

An Autonomous Bidding Agent for Simultaneous Auctions

Nicoletta Fornara^{1,2} and Luca Maria Gambardella¹

¹ IDSIA, Lugano, Switzerland
{nicoletta,luca}@idsia.ch

² Università della Svizzera italiana, Lugano, Switzerland
nicoletta.fornara@lu.unisi.ch

*Accepted for publication at Fifth International Workshop on Cooperative Information Agents (CIA-2001)*¹

Abstract. Market based mechanisms, such as auctions, will represent a widespread means of interaction for electronic commerce on the Internet. This paper is concerned with the problem of determining an efficient bidding policy in simultaneous auctions for goods that exhibit some complementarities and are substitutable. The First International Trading Agent Competition (TAC) held in Boston on July 8th, 2000 represents an interesting benchmark of this type of problems. The bidding policy of our trading agent Nidsia, who took part to the final phase of TAC competition, is described and some experiments testing our agent behavior are reported.

1 Introduction

Our research interest is to study a *bidding policy* for an agent whose task is to buy complementary and substitutable goods when such goods are sold in parallel auctions. The goal for the agent is to get the maximum utility for itself. This means that it buys and sells goods in order to maximize a given objective based on the value of goods bought and sold, and on the prices of the exchanges.

Choosing a bidding policy means deciding in which auction to bid, how much to bid and when to bid based on the goods already owned and on the information available on the running auctions. Some goods could be *complementary*, meaning that obtaining one good without another makes that good worthless. Some other goods could be *substitutable*, meaning that obtaining a certain bundle of goods can lower the value of obtaining another, or render it worthless [1].

Clearly this is a problem that everyone has to tackle when wanting to buy something using electronic commerce. However it is not easy to resolve optimally mainly because it becomes very complicated when multiple sellers offer various resources of interest.

The existing studies on this kind of problem are few. In fact most existing studies in auction theory have a different goal. Namely they try to devise a bidding policy for a team of competing auction agents, and then study what

¹ ©Springer-Verlag

social outcomes follow for the multiagent system at whole. In this case the end goal is to study what happen if you adopt that policy in an auction market to resolve resource allocation and related problems in MASs [2].

In this paper we present the bidding policy used by our trading agent, Nidsia, in the First International Trading Agent Competition (TAC) [3]. The paper is organized as follows. In section 2 we describe important aspects of the TAC game. Section 3 explains the structure of Nidsia agent. Section 4 presents experimental results testing Nidsia's behavior. Section 5 presents some ideas for future improvements and concludes.

2 Description of the Trading Agent Competition game

TAC is shopping a game in which eight software agents developed by multiple suppliers compete in a challenging market. For details on TAC visit <http://tac.eecs.umich.edu>. During the competition, the agents take part in a series of games. Each game is composed of various auctions that start at the same time, run in parallel and can last till the end of the game or finish in advance.

Each competitor is a travel agent with the goal of assembling a travel package for each one of its eight customers. These customers express their preferences for various aspects of the trip. The objective of the travel agent is to maximize the total satisfaction of its customers. The TAC server auctioneer runs at the University of Michigan and the various bidding agents connect over the Internet to the server to send bids and to update the market state.

A travel package consists of a round-trip flight, a hotel reservation in a good or bad hotel, and tickets to three types of entertainment events (Boston Red Sox Baseball, Boston Symphony or Boston Theater). A travel package is *feasible* if it contains rooms in the same hotel for every night between the arrival and departure dates. The soonest that any customer can leave is one day after arrival. An entertainment ticket is feasible and can be inserted in a feasible package if none of the tickets are for events on the same day, all of the tickets coincide with nights the customer is in town, and all assigned tickets are of different type.

From this brief description it is clear that flight tickets and the corresponding hotel rooms are complementary goods and that night rooms in the good or bad hotel and entertainment tickets on the same day are substitutable goods.

