

A simple example with three agents, two items, and two bid levels is shown in Fig. 1. The circles are labeled with the ID of the agent, and the arcs with the bid value ($\{1, 2\}$). The game consists of two stages, the first of which corresponds to the first auction involving all three agents. The second stage involves the two agents who did not win the first item, and for conciseness, we have substituted labeled triangles for subgames on the leaves of the first auction. There are 15 subgames labeled $\gamma_1 \dots \gamma_{15}$, but only three possible unique structures labeled A, B, and C.

Dotted lines connect the decision nodes in the same information set. The small squares at the leaves of the subgames represent terminal states that would be labeled with the payoffs to the agents. The actual value of the payoffs would depend upon each agent’s actual value for the item, the path taken, and the auction’s policy for setting prices. The diamonds denote the random move by nature to break ties among the bids (with the probabilities indicated in parenthesis). This type of move by nature can be handled relatively easily because it does not introduce any asymmetric information. Moreover, it is amenable to the decompositions we introduce in Section 3.

It is obvious from Fig. 1 that a particular component game, γ , can appear many times in the overall

⁵ The *Mth-price* policy derives its name from an analysis in which M is the number of units for sale; the M highest bidders win, and each winner pays the price associated with the M th highest bid.

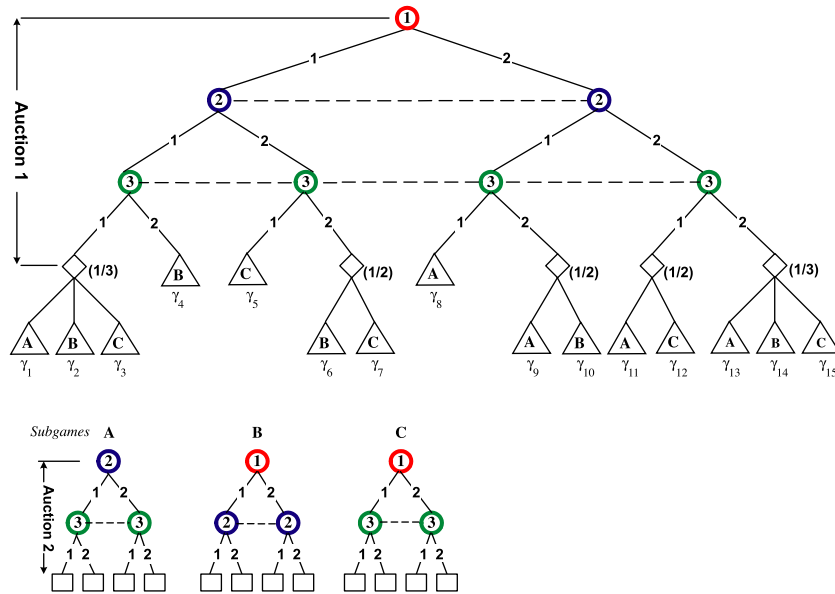


Fig. 1. A sequence of two sealed bid auctions with three agents, one item for sale in each auction and two bid levels.

game Γ . Each second-level component game appears on five different paths of the top-level game. When necessary, we will distinguish a component game using its history as a subscript: γ_{H^k} . The history information is sufficient to uniquely identify each component game instance.

In addition to the imperfect information generated by the sealed bids, the agent also faces incomplete information because it does not know the other bidders' true values, and therefore does not know the other bidders' payoffs. Harsanyi [9] demonstrated that incomplete information games can be modeled by introducing an unobservable move by nature at the beginning of the game which establishes the unknown values. This approach transforms the incomplete information game into a game with imperfect information.

Unfortunately, the move-by-nature approach is computationally problematic. The number of possible moves available to nature is m^n , where m is the size of the domain of $v_j(k)$, and n is the number of agents. Our model permits a continuous range for valuation functions, so the number of choices is not enumerable. In some special cases, analytic solutions can be found to auction games with continuous types [6]. However, this analysis is complex and typically requires restric-

tive assumptions about the distributions of values. Moreover, whether valuations are drawn from discrete or continuous domains, each different market configuration requires a separate analysis.

For these reasons, we investigate the use of Monte Carlo sampling to generate heuristic bidding policies for the incomplete information game. Our approach to the problem can be summarized as follows:

1. Create a sample complete-information game by drawing a set of valuations for other bidders.
2. Solve for a Nash equilibrium of the sample game.
3. Update the agent's bidding policy.

The first step is straightforward Monte Carlo sampling. The second and third steps are the subject of Sections 3 and 4.

3. Leveraging substructure in the complete information game

We built our agent on top of the GAMBIT Toolset. Although GAMBIT includes algorithms that can solve multiplayer games with imperfect information, it cannot solve the straightforward expansion of even very

small instances of the complete-information sequential auction game in a reasonable amount of time.

To see why, consider the size of the extensive form of a complete information sequential auction game with ties broken randomly. The assumption that bidders want only one item means that the winners of a particular auction will not participate in future auctions. Thus, auction $k+1$ has $q(k)$ fewer participants than auction k . In general, the number of agents participating in component game k is $z(k) = n - \sum_{x=1}^{k-1} q(x)$. The number of nodes in the extensive form representation of this game with c auctions is

$$\frac{m^n - 1}{m - 1} + \sum_{k=2}^c \left[\frac{m^{z(k)} - 1}{m - 1} \times \prod_{j=1}^{k-1} \left(m^{z(j)} + \text{EXT}[z(j), m, q(j)] \right) \right].$$

The core of the equation captures the number of nodes in the tree without tie breaking, and the EXT term represents the number of additional terminal nodes added to each component game due to tie breaking. The EXT term expands as

$$\begin{aligned} \text{EXT}[z(j), m, q(j)] &= \sum_{v=1}^m \sum_{i=q(j)+1}^{z(j)} \binom{z(j)}{i} (v-1)^{z(j)-i} \left[\binom{i}{q(j)} - 1 \right] \\ &+ \sum_{v=1}^m \sum_{i=2}^{z(j)-1} \binom{z(j)}{i} \sum_{h=L(j,i)}^{H(j,i)} \binom{z(j)-1}{h} (m-v)^h \\ &\times (v-1)^{z(j)-i-h} \left[\binom{i}{q(j)-h} - 1 \right], \end{aligned}$$

where

$$H(j, i) = \min(q(j) - 1, z(j) - i), \text{ and}$$

$$L(j, i) = \max(q(j) - i + 1, 1).$$

A five-agent, four-item sequential auction with five bid choices and random tie breaking has 4.5 billion decision nodes and is unsolvable with GAMBIT on current workstations. However, as Fig. 1 shows, there is structure in the problem that we can leverage to improve our representation of the game.

