A Comparison Among Bidding Algorithms for Multiple Auctions

Andrew Byde
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-300
November 23rd , 2001*

E-mail: Andrew_Byde@hp.com

We study simulations of populations of agents participating in sequences of overlapping English auctions, using three different bidding algorithms. We measure various parameters of the agents' success, to determine qualities of the underlying bidding algorithms. In particular, we show that a Dynamic Programming approach, in which beliefs regarding the opposition the agent is likely to face are built up on-the-fly, is robust enough with respect to the inaccuracy of its beliefs to outperform a greedy approach right from the moment they both start playing.

* Internal Accession Date Only Approved for External Publication

A Comparison Among Bidding Algorithms for Multiple Auctions

Andrew Byde
Hewlett-Packard Laboratories,
Filton Road, Stoke Gifford,
Bristol BS34 8QZ, UK.
Andrew_Byde@hp.com

ABSTRACT

We study simulations of populations of agents participating in sequences of overlapping English auctions, using three different bidding algorithms. We measure various parameters of the agents' success, to determine qualities of the underlying bidding algorithms. In particular, we show that a Dynamic Programming approach, in which beliefs regarding the opposition the agent is likely to face are built up on-the-fly, is robust enough with respect to the inaccuracy of its beliefs to outperform a greedy approach right from the moment they both start playing.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; I.6 [**Computing Methodologies**]: Simulation and Modeling; G.3 [**Mathematics of Computing**]: Probability and Statistics

Keywords

Multiagent-based Simulation, Electronic Commerce, Bidding and Bargaining Agents

1. INTRODUCTION

As the world of business becomes ever more closely integrated with the tools of computer science, it becomes both conceivable and (perhaps) inevitable that some decisions of economic significance which require a degree of autonomy, analysis and speed of execution that, combined, a human cannot provide, will be delegated to agents. In this paper we address the decision problem of an agent faced with purchasing a single private-value good from any of several English auctions which start at various times, may overlap, and whose termination times are uncertain. This domain is suitable because such auctions[1] are in fact conducted electronically on a daily basis, the decision problem is sufficiently complicated that a miss-placed bid could[2] lead to significant economic loss, and yet the bidding decisions may have to be made in short periods of time.

In [5] we derived a bidding algorithm based on Dynamic Programming applied to a formalization of this problem. While the derived algorithm is optimal within the set of assumptions [5] makes, it is problematic because it is based on probability distributions which have to be learned, and may not accurately reflect important dynamics of the game the agent is playing – for example, the agent assumes that the auctions in which it is playing are independent, whereas in fact they might well not be.

We now addresses this issue by conducting experiments in which populations of agents with different algorithms compete to buy goods from a sequence of English auctions. The agents use three different algorithms; one is the Dynamic Programming (DP) agent mentioned. The agents learn the probability distributions on which their bidding behaviour is based from observation of the game itself; not unexpectedly, the underlying price distribution shifts considerably from the start of the game, when nothing is known, to reach a stable state. More surprisingly, the DP algorithm is not only the clear long-term winner once the price distributions have reached a stable state, but out-performs the others right from the start.

This paper therefore provides evidence that the DP algorithm from [5] is practical, performing well even if the distributions on which its reasoning is based are only poorly known, and even if it makes assumptions, such as independence of closing-price distributions for different auctions, which do not in fact hold[3].

In the next section we review the design of the various algorithms involved. In Section 3 we describe the experiments that have been conducted, and analyse the results; Section 5 is for conclusions.

[1] In fact almost all auctions are for more complicated goods. This paper addresses the question of economic decision-making primarily; we shall expand the domain to more complicated goods and auction protocols in future work

[2] Especially in often repeated auctions

[3] This is not, of course, to say that a different, perhaps simpler, algorithm could not do better in such uncertain conditions.

2. ALGORITHM DESIGNS

In this section we give a brief description of the bidding algorithms, GREEDY, HISTORIAN, and DP[4]. For more detail, see [5]. Broadly speaking, GREEDY always bids in the auction with the currently lowest price; HISTORIAN bids in the auction with lowest *expected* price[5], if it is currently open, or otherwise in the auction with lowest price, so long as the possible return this can give in the event of a purchase exceeds the expected return from the best future auction; DP constructs a Markov Decision Process and solves it with Dynamic Programming.

To specify the algorithms further it is necessary to impose simplifying assumptions on the environment, and introduce corresponding notation.

2.1 Assumptions and Notation

2.1.1 Auction Protocol

We assume that English auctions work according to the following protocol: The auction proceeds in **rounds**. In any round except the first there is an **active** agent which is not allowed to bid – any other agent can bid, but bid prices are fixed at a certain increment above the current price. If no agents choose to bid, the good is sold to the active agent at the current price. Otherwise the auction proceeds to the next round, the price rises by the fixed increment, and a new active agent is chosen at random from those that bid.

