

Structural Leverage and Fictitious Play in Sequential Auctions

Weili Zhu & Peter R. Wurman

Computer Science
North Carolina State University
Raleigh, NC 27695-7535 USA
wzhu@unity.ncsu.edu, wurman@csc.ncsu.edu

Abstract

We model sequential, sealed-bid auctions as a sequential game with imperfect and incomplete information. We develop an agent that, through fictitious play, constructs a policy for the auctions that 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 incomplete information game is intractable, we develop more concise representations that take 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 agents that play myopically. The technique performs quite well in these empirical studies, though the tractable problem size is still quite small.

Introduction

Trading agents are software programs that participate in electronic markets on behalf of a user. Simple bidding tools, like eSnipe¹, have begun to appear that enable bidders to automate their last second bids on eBay. However, bidders often have a plethora of auctions in which they could participate, and need agents which can manage bidding across several auctions possibly hosted at multiple auction sites. In addition, it is apparently common on eBay for a small community of expert collectors to recognize each other, which creates the opportunity to directly model one's competitors.

We model this common scenario as a sequence of single-unit auctions with a small set of identified, risk-neutral participants, each of whom wants one unit of the item and has an independent, private value for the item. We assume that our agent knows the distribution of the other agents' valuations, but not their actual values.

The premise of this work is that 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.

We cast the problem as an incomplete, imperfect information game. However, the straightforward expansion of the game is intractable even for very small problems, and it is beyond the capability of the current algorithms to solve for the Bayes-Nash equilibria. Thus, we examine the construction of a bidding policy through fictitious play. 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.

We also find that the straightforward expansion of the imperfect information game cannot be solved directly by current game solvers (e.g., GAMBIT). Thus, we develop methods to take advantage of the sequential structure that greatly reduces the space required to represent the game and enables us to solve much larger games with GAMBIT.

Model

Consider an agent, i , that has the task of purchasing one item from a set of items, K , which are being auctioned sequentially. Let q be the number of items in K , and c_k be the auction for the k th item. Let J denote the other bidders in the market. The total number of bidders, including i , is $n = |J| + 1$.

Our agent has a value $v_i(k)$ for item k , and bidder $j \in J$, has valuation $v_j(k)$. In this study, we assume that the items in K are identical and that all participants are interested in only a single unit. We believe that the techniques we develop can be extended to auctions of heterogeneous items if an agent's valuations for the items are correlated.

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.

Naturally, the rules of the auctions will affect the bidders' choices of actions. Although the examples we study assume first-price, sealed-bid auctions, the techniques are generalizable to other auction formats.

Given a sequence of first-price, sealed-bid auctions, the agent must select a bid to place in each auction. Let B^k be the set of bid choices that are acceptable in auction k . Typically, we assume that B^k is the set of integers in some range, $[0, R^k]$, and is identical across all of the auctions. However, the techniques we develop admit different bid choices in each auction. Let $m = |B^k|$. Denote agent j 's bid in auction c_k as b_j^k .

A buyer that does not win in auction c_k will participate in auction c_{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, c_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 is denoted H_j^k .

We model the sequential auction scenario as an extensive form game, $\Gamma(\mathcal{A}, V_{\mathcal{A}}, B^K, K)$, where $\mathcal{A} = J \cup i$ and B^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 $\Gamma(\mathcal{A}, V_{\mathcal{A}}, B^{\kappa}, \kappa)$ where $\mathcal{A} = J \setminus j \cup i$ and $\kappa = K \setminus \{1\}$.

It is also useful to identify the game structure of individual auctions. Denote a *component* auction game $\gamma(\mathcal{A}, V_{\mathcal{A}}^k, B^k)$, in which agents \mathcal{A} , with valuations $V_{\mathcal{A}}^k$ for item k , choose bids from the set B^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, B^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 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 Figure 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. The individual subgames are labeled A–H. The small squares at the leaves of the tree represent terminal states, and for the purpose of illustration, one of these is labeled with payoffs corresponding to initial valuations of 3 for each bidder. Finally, the dotted lines connect decision nodes in the same information set.

For this example, and those used in our tests, ties were broken in favor of the agent with the lowest index. Admittedly, this creates a bias in the experimental results that we intend to address in future work. The technique we develop could be extended to allow other tie breaking policies by branching the nodes where the tie occurred with a move by nature for each of the possible outcomes. 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 the next section.