The computational aspects of game theory have been studied by economists and computer scientists in the past few years [12–14, 18, 26]. A very promising

thread of work is focused on representations of games that capture their inherent structure and facilitate solution computation. Koller and Pfeffer's GALA language [15] can be used to represent games in *sequence* form, and the authors have developed solution techniques for two-player, zero-sum games represented in this format. The success of GALA is based on the intuition that significant computational savings can be achieved by taking advantage of a game's substructure. This intuition holds for the sequential auction model, and we have employed it to improve upon GAMBIT's default approach.

The default representation of this game in GAMBIT is to expand each of the leaves with an appropriate subgame. Given that the bidders have complete information, all subgames with the same players remaining have the same solution(s). Thus, a single-unit, sealed bid (component) auction with n agents has at most n *unique* subgames—one for each possible set of non-winners. The three component games—A, B, and C—are illustrated in Fig. 1.

Our agent's approach is to create all possible component games and solve them using GAMBIT's C++ libraries. The process is essentially dynamic programming and equivalent to standard backward induction with caching. The expected payoffs from the solution to a component game, γ , involving bidders, \mathcal{J} , are used as the payoffs for the respective agents on the leaves of any component games in Γ which immediately precede γ . The agent solves all the possible smallest component games (i.e., where $k=c$) and recursively constructs higher-order subgames until it solves the root game (i.e., $k=1$).

The number of decision nodes required to express a game in its component form is

$$\sum_{k=1}^c \binom{n}{z(k)} \frac{m^{z(k)} - 1}{m - 1}.$$

The component form representation is exponential in the number of agents and the number of bidding choices. However, the total number of nodes required to express the game is exponentially less than in the full expansion. For example, five-agent, four-item sequential single-unit auctions with five bid choices and random tie-breaking requires only 1931 nodes to encode in its component form compared to the 4.5 billion required for the naive expansion.

It should be noted that the solutions that we are using in the above analysis are Nash equilibria found by GAMBIT for each particular subgame. These solutions may involve either pure or mixed strategies. It is well known [19] that at least one mixed strategy equilibrium always exists; however, it is also often true that more than one Nash equilibria exist. In this study, we simply take the first equilibria found by GAMBIT and leave the question of how, and even whether, to incorporate multiple equilibria to future research. We recognize that our results may be influenced by the order in which GAMBIT finds solutions but also consider it a concern inherent in using off-the-shelf solution technology.

It should also be noted that the procedure described above is consistent with the definition of subgame perfect equilibrium (SPE), a well-known specialization of the Nash equilibria. A profile of strategies is subgame perfect if it entails a Nash equilibrium in every subgame of the overall game [22]. All subgame perfect equilibria are Nash, but the reverse is not necessarily true.

While the decomposition provides an exponential improvement in the number of nodes needed to represent (and hence, solve) the game, the computational cost of finding equilibria for the component games remains a severely limiting factor. Indeed, although the number of bid choices is the base, not the exponent, of the complexity of the extensive form game, we will see in Section 5 that GAMBIT is unable to solve subgames if we increase the number of bid choices beyond a small number.

4. Monte Carlo approximation

In order to participate in this environment, the agent must construct a *policy*, Π , that specifies what action it should take in any state of the game that it might reach. There are many conceivable policies available to our agent.

One simple strategy is to compute the equilibrium strategy in each component game and to bid accordingly. For example, the equilibrium strategy of a single first-price, sealed bid auction in which the other bidders' valuations are drawn uniformly from $[0,1]$ is to bid $b_i^k = (1 - 1/n)v_i(k)$, where n is the number of bidders [17]. We define Π_{myopic} to be the strategy in which the

agent bids according to the equilibrium of each individual sealed bid auction. Thus, the strategy has one element for each potential game size, $\Pi_{\text{myopic}} = \{\pi_z\}$, where z is the size, in number of bidders, of the component game.

In a sequence of sealed bid, single-unit auctions, a Bayes–Nash equilibrium strategy is for a bidder to bid the expected price of the $(q + 1)$ st valuation under the assumption that her bid is among the top q (see Ref. [27] for details). We denote this policy as $\Pi_{(q+1)\text{st}}$ and use it as a benchmark in our empirical evaluation.

If the distributions from which the bidders draw values are not identical, then it would behoove our agent to have a policy that accounted for which other bidders were in the subgame. Thus, $\Pi_{\text{not-id}} = \{\pi_{\mathcal{J}} \subseteq J\}$. That is, the actions in the policy depend upon which subset, \mathcal{J} , of agents remain.

All three policies mentioned thus far are memoryless; they ignore the bids the remaining opponents made in previous auctions. On the other extreme is a policy that uses all possible history information. $\Pi_{\text{history}} = \{\pi_{\mathcal{J}, H_{\mathcal{J}}^k}\}$ encodes the entire tree because the decision at each decision node is a function of the entire history.

The policy that our agent learned in this study is $\Pi_{\text{agg-hist}} = \{\pi_{\mathcal{J}, H_{\mathcal{J}}^k}\}$ where $H_{\mathcal{J}}^k = \{h_{j \in \mathcal{J}}^k\}$, the histories of all other agents, are still in the game. This differs from Π_{history} in that policies are classified by the histories of only those bidders that remain active (\mathcal{J}) rather than by the previous actions of all bidders in J . It is based on the assumption that bidders who are no longer active in the sequential auction (because they have won an item) are irrelevant. Therefore, all component games that have the same opponents and identical previous actions by those agents are aggregated into a class of component games, $\gamma_{\mathcal{J}, H_{\mathcal{J}}^k}$.

In the example in Fig. 1, suppose player 1 is our agent. All paths that lead to subgame A can be ignored because our agent won the item in the first auction. Of the remaining subgames, the set $\{\gamma_2, \gamma_4, \gamma_{10}\}$ have identical histories—bidder 2 bid \$1 in all of them. Similarly, the sets $\{\gamma_6, \gamma_{14}\}$, $\{\gamma_3, \gamma_5, \gamma_{12}\}$, and $\{\gamma_7, \gamma_{15}\}$ can be formed by their common histories.

The agent constructs the policy by sampling the distributions of the other bidders and solving the resulting complete information game. Let L be the collection of sample games constructed, and l be a single instance. Denote the solution returned by GAM-

BIT to instance l as Ω^l . Ω^l is a profile of (possibly mixed) strategies—one for each player—that constitutes an equilibrium for this game instance. Let Ω_i^l specify the policy for agent i , and $\omega_i^l(\gamma)$ is the policy for subgame γ . Note that some decision nodes may not be reachable if the actions that lead to them are played with zero probability. To simplify the notation, we include these unreachable nodes in the following although they have no effect on the solution.