We further assume that the different auctions' rounds are synchronized: all auctions move from one round to the next simultaneously. This gives a universal measure of time, in terms of which auctions have starting times[6], but not specified stopping times.

2.1.2 Agent Utility Model

We adopt a quasi-linear model for the **utility function** of each agent. Each agent has a fixed monetary **value** v for obtaining one or more goods. Purchasing nothing gives utility 0; purchasing one or more goods at total price x gives utility $v - x$. The agent's value v is its maximal willingness to pay for the good – if the price exceeds v, it is better not to buy the good at all. We use risk-neutral agents: if presented with a certain payment of X, or a gamble, in which n payments of $x_1, x_2, \ldots, x_n$, have probabilities $p_1, p_2, \ldots, p_n$ of occurring, the agent will prefer the certain payment X if and only if $X > \sum_i x_i p_i$.

Agents have specified starting times, and hard deadlines: goods are worth nothing after the deadline has passed, so that the utility of a set of purchases of total cost x after the deadline, is $-x$.

2.1.3 Beliefs

HISTORIAN and DP use **beliefs** about the expected closing price of each auction they are bidding in. This is a function $P_a(x)$ giving the probability that the auction a will close below price x. From $P_a(x)$, the probability that an auction will close at price x, given that the price *is* x (i.e. conditioned on it not closing at any lower price) is estimated to be

$$p_a(x) = \frac{P_a(x+h) - P_a(x)}{1 - P_a(x)}, \quad (1)$$

where h is the bid increment of the auction a. They also use beliefs about the likelihood of a bid at a given price becoming the active bid in the next round, $B_a(x)$.

2.2 Historian Algorithm

HISTORIAN uses these belief functions to calculate expected utilities in the following way: For each possible price x an expected utility if active $A_a(x)$, and an expected utility if inactive $I_a(x)$ is calculated, according to the following recursive formulas, in which h is the bid increment in the auction a:

$$\begin{aligned} I_a(x) &= (1 - B_a(x)) I_a(x+h) + B_a(x) A_a(x+h), \\ A_a(x) &= (1 - p_a(x)) I_a(x+h) + p_a(x)(v - x), \\ A_a(v) &= A_i(v) = 0. \end{aligned}$$

HISTORIAN's choice of which auction to bid in goes as follows:

1. If active in some auction, do not bid.
2. Otherwise, examine all auctions that are either open, or set to open before the deadline time, and select the auction a which maximizes $I_a(\texttt{current\_price}(a))$.
3. If a is open, bid in a.
4. Otherwise, examine all current auctions, to find the auction b with lowest current price.
5. If $v - \texttt{current\_price}(b) \geq I_a(\texttt{current\_price}(a))$ then bid in b, otherwise do not bid.

2.3 Dynamic Programming Algorithm

DP uses the same belief functions as HISTORIAN, but in a very different way. The DP algorithm uses the notions of **state** and **action**: a **state** is an assignment to each auction whose start time has passed, of either "active", "inactive", or "closed"; an action is an assignment to each "inactive" auction of either "bid" or "don't bid".

All of the agent's reasoning is conducted in advance. A table is constructed, consisting of all time – state pairs (t, s) that it might consider bidding in[7]. A pair (t, s) specifies the time, which auctions are open at time t, and in which auction the agent holds a bid. Starting from the last time step in which the agent might bid, the one before its deadline, the agent iterates backwards (in time) through the table, calculating the **expected utility** and optimal action for each of the pairs (t, s), using the following procedure:

In a given state (t, s) a given action a can lead to many potential successor states – for example, if the agent bids in an

[4] In [5], this is OPTIMAL

[5] Actually, in the auction with greatest expected utility, which is not quite the same thing, in general.

[6] It is assumed that all auctions announce their starting times long in advance: all potential auctions are known to all agents.

[7] It excludes those for which the auction price would exceed its valuation, and also, to bound computation time, removes auctions which would make the number of simultaneously open auctions go above a certain fixed threshold, which in the experiments of Section 3 was 5.