Note that a particular component game, γ , can appear many times in the overall game Γ . For instance, in Figure 1,

the component game in which the second item is auctioned to bidders 2 and 3 appears five times. When necessary, we will distinguish these component games by their histories, $\gamma_{H_j^k}$. The history information is sufficient to uniquely identify each component game.

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 (1967) 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. First, 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 defines a continuous range for valuation functions, so the number of choices is not enumerable. In some special cases, analytical solutions can be found to auction games with continuous types (Fudenberg & Tirole 1996). However this analysis is complex and typically requires restrictive assumptions about the distributions of values. Second, even if we restrict the valuation functions to a countable set, the standard algorithms for solving the incomplete information game cannot solve even very small problems in a reasonable amount of time on current computing hardware.

For these reasons, we investigate the use of fictitious play 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 the next two sections.

Leveraging Substructure

We built our agent on top of the GAMBIT Toolset.² Although GAMBIT includes algorithms that can solve multi-player games with incomplete, 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 the complete information game. The assumption that bidders want only one item means that the winner of the first auction will not participate in the next auction. Thus, each auction has one fewer participant than the previous. The number of

²The GAMBIT toolset is a software program and set of libraries that support the construction and analysis of finite and extensive form multi-player games. See <http://www.hss.caltech.edu/gambit/Gambit.html>.

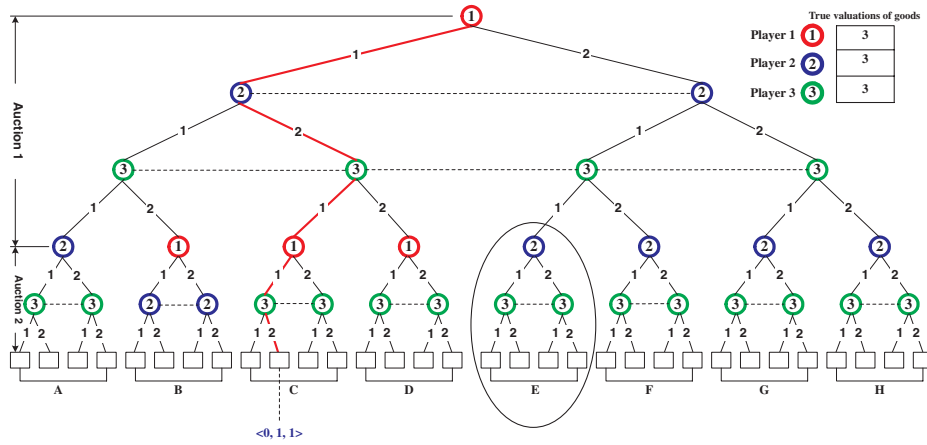


Figure 1: A sealed-bid auction with three agents, two items and two bid levels.

nodes in the extensive form representation of the game with q auctions is $\frac{m^e - 1}{m - 1}$ where $e = q(n - (q - 1)/2)$. A five agent, four item sequential auction with five bid choices has 1.5 billion decision nodes. Thus, to use GAMBIT to compute solutions to our sampled games, we need to improve its performance.

The computational aspects of game theory have been studied by economists and computer scientists in the past few years (Koller & Megiddo 1992; Koller, Megiddo, & von Stengel 1994; 1996; McLennan & McKelvey 1996). 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 (1997) 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 solution method.

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 sealed-bid auction with n agents has at most n *unique* subgames—one for each possible set of non-winners. Figure 2 illustrates the four unique component games for the sequential auction shown in Figure 1.

Our agent’s approach is to create all possible component games and solve them using GAMBIT’s C++ libraries. The process is equivalent to standard backward induction, enhanced 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 which lead to γ in Γ . The agent solves all possible smallest component games (i.e. games of size $z = n - q + 1$). Then, using the results of the lower order solutions, the agent solves all possible component games of size $z + 1$, until it solves the root game.

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

$$\sum_{t=1}^q \binom{n}{t-1} \frac{m^{n-t+1} - 1}{m - 1}.$$

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, the five agent, four item sequential auction with five bid choices requires only 1931 nodes to encode, compared to the 1.5 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 (Nash 1950), 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.

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 subgames remains a bottleneck.