To compute the policy, $\pi_{\mathcal{J}, H_{\mathcal{J}}^k}$, for a decision in game, $\gamma_{\mathcal{J}, H_{\mathcal{J}}^k}$, we take the weighted sum of the equilibrium solutions across all sample games. Let

$$w(b_i^k | \pi_{\mathcal{J}, H_{\mathcal{J}}^k}) = \sum_{l \in L} \sum_{\gamma \in \gamma_{H_{\mathcal{J}}^k}} \Pr(\gamma | \Omega^l) \Pr(b_i^k | \omega_i^l(\gamma)) \quad (1)$$

be the weight assigned to action b_i^k in the class of games identified by $\gamma_{H_{\mathcal{J}}^k}$. Here, $\Pr(\gamma | \Omega^l)$ is the probability that the game would reach subgame γ given that everyone is playing Ω^l (i.e., the product of the probabilities in the mixed strategies on the path leading to γ), and $\Pr(b_i^k | \omega_i^l(\gamma))$ is the probability associated with bid b_i^k in solution $\omega_i^l(\gamma)$.

In previous work [31], we examined a version of the update function with a bias towards actions that generate a higher utility for our agent. The inclusion of utility in the equation biases the agent toward maximizing its expected utility—a useful heuristic, perhaps, but one that is not necessarily consistent with equilibrium behavior. In this paper, we compare the effect of using the biased update function rather than the unbiased Eq. (1). The biased updated function has the form:

$$w(b_i^k | \pi_{\mathcal{J}, H_{\mathcal{J}}^k}) = \sum_{l \in L} \sum_{\gamma \in \gamma_{H_{\mathcal{J}}^k}} \Pr(\gamma | \Omega^l) u_i(\gamma, \Omega^l) \Pr(b_i^k | \omega_i^l(\gamma)), \quad (2)$$

where $u_i(\gamma, \Omega^l)$ is our agent's expected utility of the subgame rooted at γ .

Finally, we normalize the computed weights to derive the probabilities,

$$\Pr(b_i^k | \pi_{\mathcal{J}, H_{\mathcal{J}}^k}) = \frac{w(b_i^k | \pi_{\mathcal{J}, H_{\mathcal{J}}^k})}{\sum_{b \in W^k} w(b | \pi_{\mathcal{J}, H_{\mathcal{J}}^k})}. \quad (3)$$

The result of this process is a policy that specifies a (possibly mixed) strategy for each unique class of component games. We refer to a policy constructed in this manner as a Monte Carlo Approximation (MCA) policy.

5. Empirical results

To evaluate the efficacy of the approach, we simulated several market configurations in which we varied the functional form of the valuation distributions, the form of the update equation, and the strategies of the other bidders. Each of these experimental variables are described in more detail below. The experimental design is similar to our previous work [31]. However, in the results reported herein, we have added the random tie-breaking rule and multiunit auctions.

- **Market Configuration:** The market configuration includes the number of agents, the domain of the bid messages, and the number and types of auctions. We used the following configurations:
 - $\{5,5,s-s-s\}$ contains five agents, five bid levels, and a sequence of three single-item auctions.
 - $\{5,5,s-2Mth\}$ contains five agents, five bid levels, and an auction sequence in which a single-unit auction is followed by an M th-price auction for two units.
 - $\{5,5,s-2PYB\}$ contains five agents, five bid levels, and an auction sequence in which a single-unit auction is followed by a two-unit auction in which the winners pay their bid values.
 - $\{5,4,s-s-s-s\}$ contains five agents, four bid levels, and a sequence of four single-item auctions.
- **Valuation Distribution:** We used three types of distributions: uniform, left-skewed Beta and right-skewed Beta. With the exception of $\{5,4,s-s-s-s\}$, the valuations of the other agents were drawn from [1,6], while our agent's valuation is always fixed at 3.5. In the left-skewed distribution, our agent is likely to have a valuation significantly above average, while in the right-skewed distribution, it will be significantly below average. In experiments with $\{5,4,s-s-s-s\}$, the valuations of the other agents drawn from [1,5], while our agent's valuation is

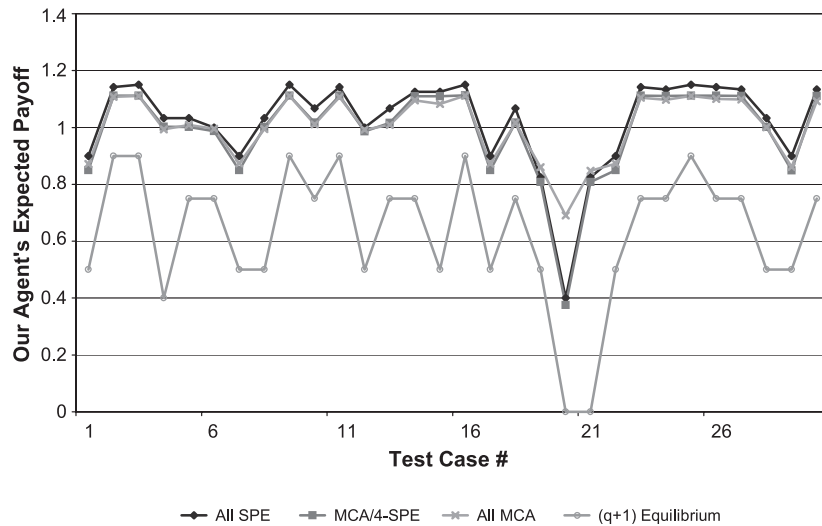


Fig. 2. Our agent's expected payoff in the $\{5,4,s-s-s\}$ market scenario with the other agents' valuations drawn from a uniform distribution, and Eq. (2) is used to update policies.

fixed at 3; this combination was chosen to draw comparisons with our earlier work [31].

- Update Equation: We examined the difference between using Eq. (1) and using Eq. (2) which biases the policy aggregation by the agent's expected utility.
- Bidder Strategies: We studied the effects of various combinations of bidder strategies.
 - All SPE: As a benchmark scenario, we assume that all agents have complete information for a test case, and all of them play the subgame perfect equilibrium computed using our structural decomposition technique with the GAMBIT engine.
 - MCA/ n -SPE: We assume the other agents had complete information, while our agent has incomplete information. Our agent implements the strategy learned from the Monte Carlo policy construction, while the other agents implement their SPE strategies. Because our agent is not playing perfectly, there is no guarantee that the other agents' SPE strategies are equilibrium responses to our imperfect play.⁶ To generate the MCA strategy, the agent trained with 200 samples.

⁶ In theory, it would be possible to determine the opponents' best responses to our heuristic strategy by marginalizing our agent and computing a reduced game in which the other agents' payoffs are impacted by our fixed behavior.