auction, it might either gain the active bid or not. The probability of each potential successor state $(t+1, s')$ given the action a can be calculated using the belief functions: For a given auction a, if the agent is inactive and does not bid, or if the agent holds the active bid, then the probability of the auction closing is estimated as $p_a(x)$ from (1); if the agent is inactive in a and bids, then the likelihood of obtaining the active bid is estimated as $B_a(x)$. Combining these probability estimates over all possible consequent states $(t+1, s')$, with the expected utilities of those states, gives the expected utility of (t, s) given the action a; maximizing with respect to action gives both the expected utility of the state itself and the action that the agent will choose in that state. We present here a pseudo-code version of the above algorithm.

```
for time ‘t’ = deadlineTime to startTime {
   for each state ‘s’ at time t {
      value(t,s) = 0
      action(t,s) = ‘do nothing’

      for each legal action ‘a’ at time t {
         newV = getUtilityForAction(t,s,a)
         if newV > value(t,s) {
            value(t,s) = newV
            action(t,s) = a
         }
      }
   }
}

getUtilityForAction(t,s,a) {
   if t = deadlineTime
      return 0

   utility = 0
   for each possible successor ‘s1’ of ‘s’
                                    given action ‘a’ {
      prob = transitionProbability(s,s1,a)

      if cost(s,s1) > 0 {

         // Here we calculate the expected cost of all
         // eventualities involving a purchase.
         //
         // Since the agent has bought a good, it will
         // ‘do nothing’ in the next time step, but
         // might yet buy more goods if it is active
         // and not outbid in some auctions.
         //
         // ‘v’ is the agent’s ‘value’ for the good,
         // as in section 2.1.2

         for each possible successor ‘s2’ of ‘s1’
                     given action a2 = ‘do nothing’ {
            prob2 = transitionProbability(s1,s2,a2)
            utility = utility + prob*prob2*(
                        ‘v’-cost(s,s1)-cost(s1,s2) )
         }
      } else {
         utility = utility + prob*value(t+1,s1)
      }
   }
   return utility
}

transitionProbability(s,s1,a) {
   prob = 1.0
   for each auction ‘b’ {
      if s(b)=‘active’
           or (s(b)=‘inactive’ and a(b)=‘no bid’) {
         if s1(b) = ‘closed’ {
            prob = prob * p_b(current_price(b))
         } else {
            prob = prob * (1-p_b(current_price(b)))
         }
      } else if s(b) = ‘inactive’ and a(b)=‘bid’ {
         if s1(b) = ‘active’ {
            prob = prob * B_b(current_price(b))
         } else {
            prob = prob * (1-B_b(current_price(b))
         }
      }
   }
   return prob
}

cost(s,s1) {
   cost = 0
   for each auction ‘b’ {
      if s(b)=‘active’ and s1(b)=‘closed’
         cost = cost + current_price(b)
   }
   return cost
}
```

As should be abundantly clear, this algorithm is of high complexity compared to the others, increasing exponentially with respect to the number of auctions that are open simultaneously (via both the number of states, and the number of actions). However, as was shown in [5], and will be further established in Section 3, even small (≤ 5) degrees of simultaneity give significant effectiveness improvements, and so in practice an agent can choose only to consider sets of auctions for which the number of auctions in which the agent plays at any time is uniformly bounded. Indeed this was a restriction that we chose to impose in the experiments we conducted.

3. EXPERIMENTAL RESULTS

To test the relative effectiveness of the various algorithms, we constructed a trading environment consisting of a sequence of auctions, whose start times (measured in rounds) are determined by a Poisson process. That is to say, the time between one auction opening and the next is a random number with density function

$$\lambda e^{-\lambda t},$$

where the variable $\lambda = \lambda_{auction}$ determines how quickly the auctions arrive: The bigger $\lambda_{auction}$, the more often auctions open[8], and hence the greater the supply of goods. In the experiments we conducted, all auctions started at price 0, and all had a bid-increment of 1.

To complement the auctions, a sequence of agents is introduced, whose starting times are also given as a Poisson process, but with a different parameter, λ_{agent}. The constant λ_{agent} parameterizes[9] the **demand** in the system. In our experiments, the agents' valuations were always selected from a uniform distribution on the interval $[10, 20]$.

In each round, each agent places its bids according to its bidding algorithm. At the end of the round, the new state

[8]The average inter-arrival time of a Poisson process with parameter λ is $1/\lambda$

[9]in conjunction with other variables, such as the valuation distribution

of play is decided: winners are selected, if appropriate, as are agents to become "active" in each auction; agents must stop trading if their deadlines have expired, and start if their start time has been reached.

In the experiments we conducted, it was assumed that all agents could observe anonymized versions of every completed auction: They could observe the closing price of each auction that had closed, and the number of bidders in each round. The same distributions were used for all auctions.

For each experimental specification we generated a large number of populations of agents and auctions, and played the agents against one-another in the auctions[10]. This randomization of the game that the agents played ensured that the algorithms were tested in a great variety of situations, facing both small and large numbers of opponents, whose values for the good were sometimes large, sometimes small, whose deadlines were un-predictably arranged, and who were sometimes using the `GREEDY` algorithm, sometimes `HISTORIAN` and sometimes `DP`.