2.1 The utility function

Customers specify their preferences on a travel packages by a preferred arrival date AP; a preferred departure date DP; a reservation value for upgrading to the better hotel HU and a reservation value for each type of entertainment event BRS, SY, PH. In the travel packages pairs of arrival/departure days are equally likely, hotel reservation values are chosen for each customer uniformly in the range \$50 to \$150 and entertainment reservation values are chosen uniformly in the range \$0 to \$200. A travel package is specified by an actual arrival date AD, an actual departure date DD, a grand hotel indicator GH in $\{0,1\}$, and a ticket

indicator for each event type BT, ST, PT, each in $\{0,1\}$. The utility for each customer is:

$$Utility = 1000 - travel_penalty + hotel_bonus + fun_bonus; \quad (1)$$

where

$$travel_penalty = 100 * (|AP - AD| + |DP - DD|); \quad (2)$$

$$hotel_bonus = GH * HU; \quad (3)$$

$$fun_bonus = BT * BRS + ST * SY + PT * PH; \quad (4)$$

The final score, for the trading agent, is the sum of the utility for each customer minus the expenses and minus the negative entertainment balances. If an agent tries to sell tickets it does not have, it is assessed a penalty of 200 for each ticket.

2.2 Auction rules

Each game instance in TAC last 15 minutes. There are three different auctions: one for flights, one for hotel rooms and one for tickets. There is a separate auction for each good and each one of the 5 possible day of the travel package, up to a total of 28 parallel auctions. A detailed description of each auction type can be found in the appendix. Some important aspects of the three kinds of auctions are listed below.

Flights (8 auctions). Available flights are infinite. The price follows a random walk and does not depend on other bidders, but rather only on the auctioneer. If a bid is higher than the asking price, the bidder always get the flight. Further, since the price increase and decrease with equal probability, the expected change in price is 0.

Hotel Rooms (8 auctions). The auction for rooms is a standard English ascending auctions Mth price, with $M=16$ [4]. The final clear price depends on other buyer bids, so in a very competitive scenario prices could be very high. In this case, a bid higher than the asking price is not guaranteed to get a hotel room. Rather, depends on the behavior of other buyers. Further more, it is crucial to note that we does not know the outcome until after the auction is closed.

Entertainment Tickets (12 auctions). Entertainment tickets are sold using a continuous double auction [5]. In this case the final clear price depends on the other buyer and on the other seller bids. If a bid beats the ask-quote and comes in before other buyer bids, success is certain. If a bid is less than the bid quote and comes in before other seller bids, success is certain.

2.3 Consideration on the TAC game

Because TAC game has some properties common to many problems, it is a good benchmark for studying the behavior and the performance of a software trading

agent. Like in a real market place, goods are sold in parallel auctions, usually each auction, of a different vendor, has its own rules and characteristics and the available quantity of goods is limited. Further, like in the real life, the utility of getting some goods is related to conquering some other resources, so the trading agent's goal is to buy a bundle of goods. Still, in a real market place a buyer has to choose among various vendors where to buy substitutable goods. Keeping on with the analogies, in electronic commerce, like in the TAC game, it is not easy to know the name or important properties of the others competitors. It is not possible to repeat the same game, with the same customers' preference and the same contestants. Furthermore each auction game is too short to extract reliable statistical information about the behavior of competitors. This makes hard to introduce learning techniques in a trading agent for TAC.

On the other way, there are some unrealistic rules in the TAC game. The number of available flights is unlimited, so the corresponding auction is very similar to a shop with an expected change in price equal to zero. Another unrealistic characteristic is the absence of a penalty in the utility function, when the trading agent does not buy a feasible travel package for one of its customer.

3 Description of Nidsia trading agent

There are two main problems that the trading agent has to tackle in playing in the TAC game.

The allocation problem: given the set of goods it already holds, it has to decide how to optimally allocate them to its eight customers. The solution of this problem is relevant during the bidding phase and at the end of the game. It can be proved that the problem of finding the allocation that guarantees the maximum utility for the agent, can be formalized as an integer linear programming problem; specifically, as a weighted set packing problem, that is NP-complete. Although only few actual instances of the problem lead to a significantly longer solution times (in only 1% of 1866 problem instances, taken before the finals, the optimization took 6 or more seconds) [6]. Two of the competitors, RoxyBot and ATTac-2000, implemented an optimal strategy algorithm for allocation, while the other TAC participants, like us, used some form of greedy strategy [7].