- All MCA: In this scenario, all agents construct and play strategies generated with Monte Carlo policy construction. Note that for these simulations, each opponent must be retrained with each new draw of its valuation.
- $(q+1)$ -Equilibrium: This is another benchmark for the sequence of single-unit auctions. In the $(q+1)$ -equilibrium strategy, all agents play the sequential auction equilibrium strategy [27]. Each agent bids the expected price of the $(q+1)$ st valuation under the assumption that its bid is among the top q .

In the experiments, we measure the utility for our agent (computed as the difference between its value and the price it pays if it wins), the social welfare (the aggregate value of all of the winning agents), and the revenue achieved by the seller. The experiments were run on a Beowulf cluster of eight Linux computers.

In some cases, our agent may find that the game has progressed down a path for which it learned no policy. In such cases, our agent picks the most similar subgame for which it does have a policy. The similarity measure favors subgames with the same bidding pattern, but possibly different agents, over subgames with the same agents but different bidding patterns.

Fig. 2 shows our agent's utility on 30 randomly selected problem instances from the $\{5,4,s-s-s\}$

market scenario with other agents' valuations drawn from the uniform distribution. For each problem instance, the four strategy combinations were tested, and the update Eq. (2) is used. The performance of the Monte Carlo strategy is quite close to that of the subgame perfect equilibrium, both when the other agents play perfectly and when they construct their own Monte Carlo strategies. From this result, we conclude that the approximation technique generates policies that perform quite well in this environment.

The $(q+1)$ -equilibrium strategy is included in Fig. 2, although it is important to note that it represents a slightly different game than the other three. Agents must be allowed to place real-valued bids in the $(q+1)$ -equilibrium strategy, while in the other three, we are restricting bids to integer values. This distinction explains, for instance, why our agent achieves zero utility in Fig. 2 under the $(q+1)$ -equilibrium strategy when it has the lowest value among the five agents. When bid values are restricted, it is more likely that our agent will end up in a tie and therefore achieve a positive surplus with some probability. Nevertheless, the pattern of the payoffs for the $(q+1)$ -equilibrium strategy is quite similar to our empirical results.

One aspect of our previous work which we wanted to examine was the effect of the utility term in Eq. (2).

Fig. 3 shows our agent's expected utility on the same 30 test cases when trained with the same training data and Eq. (1). Although Figs. 2 and 3 look nearly identical, close inspection shows that Eq. (2) performs slightly better than Eq. (1) in the sense that it more closely approximates the subgame perfect outcomes. For this reason, we continue to use Eq. (2) in the rest of the empirical tests.

Figs. 4 and 5 show similar correspondence between the strategies when the other agents' valuations are drawn from right-skewed and left-skewed Beta distributions, respectively. Notice that in the left-skewed distribution, our agent achieves higher payoffs, while in the right-skewed case, our agent receives lower payoffs. This result is expected given that the expected average valuation will be lower when the opponents are drawn from a left-skewed distribution and higher when drawn from a right-skewed distribution.

The next set of experiments involved five-agent, three-item scenarios. We compared two multiunit auction scenarios, $\{5,5,s-2Mth\}$ and $\{5,5,s-2PYB\}$, against a sequence of three single-unit auctions, $\{5,5,s-s-s\}$, over the same 30 uniform-distribution sample instances tested above. Figs. 6 and 7 show how closely the performance of the MCA strategy tracks that of the subgame perfect strategy for $\{5,5,s-$

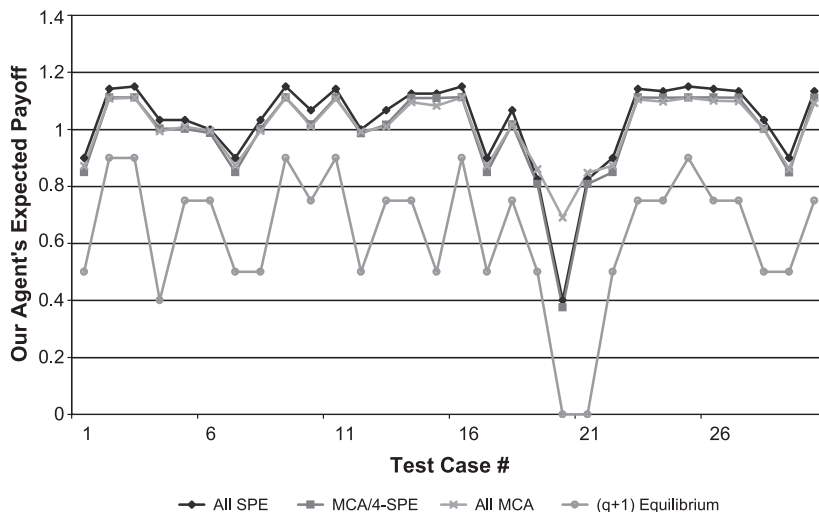


Fig. 3. Our agent's expected payoff in the $\{5,4,s-s-s\}$ market scenario with the other agents' valuations drawn from a uniform distribution, and Eq. (1) is used to update policies.

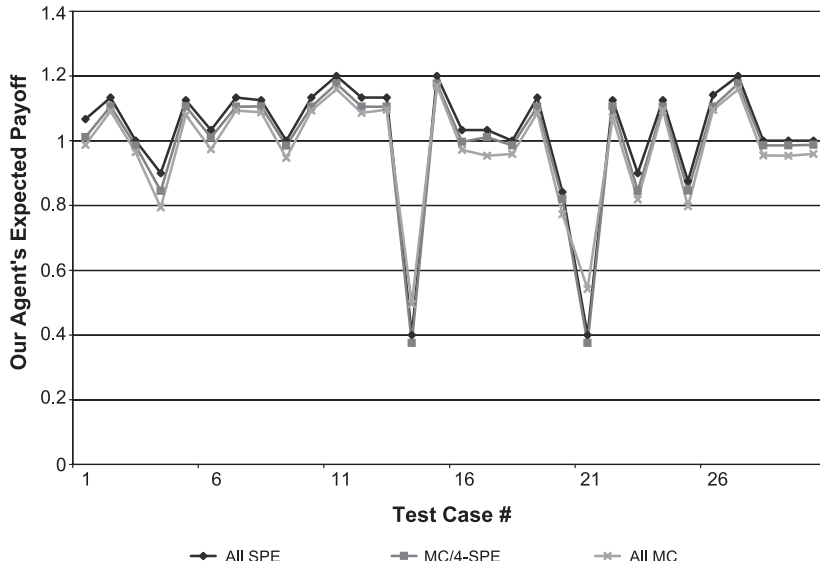


Fig. 4. Our agent's expected payoff in the {5,4,s-s-s} market scenario with the other agents' valuations drawn from a right-skewed Beta distribution.