3.1 Effectiveness

In the first series of experiments, we used the following experimental parameters[11]:

$$\begin{aligned} \lambda_{auction} &= 2/15 \\ \lambda_{agent} &= 1/3 \\ \text{Number of auctions} &= 100 \\ \text{Number of agents} &= 400 \\ \text{Agent operation length} &= 50 \end{aligned} \quad (2)$$

Figure 1 shows a graph of the distribution of utility extracted for each type of agent, in a population consisting of one third of each type of agent. These data are for trades which occurred after at least 30 other trades, but were earlier than the 90^{th} trade. As we shall soon see, by averaging in this area, we avoid end effects which will be discussed later.

The most obvious feature of this graph is that a return of zero surpasses all other outcomes in terms of likelihood. This is of course because demand outstrips supply by a factor of almost 3 - we should expect that almost two thirds of agents would not trade. In these experiments, the percentage of times that an agent did not trade at all was 70%, 61% and 51% for `GREEDY`, `HISTORIAN` and `DP`, respectively. in Figure 2 we remove non-trading agents.

Of the agents which *did* trade, the `DP` agents extracted a higher proportion of high value trades than the `HISTORIAN` agents, which in turn out-performed the `GREEDY` agents. Even though the `DP` agents sometimes made losses (negative utility), because of over-purchasing, this only happened 0.6% of the time, on average, which was more than compensated for by its larger likelihood of obtaining goods relatively cheaply: the average utility extracted for each type was 4.84, 3.95 and 3.12 respectively.

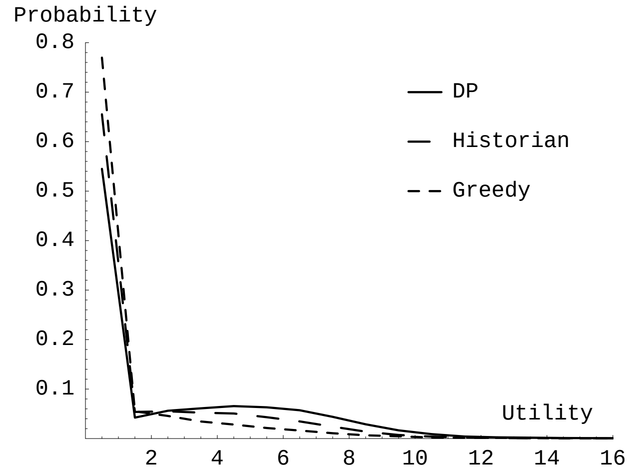

Figure 1: Distribution of utility obtained after 12–30 trades for each type of agent. Each type represents one third of the population of agents.

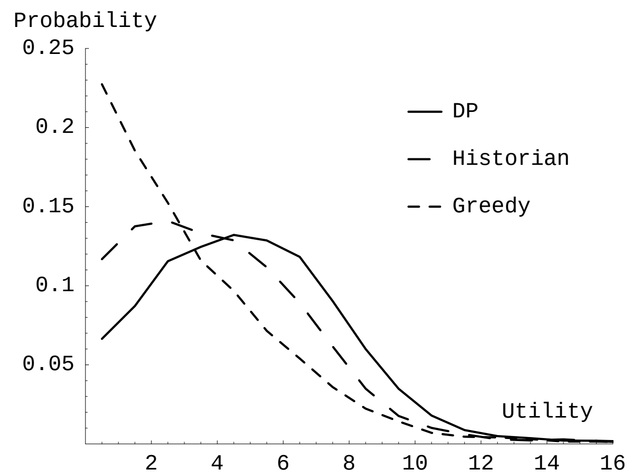

Figure 2: Distribution of utility obtained after 12–30 trades for each type of agent, ignoring agents which did not trade.

In Figure 3 the failures to trade are put back in, as are all the data points associated at the beginning and ends of the time series. As one can see, `DP` not only out-performs the other two algorithms in the region where the trading dynamics are stochastically stable, but also in the end regions when the beliefs the adaptive agents hold are inconsistent with the true supply and demand. In the next section we will examine the issue of adaptivity in greater depth.

3.2 Adaptivity

Agents adapt to market conditions by adjusting their probability distributions $P_a(x)$ and $B_a(x)$ in light of observed trades. It follows that the number of trades observed is a good measure of the "experience" of an agent, and the degree to which it has learned an appropriate price distribution. To demonstrate the movement of the price distributions with respect to time, we break down set of all trades observed according to the number of results that had been

[10]The game ends when there are either no more agents trading, or no more auctions in which to trade.