The acquisition problem: given the set of goods it already holds, the valuation of a feasible travel package for each one of its customers, the current ask and bid price of the open auctions, decide where, how much and when to bid in the auctions. We call the result of this decision the *bidding action* of the agent and we call *bidding policy* the method that we used to choose the action in each possible state. The crucial point of this decision is that the trading agent does not know the independent value of the individual goods because it knows only the value of complete bundle of goods that forms a feasible package. Auction theory tells us when, where and how much an agent should bid only as a function of its separate valuation of a single resource given the auction mechanism [8]. Moreover, the agent does not know the final assignment of the auctions, that is the effect of its actions, therefore the problem can be considered as a decision problem under uncertainty. These considerations lead us to construct an agent bidding policy, in which bids for any resource are conditioned by the possible outcome of

the other bids. Our approach is inspired by value iteration, a standard stochastic dynamic programming algorithm, and Boutilier, Goldszmidt and Sabata's paper on bidding in sequential auctions [1].

The allocation and acquisition problems are strictly connected; in fact it is necessary to use the allocation module to evaluate the goodness of every bidding action. It is possible to simplify the approach to these two problems facing them in a distributed manner. That is, do not try to compute one collective bidding policy that satisfies all the customers' requests at once. But try to compute a bidding policy for each customer independently, then the overall bidding policy is the sum of each isolated approach. The disadvantage of this strategy is in the loose of the centralized view of the problem. The agent loses the chance to exchange goods among its own customers, during the game. Because of its simplicity, we decided to use the distributed approach to compute the bidding policy and to allocate the goods at the end of the game in a centralized manner.

3.1 Bidding Policy

Our main goal was to investigate the more general problem of studying bidding policies in parallel auctions, so our agent is not restricted to this game. Nidsia computes its bidding policy in the same way for the entire game. What changes during an instance of the competition is the probability to really obtain the desired goods. Some other competitors, instead, decide to split their bidding strategy in two parts: one strategy for the first part of the game, with the only goal of keeping open the hotel auctions and another strategy for the final part of the game, in which all competitors start to bid seriously to get the goods.

When deciding its bidding action, the trading agent considers all customers in turn; this discussion therefore applies to an arbitrary customer. The agent's state s_t is a bit vector that describes the agent's current holdings for the current customer at time t . A bidding policy is a function from states to actions, where an action a is a vector of bids, one per auction, and each bid is a price-quantity pair.

The number of possible bids (and therefore actions) is numerable infinite, if one considers all possible discrete values of price and quantity. To reduce the space of actions to a manageable size, we only consider bids in which the quantity is 1 and the price for auction i is the ask-quote $q_{t,i}$ at time t plus a fixed increment δ . With this simplification, an action a is a bit vector, where $a_i = 1$ if a bid is submitted at price $q_{t,i} + \delta$ and $a_i = 0$ if no bid is submitted.

To further reduce the complexity of our model, we focus only on auctions for travel goods (i.e., flights and hotels), and therefore, by the nature of the TAC auction mechanisms, are primarily concerned with hotel auctions.

Another problem of this kind of model is that the time needed to reach a new state, s_{t+1} , is not known, because it depends on the time needed by the auctioneer to compute another price quote or to terminate the auction and assign goods to bidders. This asynchronism among the auctions makes impossible to forecast the behavior of the system during an entire game. Because of this, in computing the bidding policy our agent assumes, at every computation of the bidding policy, that the following state, s_{t+1} , will be the final one. Under these simplifying assumptions, our agent computes an optimal bidding policy.

For each customer and given current holdings, Nidsia computes the expected utility of each of 256 possible actions, corresponding to whether or not each of the 8 possible hotel rooms is included in the action or not. Then the trading agent bids according to the action that maximizes expected utility.

The expected utility $E[U(s_t, a)]$ of taking action a in state s_t is the sum over all possible states s_{t+1} of the probability $P(s_{t+1}|s_t, a)$ of reaching state s_{t+1} times the utility $V(s_{t+1})$ of state s_{t+1} .

The quantity $P(s_{t+1}|s_t, a)$ is computed as the product of the probability of the outcomes of the bids described by the action a , taken in state s_t , that lead to state s_{t+1} . At this level also the previous bids that are still active are taken in consideration. This formulation assumes that the probability distributions among the various auctions are independent.