Fictitious Play

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. One simple strategy that our agent could implement 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 (McAfee & McMillan 1987). We define Π_{myopic} to be the strategy in which the agent bids according to the

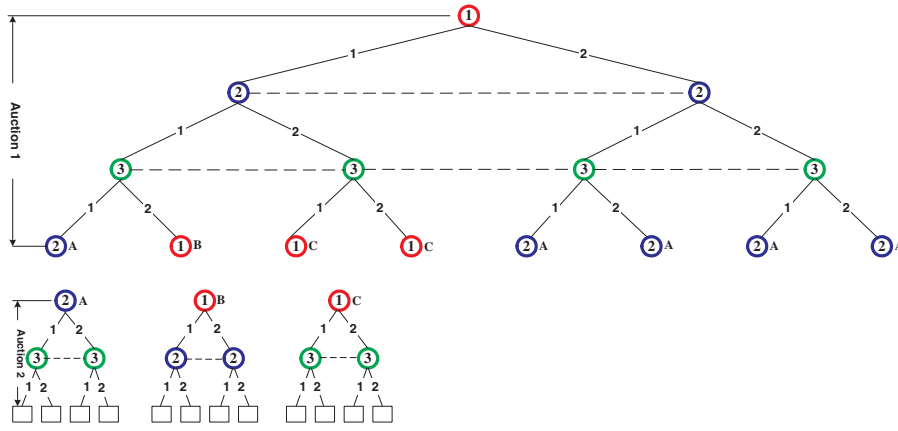


Figure 2: The decomposition of the three bidder, two item game into unique component games.

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.

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.

$\Pi_{\text{not-id}}$ is memoryless; it ignores 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 agents still in the game. 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 agents and identical previous actions by those agents, are aggregated into a class of component games, $\gamma_{\mathcal{J}, H_{\mathcal{J}}^k}$. This differs from Π_{history} because policies are classified by the histories of only those bidders that remain active (\mathcal{J}), rather than by J .

In the example in Figure 1 suppose player 3 is our agent. Subgame B can be ignored because player 3 won the item in the first auction. Of the remaining subgames, the set $\{A, E, F\}$ have identical histories—bidder 2 bids \$1 in both of them—as does the set $\{G, H\}$. Thus, our agent will construct a policy for each of the following decision sets: $\{A, E, F\}$, $\{C\}$, $\{D\}$, and $\{G, H\}$.

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 a single instance. Denote the solution to instance l as Ω^l . Ω^l is a (possibly mixed) strategy that the agent would play in equilibrium for this game instance. Thus, Ω^l specifies a probability function over all bid choices at each component game, and $\omega_{\mathcal{J}}^l$ 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 even though they have no effect on the solution.

We now compute the policy, $\pi_{\mathcal{J}, H_{\mathcal{J}}^k}$, for a decision node $\gamma_{\mathcal{J}, H_{\mathcal{J}}^k}$ by taking the weighted sum of the equilibrium solutions across all sample games. Let

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

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 i would reach γ given that it is playing Ω^l (i.e. the product of the probabilities on the path leading to γ), $U(\gamma, \Omega^l)$ is the expected utility the subgame rooted at γ , and $\Pr(b_i^k | \omega_{\mathcal{J}}^l)$ is the probability associated with bid b_i^k in solution $\omega_{\mathcal{J}}^l$. The first two terms in the summation represent the amount to weight this sample in comparison with other observations of this decision node. The inclusion of utility in the equation biases the agent toward maximizing its expected utility—a useful heuristic, but one that is not necessarily consistent with equilibrium behavior.

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

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

The result of this process is a policy that specifies a (possibly mixed) strategy for each unique class of component games.

There are obvious connections between our scheme and *belief revision*. Space does not permit a full discussion, but we point out that the 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.