[11]The number of agents is actually irrelevant, so long as it is large enough: see footnote 12

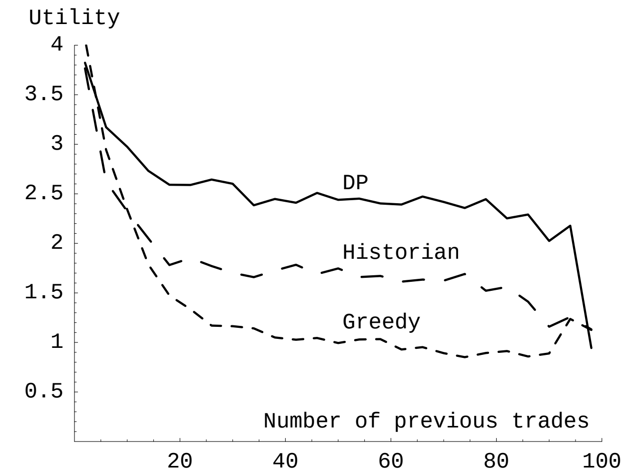

Figure 3: Distribution of average utility obtained with respect to number of previous trades. All agents are included.

announced before the purchaser started operating. A contour plot of this price density function is shown in Figure 4: For each number of previous trades, a vertical slice through the plot gives the likelihood of the next trade occurring at each price level. The darker the chart at a given price level, the more likely it is that the next trade will occur there.

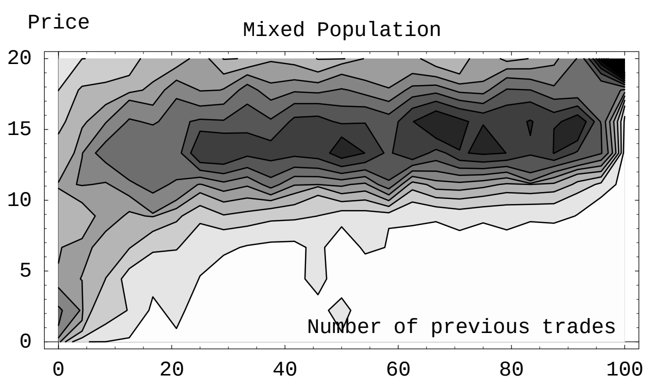

Figure 4: Contour plot of price distribution for `DP` agents in the mixed population. Darker means more likely.

Notice in particular

1. At the very beginning, a lack of agents means that auctions frequently close at low prices: the price distribution is noticeably concentrated towards the bottom end of the price scale.

2. After 30 trades the distribution is stable, up to random fluctuations in supply and demand that naturally occur over time. The learning mechanism is thus effective in adjusting agent behaviour to be consistent with market conditions.

3. At the very end, a surplus of agents combined with a dwindling supply of auctions[12] leads to very high prices: the price distribution becomes concentrated around the highest prices.

In Figure 5 we compare the utility distribution of early trades for `DP` agents in mixed and homogeneous populations. As one can see, the initial end effect is much more pronounced in the case of the homogeneous population. The reason for this is that the learning mechanism of `DP` (and `HISTORIAN`) agents tends, over short periods of time, to reinforce existing price distributions. As explained before, at the beginning of the experiment there is a paucity of both agents and auctions, an inevitable consequence of which is that the first few auctions to open have few agents to operate in them; they close at low prices, due to the lack of competition. It follows that the first data points on which the `DP` agents[13] base their estimates of closing prices are artificially low with respect to the true balance of supply and demand. As a result, the agents drop out of auctions as soon as the prices exceed these low thresholds, hoping to trade at better prices in later auctions[14]. Of course the fact

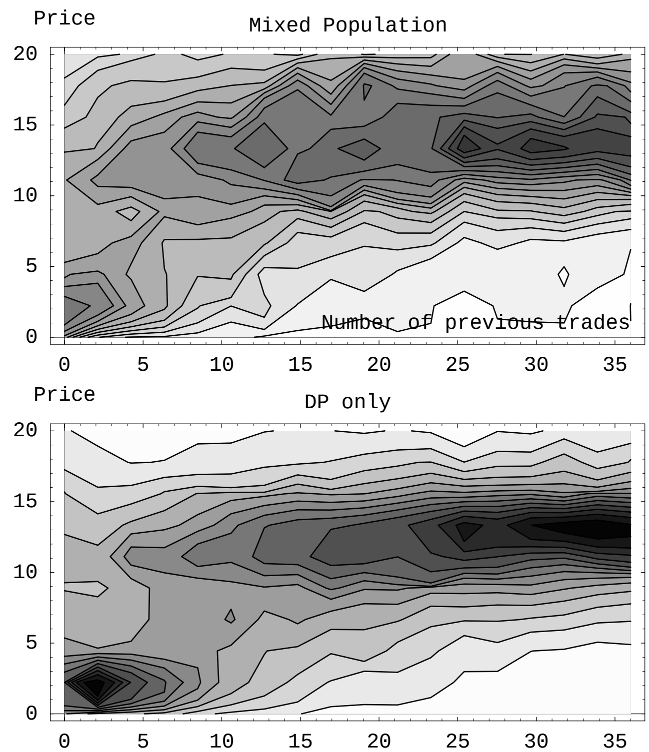

Figure 5: Contour plots of price distribution for `DP` agents in a homogeneous population, and for a homogeneous population of `DP` agents.