The probability of obtaining item i is assumed to be near 0 at the beginning of the game and near 1 at the end of the game. Specifically, for the purposes of TAC game, these probabilities were given by the following equation for a straight line: $F(t) = mt + b$, that satisfies the condition: $F(1) = 0.1$ and $F(15) = 1$, that is $m = 0.9/14$, $b = 0.1 - m$ and 15 is the number of seconds in a TAC game. The probability of failing to obtain an item at time t is $1 - F(t)$.

The utility $V(s_t)$ to be at state s_t is taken to be the reward $r(s_t)$ for being in state s_t less the cost $c(s_t)$ of obtaining the items held in this state: i.e. $V(s_t) = r(s_t) - c(s_t)$. The cost $c(s_t) = \sum s_{t,i} c_{t,i}(h)$, where $c_{t,i}(h) = 0$ if Nidsia owns item i at time t , and $c_{t,i}(h) = q_{t,i}$ otherwise, for h representing hotel rooms. The reward $r(s_t)$ is taken to be the maximum possible value obtainable among all feasible packages that include the hotels indicated by bit vector s_t .

Formally,

$$E[U(s_t, a)] = \sum_{s_{t+1}} P(s_{t+1}|s_t, a) V(s_{t+1}) \quad (5)$$

$$P(s_{t+1}|s_t, a) = \prod_i P(s_{t+1,i}) \quad (6)$$

$$P(s_{t+1,i}) = s_{t+1,i} F(t+1, i) + (1 - s_{t+1,i})(1 - F(t+1, i)) \quad (7)$$

3.2 Allocation Strategy

Our agent allocates its goods to customers according to a fixed heuristic, rather than computing optimal allocations (using e.g. integer linear programming). One minute before the end of the game, Nidsia bids on flights that coincide with the hotel room auctions that it expects to win for each client. Also at this time, the initial endowment of entertainment tickets is greedily allocated to customers. Unused tickets are auctioned off, and useful tickets currently on sale are purchased. At the end of the game, Nidsia confirms that its customers have all the necessary goods to complete their travel, and it heuristically tries to allocate any unused goods so as to satisfy as many customers as possible.

4 Results

Nidsia was eleventh in the final TAC game competition. The resulting behavior of our trading agent during one of the games is the following: during the initial part of the game, the agent bids in every auction, because the prices and the probability of success are very low. So it bids for substitutable goods to be sure to get at least one of them. Then after some minutes, the agent starts to react in accordance with the way the prices are changing and it begins to concentrate its bids on the more convenient goods. In general Nidsia prefers short travel packages, because they have a much higher probability to be obtained entirely. Further our agent does not consider the utility of the entertainment tickets when it computes its bidding policy, so is not interested to remain longer to get the fun bonus.

During the entire game the probability distribution of success in the auctions is a crucial point. If the probability of success is underestimated, like in the first minutes of the game, the agent bids in almost all auctions, to avoid to be left without some important goods but it risks to buy much more goods than it really needs. On the opposite side, if the probability of success is overestimated the agent is quite sure to get some goods. It bids exactly on the goods to form the best feasible packages and it does not take into account the risk of being left without necessary goods.

The TAC competition showed that our agent was able to produce a satisfactory performance. We then decided to carry out some experiment to study Nidsia behaviour in different situations. In particular we wanted to manipulate some factors that were not under our control in the actual competition, like for example the composition of the group of competitors.

We have run several experiments, three of which we will now describe.

Experiment 1. In the first experiment Nidsia trading agent competes against seven dummy agents. Dummy agents are provided by the TAC server. The behavior of a dummy agent is not random, but is based on a simple and fixed strategy. This scenario is the less competitive one. The results are that Nidsia wins almost always (see Table 1). The average utility of Nidsia is 2'111, while the average utility of the best dummy agent is 792. The difference between the mean utilities is statistically significant (2-tailed t -test, $p = 5.02 \cdot 10^{-14}$).

Experiment 2. In the second experiment Nidsia competes against three instances of itself and against four dummy agents. This second scenario is more competitive than the previous one. The result is that the 4 Nidsia agents are placed almost always in the first places in the race (see Table 2). The average utility of the first Nidsia agents is 2656. It is possible to notice that in these games there are less negative scores because the four Nidsia agents exploit and share the available resources better. In this experiment, like in the previous one, the difference between the mean utilities, of Nidsia and dummy sample, is statistically significant (2-tailed t -test, $p = 6.71 \cdot 10^{-15}$).