2Mth} and {5,5,s-2PYB}, respectively. Fig. 8 shows the contrast in our agent's payoff for the three scenarios. The results from {5,5,s-2Mth} and {5,5,s-2PYB} are nearly identical (and may appear to be a single line), while significant variation exists

in results from {5,5,s-s-s}. Notice that our agent performed significantly better in both {5,5,s-2Mth} and {5,5,s-2PYB} than in {5,5,s-s-s}. It is clear that, overall, the agents are bidding lower in the multiunit scenarios, and our agent is playing a mixed strategy

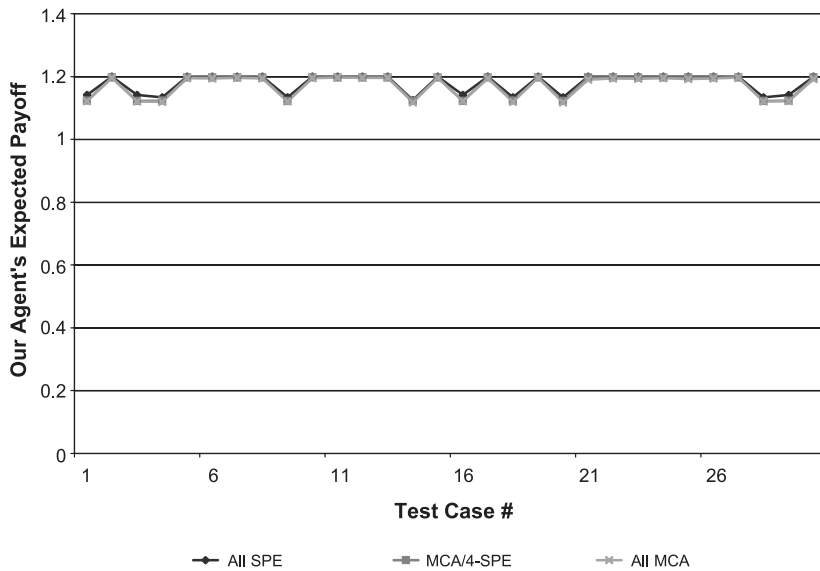


Fig. 5. Our agent's expected payoff in the {5,4,s-s-s} market scenario with the other agents' valuations drawn from a left-skewed Beta distribution.

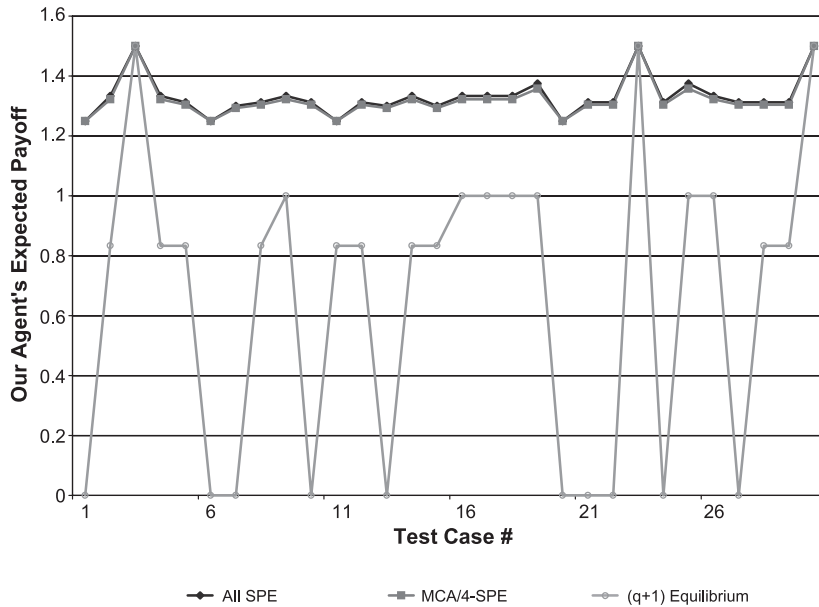


Fig. 6. Our agent's expected payoff in the {5,5,s-2Mth} scenario with the other agents' valuations drawn from a uniform distribution.

that is more successful. However, it remains to be seen whether there is a game theoretic explanation for this outcome, or whether it is a byproduct of our technique or the manner in which GAMBIT returns solutions.

Fig. 9 shows the social welfare achieved in all three scenarios. The welfare achieved in scenario {5,5,s-s-s} is slightly better than the two multiunit cases, whose graphs are again nearly coincident. This is consistent with the observation that the agents are

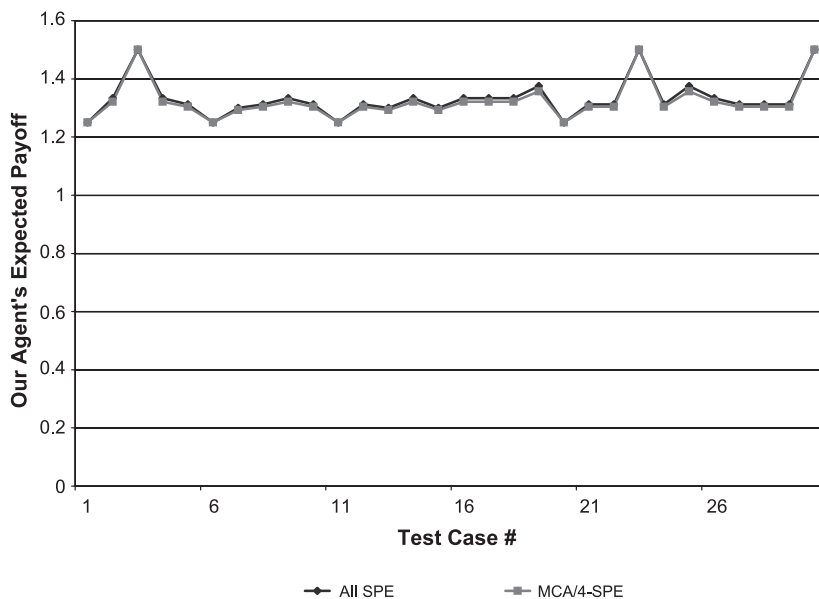


Fig. 7. Our agent's expected payoff in the {5,5,s-2PYB} scenario with the other agents' valuations drawn from a uniform distribution.