Empirical Results

To evaluate the agent’s ability to perform effectively in this environment, we experimented with a scenario involving

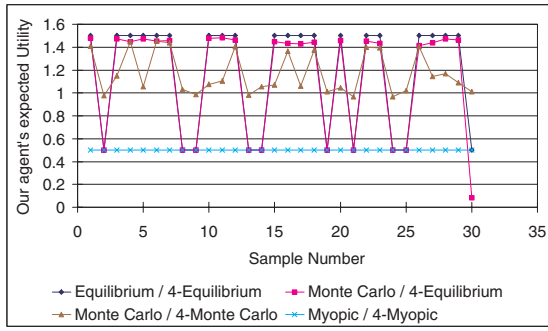


Figure 3: Our agent’s expected utility when the other bidders’ valuation are drawn from a uniform distribution.

four other bidders with valuations drawn from $[1, 5]$, and a sequence of four first-price, sealed-bid auctions. The bidders actions were restricted to four bid options. Our agent’s valuation always remains in the middle of the valuation interval, and we varied the distribution of the other bidders’ valuations. The agent trained with 100 samples of the opponent valuations. We then compared our agent’s performance in various combinations of its strategy and the strategies of the other bidders.

Equilibrium / 4-Equilibrium - represents the combination in which all participants play the Nash equilibrium strategy. It assumes all participants have complete information. We use it as a perfect benchmark because we can’t play better than our agent’s Nash equilibrium strategy with other players all playing their Nash equilibrium strategies.

Monte Carlo / 4-Equilibrium - shows the performance of our agent using the heuristic strategy when the four other bidders play their Nash equilibrium strategy. This is a *best defense* scenario, in which we assume that the other players know our agent’s valuation and will play the game perfectly even though our agent does not know their valuations.

Monte Carlo / 4-Monte Carlo - all five participants learn and then play a heuristic strategy generated by Monte Carlo simulation.

Myopic / 4-Myopic - represents the case where all agents use a simple myopic strategy. The myopic strategy is to play the equilibrium strategy for each auction individually.³

Figure 3 shows our agent’s expected utility in each of 30 different problem instances where the other bidders’ valuations were drawn from a uniform distribution. For each problem instance we tested the four combinations of strategies listed above.

The experimental results show that our learned strategy performs better than the myopic strategy and quite close to the optimal equilibrium strategy. On average, the heuristic strategy performs better against other bidders also playing the heuristic strategy than against bidders playing Nash equilibrium strategies. However, we suspect that this may

³An anonymous reviewer correctly pointed out that a better strawman would be the sequential auction equilibrium strategy: bid the expected price of the $(q - k + 2)$ th item under the assumption the agent wins (Weber 1983).

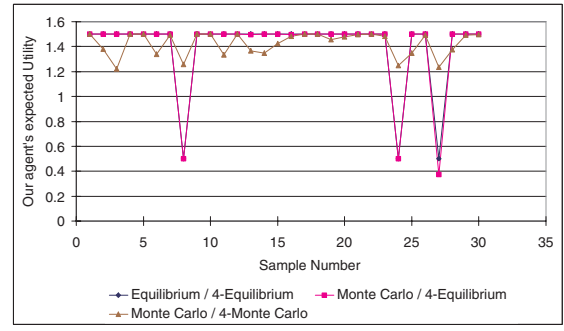


Figure 4: Our agent’s expected utility when other bidders’ valuation are drawn from a left-skewed Beta distribution.

be an effect of the tie-breaking rule, which favors our agent because it has the lowest index.

Figure 4 shows the corresponding results when the other bidders’ valuations are drawn from a left-skewed Beta distribution in the range $[1, 5]$. The graph shows the same features as the Figure 3, including the fact that the agent’s performance is quite close to the optimal play. Our agent’s averaged utility is higher against the left-skewed bidders than against the uniform bidders. This is consistent with our expectation that with left-skewed valuations, our agent is more likely to be among the highest valuers and thus will have a better chance of winning with greater surplus. Symmetric results were produced with a right-skewed distribution.

We also measured the effect of the heuristic behavior on the social welfare. Figure 5 illustrates the social welfare of this sequential auction. In all of the experiments, the Nash equilibrium always supported an optimal allocation. The heuristic strategy will (on expectation) result in an allocation very close to the perfect one. On this measure, the myopic strategy performs better than our heuristic strategy because it ensures the best allocation in each individual auction, which will result in an optimal allocation for the overall game.

Related Work

A great deal of research has addressed auctions for multiple units of homogeneous objects. See Klemperer (1999) for a broad review of auction literature, including a discussion of sequential auctions for homogeneous objects. Weber (1983) shows 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.