Experiment 3. In the third experiment two different instances of Nidsia compete against six dummy agents. In this scenario we want to compare the

game n°	7845	7848	7856	7859	7863	7867	7868	7869	7870
1° Ndsia	3577	1673	3047	2420	2297	2226	2491	1757	2287
2° dummy	85	526.5	1993	360	1887	1910	356	589	1034
3° dummy	-1047	289.5	1957.5	-728	1526	1886	-807	-190.5	716.5
4° dummy	-2649.5	175.5	-150.5	-798	-1979	1820	-1269	-264.5	520.5
5° dummy	-3192	-3278	-2365	-1021	-2001	1394	-1979.5	-618.5	-1937
6° dummy	-3296.5	-3816	-3326.5	-1074.5	-2113	-1353.5	-3003	-2633	-2591.5
7° dummy	-3752	-5875.5	-3844.5	-2357	-4377	-3365	-4532.5	-2912.5	-3086
8° dummy	-4325	-5908	-4542	-4323.5	-4769	-3454.5	-5012	-3240	-5715.5
game n°	7884	7926	7936	7937	7941	7947	7847	average	
1° Ndsia	1513	1140	2521	3504	2405	2252	4° Ndsia	-1330	2,111
2° dummy	1405	-1009	-792	1869	947	1309.5	1° dummy	207.5	792
3° dummy	618	-3909	-2079	1520	-2996	1307	2° dummy	-1042	-186
4° dummy	241	-4277	-4642	903	-3133	1002.5	3° dummy	-1174.5	-980
5° dummy	-2187	-4330	-4679	-397	-3526	-598	5° dummy	-1402.5	-2,007
6° dummy	-3233	-4819	-4956.5	-1381	-3776	-2625.5	6° dummy	-1511	-2,844
7° dummy	-3794	-5322	-5897.5	-4325	-3957	-4183	7° dummy	-4276	-4,116
8° dummy	-4084	-6089	-6317	-6339	-4907	-6960.5	8° dummy	-4442.5	-5,027

Table 1. Results of the games in experiment 1

game n°	7838		7846	7858	7864	7871	7873	7874	7880	7898		
Ndsia	2°	-122	1°	4095	2366	1986	3135	2640	2900	2802	2258	
Ndsia	3°	-183	2°	2676	2365	1316	2884	1606	2610	2247	1580	
Ndsia	6°	-3830	3°	2309	2204	-273	1843	1470	2022	1912	930	
Ndsia	8°	-6096	4°	2169	963	-1977	1445	1126	1258	1876	628	
dummy	1°	1488	5°	2010	-2567	-2545.5	555	-1248	-2542	-2288	-2976	
dummy	4°	-535	6°	664	-3521.5	-2688	-1425	-2655	-3517	-2473	-3799	
dummy	5°	-1479.5	7°	-182	-3730	-3086.5	-1791	-3255	-4231.5	-4124	-4829	
dummy	7°	-4446.5	8°	-1237	-5112.5	-3703	-2398	-3853	-4890.5	-6297	-6066	
game n°	7881	7899		7882		7883		7887		7890	average	
Ndsia	1°	2453	3440	1°	3532	1°	3471	1°	2896	1°	1982	2656
Ndsia	2°	1739	1925	2°	2498	2°	3136	3°	1993	2°	1734	2008
Ndsia	3°	1541	221	3°	1536	6°	768	5°	1764	4°	1453	1058
Ndsia	7°	-1866	-2536	5°	-21	7°	754	6°	1120	5°	1232	5
dummy	4°	-459	104	4°	999	3°	2903	2°	2174	3°	1615	-185
dummy	5°	-603	-77	6°	-2398.5	4°	1713	4°	1822.5	6°	-420.5	-1328
dummy	6°	-1524	-501.5	7°	-2614.5	5°	1448	7°	959	7°	-2441.5	-2092
dummy	8°	-2881	-5648.5	8°	-3738	8°	-3196	8°	-2933.5	8°	-3767	-4011