[12]In all the experiments we ran, agents out-lasted auctions. This was a deliberate choice: as either auctions or agents run out, the dynamics of the game inevitably change; assuring a surplus of one over the other simply forces consistency in the type of end-effect observed.

[13]The same is true for `HISTORIAN` agents

[14]The effect is not nearly so pronounced for the mixed population because `GREEDY` agents do not drop out of auctions to wait for cheaper ones.

that they drop out means that any agent left *in* via holding the active bid, will win the auction, also at a low price. This self-reinforcing belief that auctions should all close at low prices, and the corresponding consequence that they *do*, changes only as agents reach their deadlines without having bought any goods. As the deadline approaches, their algorithms stipulate staying in the last few auctions until the prices exceed their valuation, a policy which inevitably drives prices up to realistic levels.

It is clear that a similar feedback loop would be created if there was an initial abundance of agents: the agents would observe high initial closing prices due to the relatively high demand, and conclude that prices were always high. This would induce them to bid higher than they needed to, thus maintaining high prices. Prices eventually come down because agents would start winning auctions at prices lower than they had expected, and hence would adjust their reasoning to admit the possibility of winning at lower prices.

3.3 Dependence on Simultaneity

In [5] it was observed that the DP algorithm often chooses to "over-bid" by bidding in more than one auction simultaneously even though it risks buying more goods than are necessary by doing so. It was hypothesized that this behaviour is a major contributor to DPs success with respect to the other algorithms (which only ever bid for a single good). We measure the *simultaneity degree* of an agent by the average over the agent's lifespan of the number of auctions open at any time.

Figure 6 shows graphs of the distribution of earnings for agents whose average simultaneity degree is in the ranges 1 – 2, 2 – 3 and 3 – 4. When there are more auctions open, the earnings curves separate – to the advantage of DP, which obtains a higher proportion of high-value deals: as simultaneity increases, the effectiveness of DP relative to HISTORIAN increases from 7% better to 33% better, then to 41% better. This shows that DP really does profit disproportionately[15] from high simultaneity.

Interestingly, for trades when the degree of simultaneity was between 1 and 2, GREEDY seems to out-perform DP, by obtaining more low-value trades rather than not trading at all. It could be that this is a manifestation of the assumption of independence of auctions breaking down: when there are few auctions, they are strongly correlated.

3.4 Dependence on Valuation

When the data is analyzed with respect to the valuation of the agents, we find two phenomena of note. The first is perhaps to be expected: As can be seen in Figure 7, the greater the agent's valuation, the greater the relative advantage of using the DP algorithm. The intuition for this result is that the greater the valuation of the agent, the more purchase options it has, and hence the greater the leverage of using a relatively intelligent algorithm. Agents with lower valuations have few options because they are often priced out of the market, a fact that is not altered by using clever software.