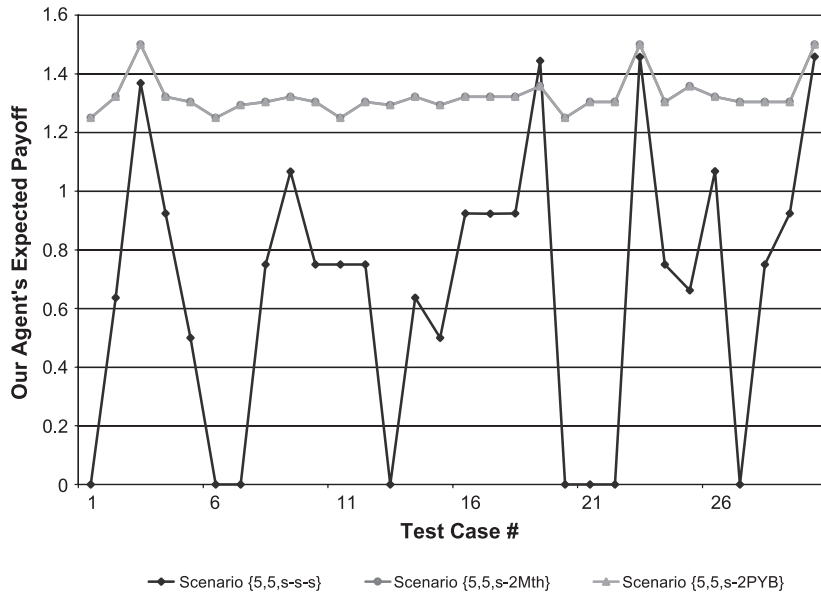


Fig. 8. Comparison of our agent's expected payoff among different types of auctions by using MCA strategy, while the other agents' valuations are drawn from a uniform distribution.

behaving more collaboratively in the multiunit auction by bidding lower and letting the tie-breaking determine the winner. When the agent with the highest value allows the allocation to be determined by tie-

breaking rather than by placing a better bid, it is more likely that a less than optimal allocation will result.

Fig. 10 shows the effect of the different auction scenarios on the sellers' revenue. Again, because

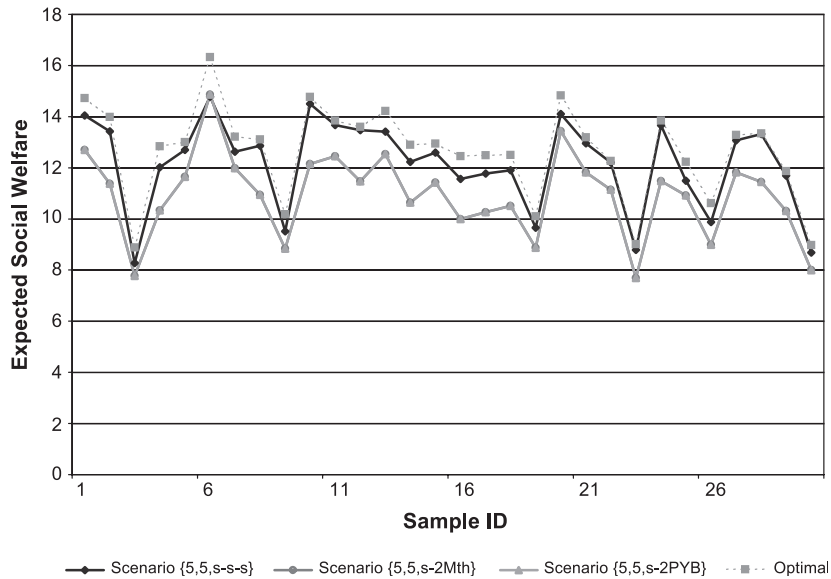


Fig. 9. Comparison of the expected social welfare among different auction scenarios when our agent plays its MCA strategy, and the other agents' valuations are drawn from a uniform distribution.

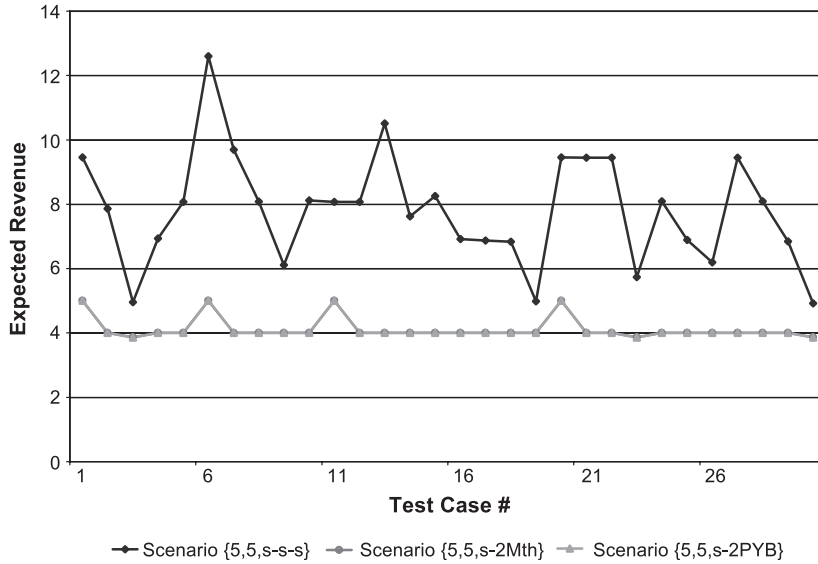


Fig. 10. Comparison of the expected revenue among different auction scenarios when our agent plays its MCA strategy, and the other agents' valuations are drawn from a uniform distribution.

buyers are acting more competitively in the single-unit auctions, the sellers achieve greater revenue than in the multiunit auction scenarios.

6. MCA strategies and sequential equilibria

The notion of *sequential equilibrium*, first introduced by Kreps and Wilson [16], is closely related to the subgame perfect equilibrium concept proposed by Selten [22] but extended to games of incomplete information. In particular, a sequential equilibrium is defined in terms of beliefs at decision points in the game and requires that an equilibrium policy be consistent with those beliefs. In this section, we show that the MCA policy at a node implicitly captures the agent's beliefs about which opponent valuations would explain the fact that the agent arrived at a particular decision point in the game tree.

Building on the notation above, let Ω^V be an equilibrium profile of the game when agents have valuation profile V . In this analysis, we do not aggregate games that have compatible histories, thus, we develop the conditional probabilities in terms of unique histories rather than subgame groups. Let $\Pr(H_j^k | \Omega^V)$ be the probability that the policies select-

ed by Ω^V follow history H_j^k . Let Φ be our agent's belief function, and $\Phi(V)$ be our agent's belief that the other agents have valuation profile V . Given history H_j^k , the probability that the other agents have profile V is given by

$$\Pr(V|H_j^k) = \frac{\Pr(H_j^k | \Omega^V) \Phi(V)}{\int_{\vartheta} \Pr(H_j^k | \Omega^{\vartheta}) \Phi(\vartheta)}$$

In other words, the probability that the other agents have profile V given the observed history is the probability that the history is played given profile V divided by the probability that the history is played among all possible valuation profiles.