Table 2. Results of the games in experiment 2

performance of Nidsia agent with the performance of an agent equal to Nidsia, but that computes a different probability of success for each auction. We call it Nidsia2. Our idea is to exploit the information, about the current game, that we can extract from the on going of prices in the market. In Nidsia2, if an auction is very competitive, meaning that its price-quote rises very quickly, the probability of success is lower than in a less competitive auction. It results that Nidsia2 is a risk adverse agent. Precisely, for each item i the probability of obtaining it, is given by the following equation for an exponential function:

$$F'(t, i) = e^{-((\Delta price / \Delta time) / \Delta rate_{max})} F(t, i) \quad (8)$$

where

$$\Delta rate_{max} = 100 \quad (9)$$

The probability of failing to obtain an item at time t is $1 - F(t, i)$. In Figure 1 three different kinds of price trends are shown and in Figure 2 are shown the corresponding probability trends. In this experiment Nidsia competes against

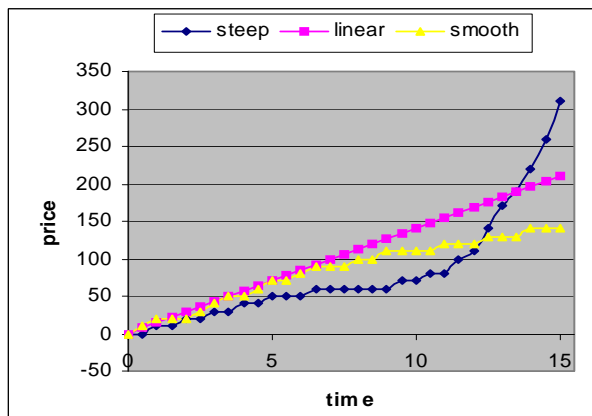


Fig. 1. Some price trends

Nidsia2 and against 6 dummy agents. The two Nidsia agents are placed first and second in almost all the test games, but Nidsia2 does not perform always better than Nidsia. A possible explanation is that the information that is possible to exploit during a single game, is not enough to have a more successful agent. The difference between the mean utilities of Nidsia and Nidsia2 sample, is not significant (2-tailed t -test, $p = 0.46$).

4.1 Comments

The main problem of our trading agent is that when it decides (two minutes before the end of the game) on which flights to bid, relies on hypothesis that

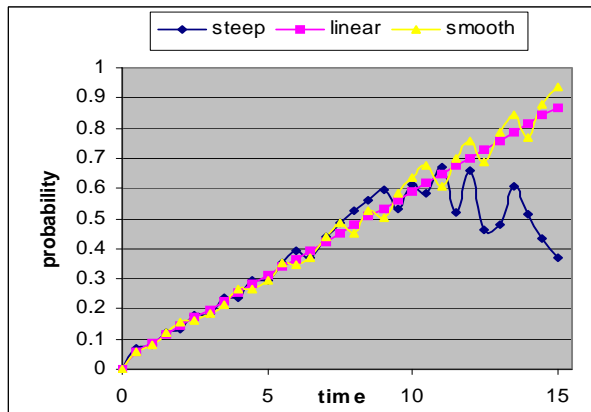


Fig. 2. Probability trends corresponding to above price trends

all its active bids will be successful. But if it overestimates the probability of success, especially in some high competitive game, it gets stuck in a situation where it has the flights, that usually are very expensive, but it does not have the necessary rooms to compose a feasible package. We notice also that the expected utility of the best action could be near the expected utility of some other actions and we decide to choose the action that bid for the little number of goods.

With regard to the allocation module, at the end of the game, Nidsia uses a heuristic strategy, although during the official competition it gets 95% of the optimal utility. As concern the distributed approach to tackle the acquisition problem during the game. It is possible to notice that it is not very relevant, because in many games of the TAC competition, few goods are bought during the first and middle part of the game, so there are few goods to be redistributed among the customers.

TAC is not a zero-sum game, but there is high competition for hotel room resources. In some games it is possible to notice that trading agents compete strongly on a special good, so the price rises and the utility average decreases. While in some other games there is less competition and the utility average is higher.