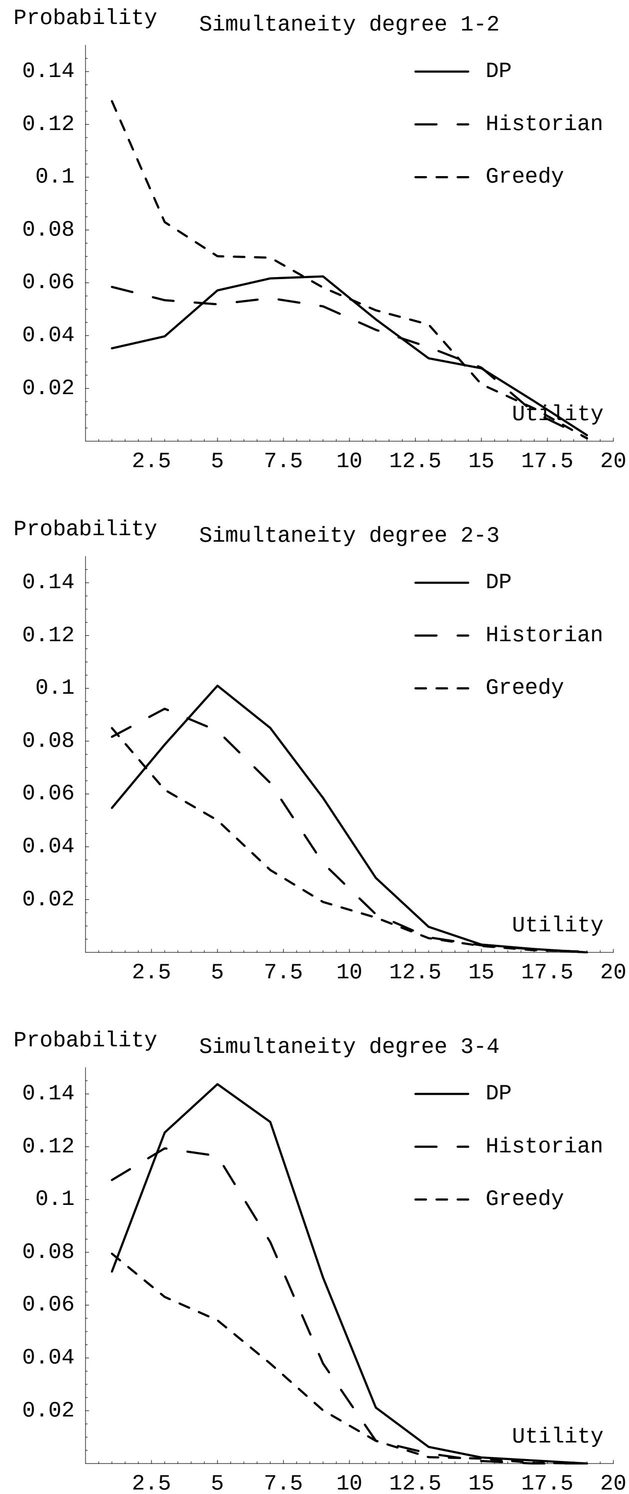

Figure 6: Dependence of relative earnings on simultaneity degree.

The second phenomenon, shown in Figure 8 is one observed in [10]: when all agents use DP they all do worse (in expec-

[15]All agents benefit from an abundance of auctions, since higher supply means lower prices. Since all agents had the same duration, abundance is clearly correlated with simultaneity degree.

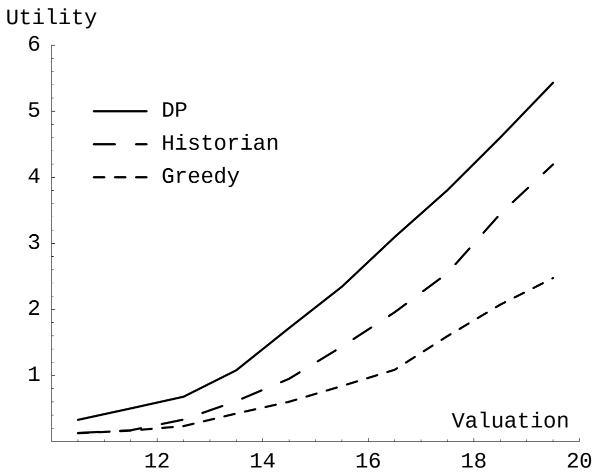

Figure 7: Comparison of earnings for each type of agent in the mixed population, indexed by agent valuation.

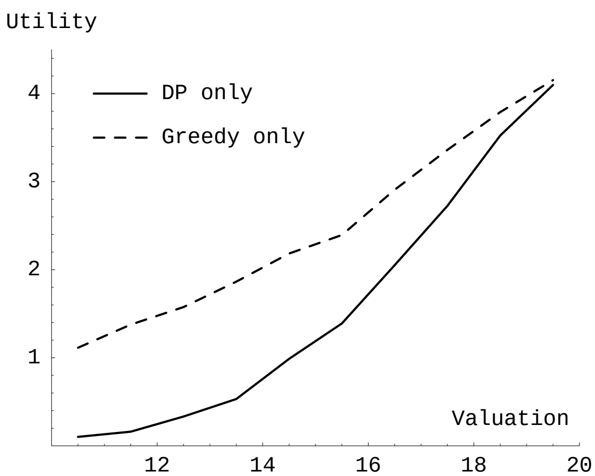

Figure 8: Comparison of earnings for populations consisting of all-GREEDY and all-DP agents, indexed by agent valuation.

tation) than if they had all used GREEDY. Notice that this is true even of the agents with highest valuations, who might be expected to do better with DP irrespective of the competition. In fact the advantage they get from using DP is eliminated by the competition using DP as well.

4. RELATED WORK

For a while now, there have existed auction house test-beds for bidding algorithms, of which the most famous is probably the Michigan AuctionBot [14]. The Spanish Fishmarket [12] provides a sophisticated platform and problem specifications for comparison of different bidding strategies in a Dutch auction, where a variety of lots are offered sequentially. Cliff at al. [6] and Preist et al. [11] present adaptive agents able to effectively bid in many-to-many marketplaces, and are the first examples of work which borrow techniques from experimental economics to analyze the dynamics of agent-based systems. Greedy agents for bidding in multiple auctions, and inspiration for the HISTORIAN algorithm come from [9], which discusses the multiple-unit case. In [10], Preist et. al present experiments involving a version of the algorithm GREEDY, though for a somewhat different auction protocol, and for use in purchasing multiple goods.