In addition to beliefs, a sequential equilibrium must also define a policy for a subgame that is consistent with the beliefs. Here, we simply let the policy be the *average policy*, that is, the policy constructed by taking an average over all action profiles, weighted by the likelihood of seeing V , given that we have reached the subgame. In other words, the probability that our agent plays b_i^k in subgame $\gamma_{H_j^k}$ is

$$\Pr(b_i^k | H_j^k) = \int_V \Pr(V | H_j^k) \Pr(b_i^k | \omega_i^V(H_j^k)).$$

The MCA approach is a numerical approximation of the above. For a sufficient number of samples L ,

$$\Pr(V | H_j^k) \approx \frac{\Pr(H_j^k | \Omega^V) \Phi(V)}{\sum_{l \in L} \Pr(H_j^k | \Omega^l) \Phi(l)}.$$

Because all samples are equally likely to be drawn, then $\Phi(V) = \Phi(l)$, and the above reduces to

$$\Pr(V | H_j^k) = \frac{\Pr(H_j^k | \Omega^V)}{\sum_{l \in L} \Pr(H_j^k | \Omega^l)}. \quad (4)$$

The numerical approximation of the average policy is

$$\Pr(b_i^k | H_j^k) = \sum_{l \in L} \Pr(V | H_j^k) \Pr(b_i^k | \omega_i^l(H_j^k)).$$

Substituting in Eq. (4) gives

$$\Pr(b_i^k | H_j^k) = \frac{\sum_{l \in L} \Pr(H_j^k | \Omega^l) \Pr(b_i^k | \omega_i^l(H_j^k))}{\sum_{l \in L} \Pr(H_j^k | \Omega^l)}. \quad (5)$$

We can now show the correspondence between Eqs. (3) and (5). First, notice that the denominator of Eq. (3),

$$\sum_{b \in W^k} \sum_{l \in L} \sum_{\gamma \in \gamma_{H_j^k}^l} \Pr(\gamma | \Omega^l) \Pr(b_i^k | \omega_i^l(\gamma))$$

reduces to

$$\sum_{l \in L} \sum_{\gamma \in \gamma_{H_j^k}^l} \Pr(\gamma | \Omega^l).$$

Now the difference between the two formulations reduces to the variations in the notation. In Eq. (3), we have used notation consistent with $\Pi_{\text{agg-hist}}$ which aggregates the subgames with compatible histories. Thus, the condition on the LHS of the equation is in terms of the group of equivalent subgames, and the numerator on the RHS includes a summation over those same subgames. Despite that difference, the functional form of the two equations is identical.

7. Related work

This paper continues the study begun by Zhu and Wurman [31] which studied single-unit sequential

auctions with deterministic tie-breaking. In this paper, we admit multiunit auctions, random tie-breaking rules, and slightly larger problem sizes. Moreover, we connect the MCA approach directly to belief updating and sequential equilibria.

Our main focus is to study the feasibility of using the game theory as a solution tool in a computational agent adaptable to various electronic market configurations. The copious research on auctions and game theory provides a backdrop for our effort. See Klemperer [11] for a broad review of auction literature including a discussion of sequential auctions for homogeneous objects. Weber [27] showed that the equilibrium strategies for the bidders when the objects are sold in sequential first-price, sealed bid auctions is to bid the expected price of the object in each auction. This result is developed under the assumption that only the clearing price is revealed in previous auctions. In many current online auction environments, the actual bids and their associated bidders are revealed. As far as we know, none of the theoretical results have addressed the model with complete bid revelation. In addition, we are not aware of any research on sequences of auctions with different rules.

Monte Carlo sampling has been previously used in conjunction with games of incomplete information. Frank et al. [5] described an empirical study of the use of the Monte Carlo sampling method on a simple complete binary game tree. They drew the discouraging conclusion that the error rate quickly approaches 100% as the depth of the game increases. However, perhaps because Frank et al. considered only pure strategy equilibrium in a two-person, zero-sum game, these negative results did not evidence themselves in our study.

Bampton [2] investigated the use of Monte Carlo sampling to create a heuristic policy for the (imperfect information) game of Bridge. In Bampton's paper, he simply collected the player's decision in every sampled game and accumulated the chance–minimax values for each alternative at each decision node. Our method of accumulating sampled data is quite different from Bampton's approach, again because our game is not a two-player, zero-sum game.

Researchers in artificial intelligence have recently been studying trading agents. A significant amount of work has gone into agents for the Trading Agent Competition (TAC) [7,24,28]. The TAC environment

is significantly more complex than the simple scenarios presented here, and to date, none of the implemented agents model opponent behavior in a significant way.

Anthony et al. [1] investigated agents that can participate in multiple online auctions. The authors posited a set of “tactics” and then empirically compared the performance of these tactics in a simulated market that consists of simultaneous and sequential English, Dutch, and Vickrey auctions. While the bidding strategies seem to resonate with particular aspects of human behavior (e.g., the “desperateness” strategy), they do not seem to have a foundation in any theory.

Boutilier et al. [3] developed a sequential auction model in which the agent values combinations of resources, while all other participants value only a single item. Unlike our model, the Boutilier formulation does not explicitly model the opponents, although, like our model, it benefits from a dynamic programming approach to solving the decision problem.

Hon-Snir et al. [10] proposed an iterative learning approach to solve repeated first-price auctions. They developed a repeated auction model which converges to an equilibrium strategy for a one-shot auction after many rounds of repeated auctions. In addition to the differences in the overall structure of the marketplace, their work differs from ours in that they treat the other bidders as naive players. Specifically, they assume the opponents’ next bid vectors are distributed according to a weighted empirical distribution of their past bid vectors.

8. Conclusion

This study represents a first step in exploring the implementation of computational game theory in a simple trading agent. We show how Monte Carlo sampling can be used to construct a bidding policy that performs comparably to the subgame perfect equilibrium. This strategy takes advantage of information revealed in prior auctions in the sequence to improve play in later auctions. Importantly, the architecture is flexible in that it can handle a variety of simple auction types and different types of other bidders. Equally important, the approach is computationally limited by our ability to solve the component games which suggests that algorithms for solving component games, particularly ones with well-struc-

tured payoff and action spaces, is an important area for further research.

We plan to continue this work and integrate more auction types and to explore scenarios in which the agent’s and other bidders’ preferences are more complex, including scenarios in which the buyers may want more than one item. We would also like to add an aggregate buyer to the model to represent the large number of unmodeled opponents often found in public markets. Finally, we plan to explore auction sequences in which the bidders’ valuations are correlated across the items but not necessarily identical.

Acknowledgements

This project was funded by NSF CAREER award 0092591-0029728000 to the second author, and follows on the Masters Thesis of Weili Zhu. We wish to thank William Walsh, the members of the Intelligent Commerce Research Group at NCSU, and the anonymous reviewers for their insightful comments. We are also indebted to the operators of the NCSU Beowulf cluster, on which we ran the experiments, and to the developers of GAMBIT, particularly Ted Turocy and Andrew McLennan. Any errors are purely our own.