Anthony, et al. (2001) investigate agents that can participate in multiple online auctions. The authors posit a set of “tactics” and then empirically compare the performance of these tactics in a simulated market that consists of simultaneous and sequential English, Dutch, and Vickrey auctions. Boutilier et al. (1999) develop a sequential auction model

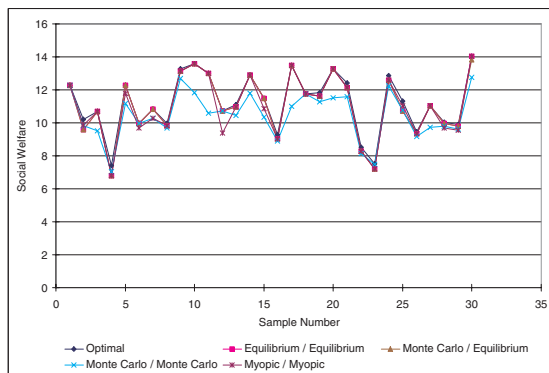


Figure 5: Social welfare averaged over the 30 sample games with valuations drawn from the uniform distribution.

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.

Monte Carlo sampling has been previously used in conjunction with games of incomplete information. Frank et al. (1997) describes an empirical study of the use of Monte Carlo sampling method on a simple complete binary game tree. They draw the discouraging conclusion that the error rate quickly approaches 100% as the depth of the game increases. However, perhaps because Frank et al. consider only pure strategy equilibrium in a two-person, zero-sum game, these negative results did not evidence themselves in our study.

Howard James Bampton (1994) 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.

Conclusion

This study represents a first step in exploring the implementation of computational game theory in a simple trading agent. By using fictitious play, the agent is able to learn a sophisticated strategy whose performance is comparable to optimal play in our tests. 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.

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. We would also like to add an aggregate buyer to the model to represent the large number of unmodeled opponents often found in these markets. Finally, we plan to explore auction sequences in which the bidders' valuations are correlated across the items,

but not necessarily identical.

Acknowledgments

This project was funded by NSF CAREER award 0092591-0029728000 to the second author. We wish to thank William Walsh, the members of the Intelligent Commerce Research Group at NCSU, and the anonymous reviewers for their insightful comments. Although not all of the reviewers' comments could be addressed in this paper, we look forward to doing so in future work.

References

- Anthony, P.; Hall, W.; Dang, V.; and Jennings, N. R. 2001. Autonomous agents for participating in multiple on-line auctions. In *IJCAI Workshop on E-Business and the Intelligent Web*, 54–64.
- Bampton, H. J. 1994. Solving imperfect information games using the monte carlo heuristic. Technical report, University of Tennessee, Knoxville.
- Boutilier, C.; Goldszmidt, M.; and Sabata, B. 1999. Sequential auctions for the allocation of resources with complementarities. In *Sixteenth International Joint Conference on Artificial Intelligence*, 527–534.
- Frank, I.; Basin, D.; and Matsubara, H. 1997. Monte-carlo sampling in games with imperfect information: Empirical investigation and analysis. In *Game Tree Search Workshop*.
- Fudenberg, D., and Tirole, J. 1996. *Game Theory*. MIT Press.
- Harsanyi, J. C. 1967-8. Games with incomplete information played by bayesian players. *Management Science* 14:159–182,320–334,486–502.
- Koller, D., and Megiddo, N. 1992. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior* 4:528–552.
- Koller, D., and Pfeffer, A. 1997. Representations and solutions for game-theoretic problems. *Artificial Intelligence* 94(1):167–251.
- Koller, D.; Megiddo, N.; and von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *26th ACM Symposium on the Theory of Computing*, 750–759.
- Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior* 14(2):247–259.
- McAfee, R. P., and McMillan, J. 1987. Auctions and bidding. *Journal of Economic Literature* 25:699–738.
- McLennan, A., and McKelvey, R. 1996. Computation of equilibria in finite games. In Amman, H.; Kendrick, D. A.; and Rust, J., eds., *The Handbook of Computational Economics*, volume I. Elsevier Science B.V. 87–142.
- Nash, J. 1950. Two-person cooperative games. *Proceedings of the National Academy of Sciences* 21:128–140.
- Weber, R. J. 1983. Multiple-object auctions. In Engelbrecht-Wiggans, R.; Shubik, M.; and Stark, R. M., eds., *Auctions, Bidding and Contracting: Uses and Theory*. New York University Press. 165–91.