Gjerstad et. al. [8] use a belief-based modeling approach to generating appropriate bids in a double auction, combining belief-based learning of individual agents bidding strategies with utility analysis, as done here. However, it is applied to a single double auction marketplace, and does not allow agents to bid in a variety of auctions. Vulkan et.al. [13] use a more sophisticated learning mechanism that combines belief-based learning with reinforcement learning. Again, the context for this is a single double auction marketplace. Unlike Gjerstad's approach, this focuses on learning the distribution of the equilibrium price. In [7], Garcia et al. consider the development of bidding strategies in the context of the Spanish Fishmarket tournament. Agents compete in a sequence of Dutch auctions, and use a combination of utility modeling and fuzzy heuristics to generate their bidding strategy. Their work focuses on Dutch rather than English auctions, and on a sequence of auctions run by a single auction house rather than parallel auctions run by multiple auction houses. Also dealing with multiple protocols, and in a more open setting than the one considered here, is [1].

The work of Boutilier et al. [2, 4, 3] in this area is relevant because of its application of Dynamic Programming; typically a sequence of non-overlapping sealed bid auctions are considered, selling goods that may have complementarities or substitutabilities. Thus they investigate the impact of a complex utility function on agent behaviour, showing that agents can learn to coordinate their bidding choices so as to find the best distribution of goods.

5. CONCLUSIONS

In this paper we examined three bidding algorithms of increasing sophistication and computational complexity capable of bidding in multiple concurrent English auctions, and tested them by competing them against one another in simulations. In particular, we wanted to see whether DP, whose reasoning is built on probability distributions that it can only approximately know, and on assumptions - such as the independence of auction closing price probabilities - which are obviously false, could none the less outperform other algorithms that have been considered for multiple auction scenarios in the past.

We found that DP out-performs both GREEDY and HISTORIAN, despite the problems with its reasoning referred to above. Furthermore, it out-performs them even initially, when its beliefs are very crude and often wrong. We find that, as in [5], DP gains a relatively high proportion of high-value deals when the degree of simultaneity is high, presumably because of strategic over-bidding as before.

Having demonstrated that the DP approach works not only in theory but also in simple simulations, there never the less remain important questions regarding how well it will perform when other assumptions, such as the fact that auctions proceed in synchronized rounds, break down. In future work we will address the continued effectiveness of Dynamic Programming-based algorithms to simultaneous bidding in multiple auctions, when considering the following alterations to the formal framework:

1. Goods with partial common value,
2. Goods with complementarities and substitutes between them, as in [2],
3. Auctions with different and more complicated protocols, and
4. Agents with different tasks or capabilities, such as the ability to sell off excess goods.

6. REFERENCES

[1] P. Anthony, W. Hall, V. Dang, and N.R. Jennings. Autonomous agents for participating in multiple on-line auctions. In *Proc. IJCAI Workshop on E-Business and the Intelligent Web*, pages 54–64, 2001.

[2] C. Boutilier. Sequential optimality and coordination in multi-agent systems. In *Proc. IJCAI 99*, 1999.

[3] C. Boutilier, M. Goldszmidt, and B. Sabata. Continuous value function approximation for sequential bidding policies. In *Proc. UAI '99*, 1999.

[4] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *Proc. IJCAI '99*, 1999.

[5] A. Byde. A dynamic programming model for algorithm design in simultaneous auctions. In *Proc. WELCOM '01, Heidelberg, Germany*, LNCS. Springer Verlag, 2001.

[6] D. Cliff and J. Bruten. Less than human: Simple adaptive trading agents for CDA markets. In *Proceedings of the 1998 Symposium on Computation in Economics, Finance, and Engineering: Economic Systems*, 1998.

[7] P. Garcia, E. Giminez, L. Godo, and J. Rodriguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *Proc. 13th Biennial European Conference on Artificial Intelligence*, 1998.

[8] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behaviour*, 22(1):1–29, 1998.

[9] C. Preist, C. Bartolini, and I. Philips. Algorithm design for agents which participate in multiple simultaneous auctions. In F. Dignum and U. Cortes, editors, *Agent Mediated Electronic Commerce III*, Lecture Notes in AI. Springer Verlag, September 2001.

[10] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. Proc. 5^{th} Int. Conf. on Autonomous Agents, 2001.

[11] C. Preist and M. van Tol. Adaptive agents in a persistent shout double auction. In *Proceedings of the 1st International Conference on the Internet, Computing and Economics*. ACM Press, 1998.

[12] J. Rodriquez-Aguilar, P. Noriega, C. Sierra, and J. Padget. A Java-based electronic auction house. In *Proc. Second International Conference on the Practical Application if Intelligent Agents and Multi-Agent Systems*, pages 207–224, 1997.

[13] N. Vulkan and C. Preist. Automated trading in agents-based markets for communication bandwidth. In *Proc. UKMAS*, 1999.

[14] P. Wurman, M. Wellman, and W. Walsh. The Michigan internet AuctionBot: A configurable auction server for human and software agents. In *In Proc. 2^{nd} Int. Conf. on Autonomous Agents*, 1998.