References

- [1] P. Anthony, W. Hall, V. Dang, N.R. Jennings, Autonomous agents for participating in multiple on-line auctions, IJCAI Workshop on E-Business and the Intelligent Web, Seattle, WA, 2001, pp. 54–64.
- [2] H.J. Bampton, Solving imperfect information games using the Monte Carlo heuristic, Technical Report, University of Tennessee, Knoxville, 1994.
- [3] C. Boutilier, M. Goldszmidt, B. Sabata, Sequential auctions for the allocation of resources with complementarities, Sixteenth International Joint Conference on Artificial Intelligence, Stockholm, 1999, pp. 527–534.
- [4] A. Byde, C. Preist, N. Jennings, Decision procedures for multiple auctions, Proceedings of the first International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), 2002, pp. 613–620.
- [5] I. Frank, D. Basin, H. Matsubara, Monte-Carlo sampling in games with imperfect information: empirical investigation and analysis, Game Tree Search Workshop, 1997.
- [6] D. Fudenberg, J. Tirole, Game Theory, MIT Press, 1996.
- [7] A. Greenwald, P. Stone, Autonomous bidding agents in the trading agent competition, IEEE Internet Computing 5 (2) (2001 April) 52–60.

- [8] K. Guler, B. Zhang, Bidding by empirical Bayesians in sealed bid first price auctions. Technical Report HPL-2002-212, HP Laboratories, Palo Alto, 2002.
- [9] J.C. Harsanyi, Games with incomplete information played by Bayesian players. *Management Science*, 14:159–182, 320–334, 486–502, 1967–8.
- [10] S. Hon-Snir, D. Monderer, A. Sela, A learning approach to auctions, *Journal of Economic Theory* 82 (1998) 65–88.
- [11] P. Klemperer, Auction theory: a guide to the literature, in: P. Klemperer (Ed.), *The Economic Theory of Auctions*, Edward Elgar, Massachusetts, 2000, pp. 3–62.
- [12] D. Koller, N. Megiddo, The complexity of two-person zero-sum games in extensive form, *Games and Economic Behavior* 4 (1992) 528–552.
- [13] D. Koller, N. Megiddo, B. von Stengel, Fast algorithms for finding randomized strategies in game trees, 26th ACM Symposium on the Theory of Computing, 1994, pp. 750–759.
- [14] D. Koller, N. Megiddo, B. von Stengel, Efficient computation of equilibria for extensive two-person games, *Games and Economic Behavior* 14 (2) (1996) 247–259.
- [15] D. Koller, A. Pfeffer, Representations and solutions for game-theoretic problems, *Artificial Intelligence* 94 (1) (1997) 167–251.
- [16] D.M. Kreps, R. Wilson, Sequential equilibria, *Econometrica* 50 (4) (1982) 863–894.
- [17] R.P. McAfee, J. McMillan, Auctions and bidding, *Journal of Economic Literature* 25 (1987) 699–738.
- [18] R.D. McKelvey, A. McLennan, Computation of equilibria in finite games, in: H. Amman, D.A. Kendrick, J. Rust (Eds.), *The Handbook of Computational Economics*, vol. 1, Elsevier Science, Amsterdam, 1996, pp. 87–142.
- [19] J. Nash, Two-person cooperative games, *Proceedings of the National Academy of Sciences* 21 (1950) 128–140.
- [20] A.E. Roth, A. Ockenfels, Last-minute bidding and the rules for ending second-price auctions: evidence from eBay and Amazon auctions on the internet, *American Economic Review* 92 (4) (2002 June) 1093–1103.
- [21] J. Rust, J.H. Miller, R. Palmer, Characterizing effective trading strategies: insights from a computerized double auction tournament, *Journal of Economic Dynamics and Control* 18 (1994) 61–96.
- [22] R. Selten, Re-examination of the perfectness concept for equilibrium points in extensive games, *International Journal of Game Theory* 4 (1975) 25–55.
- [23] H.S. Shah, N.R. Joshi, P.R. Wurman, Mining for bidding strategies on eBay, *WEBKDD 2002 Workshop*, Edmonton, Alberta, Canada.
- [24] P. Stone, M.L. Littman, S. Singh, M. Kearns, ATTac-2000: an adaptive autonomous bidding agent, *Journal of Artificial Intelligence Research* 15 (2001) 186–206.
- [25] G. Tesauro, R. Das, High-performance bidding agents for the continuous double auction, *IJCAI Workshop on Economic Agents, Models, and Mechanisms*, IJCAI, Seattle, Washington, 2001, pp. 42–51.
- [26] B. von Stengel, Computing equilibria for two-person games, in: R.J. Aumann, S. Hart (Eds.), *Handbook of Game Theory*, vol. 3, North-Holland, Amsterdam, 2002, pp. 1723–1759.
- [27] R.J. Weber, Multiple-object auctions, in: R. Engelbrecht-Wiggans, M. Shubik, R.M. Stark (Eds.), *Auctions, Bidding and Contracting: Uses and Theory*, New York University Press, New York, 1983, pp. 165–191.
- [28] M.P. Wellman, A. Greenwald, P. Stone, P.R. Wurman, The 2001 trading agent competition, *Fourteenth Conference on Innovative Applications of Artificial Intelligence*, Edmonton 34, 2002, pp. 935–941.
- [29] M.P. Wellman, P.R. Wurman, K.A. O'Malley, R. Bangera, Shou-De Lin, D.W.E. Reeves, W.E. Walsh, Designing the market game for a trading agent competition, *IEEE Internet Computing* 5 (2) (2001 Mar.–Apr.) 43–51.
- [30] P.R. Wurman, W.E. Walsh, M.P. Wellman, Flexible double auctions for electronic commerce: theory and implementation, *Decision Support Systems* 24 (1998) 17–27.
- [31] W. Zhu, P.R. Wurman, Structural leverage and fictitious play in sequential auctions, *Eighteenth National Conference on Artificial Intelligence*, Edmonton 35, 2002, pp. 385–390.



Gangshu Cai is a PhD candidate in Operations Research and Computer Science at North Carolina State University. He received his MS in Economics and Statistics in 1999 and BS in Physics in 1996 from Peking University.



Peter R. Wurman is an Assistant Professor of Computer Science at North Carolina State University where he teaches courses in e-commerce. His research is focused on the intersection of artificial intelligence and economics, with particular focus on mechanism design and the development of trading agents. He received his PhD from the University of Michigan in 1999, and an SB from MIT in 1987. Dr. Wurman is involved in numerous academic activities including helping to initiate the Trading Agent Competitions.