5 Conclusion

As we already said, our trading agent's algorithm was not tailored to the particular auction mechanisms of TAC, but rather is more general in its applicability to combinatorial auctions of substitutable and complementary goods. As a result, the implementation of our algorithm required some strong simplifying assumptions. Nonetheless, Nidsia's overall performance illustrates the promise of this general method.

In a future work it would be interesting to compare a *centralized* with a *distributed* approach. The former consists in computing the bidding policy consider-

ing all the customers' requests at one. The latter approach consists in computing separately a bidding policy for each customer, as Nidsia did.

Further, it could be interesting to improve the agent using a broad the set of possible value of bids in the auctions. Some other improvements consist in trying to introduce some learning feature in the trading agent. For example it could learn some of the parameters that it uses, like the instant when it starts to bid for flights, the amount to increase the ask price when bid in auctions. Or harder, try to learn some characteristics of the other competitors, like their preferences or if they are risk neutral, risk averse or risk seeking.

Acknowledgements

We thank Marco Colombetti for advice and for helpful comments on a draft version of the paper.

We thank the TAC team for their continuous collaboration and helpfulness.

We thank also Amy Greenwald for useful suggestions.

References

1. Craig Boutilier, Moises Goldszmidt and Bikash Sabata. Sequential auctions for the allocation of resources with complementarities. Proceedings of IJCAI-99, 1999
2. T. W. Sandholm. Distributed rational decision making. In G. Weiss, editor, Multi-agent Systems, pages 201–258. The MIT Press, 1999
3. Michael P. Wellman, Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Shou-de Lin, Daniel Reeves, and William E. Walsh. A trading agent competition. In Press, 2001
4. Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. Decision Support Systems, 24:17-27, 1998
5. D. Friedman and J. Rust, editors. The Double Auction Market: Institutions, Theories, and Evidence. Addison-Wesley, New York, 1993
6. Peter Stone, Michael L. Littman, Satinder Singh, and Michael Kearns. ATTac-2000: An adaptive autonomous bidding agent. In Proceedings of the Fifth International Conference on Autonomous Agents, 2001. To appear, available at <http://www.research.att.com/pstone/papers.html>
7. Amy Greenwald and Peter Stone. Autonomous bidding agents in the trading agent competition. IEEE Computing, 2001. To appear. Available at <http://www.research.att.com/pstone/papers.html>
8. R. Preston McAfee and John McMillan. Auctions and bidding. Journal of Economic Literature, 25:699-738, 1987

Appendix

A schematic description of the three auctions type.

Flights

number of sellers: single seller auction;

number of auctions: one auction for each day and direction;

time of clear: the auctions will clear continuously;
clear price: if the current quoted price is less than your bid, you will get the ticket at the current price. If the current quoted price is greater, your bid will sit in the auction until the price becomes less than or equal to your bid, at which point you will get it for the price you bid. Of course, this may never happen if the price never goes that low. Matching is based on earliest submission time;
time of price quote: quotes are continuous;
price quote: the ask quote is set according to a stochastic function. The process used to update flight prices is a random walk, starting between \$250 and \$400 and perturbed by -\$10 to \$10 every 30 to 40 seconds. Prices will always be in the range \$150 to \$600. All distributions are uniform;
availability of goods: unlimited;
bid admittance conditions: all buy bids are accepted, multi-point buy bids are allowed;
bid withdrawal conditions: bids can be withdrawn at any time if they have not transacted;
reservation price: the ask quote;

Hotels

auction type: standard English ascending auctions, Mth price;
number of sellers: single seller auction;
number of auctions: there are two hotels in Boston: the Boston Grand Hotel and Le Fleabag Inn, so there is one auction for each combination of hotel and night.
time of clear: it will clear at the earliest, 1 minute after the game has started or at latest when the game ends. It might clear earlier if there has been no activity (no bids) for a random chosen period of time;
clear price: is the price of the lowest winning bid, matching is based on earliest submission time;
time of price quote: continuous quotes;
price quote: the ask quote is the 16th highest price;
availability of goods: 16 rooms;
bid admittance conditions: must beat quote, multiple buy points in bid allowed;
bid withdrawal conditions: no withdrawal, but since the hotel auctions are ascending, once a bid is "out of the money" (i.e., the ask price is above the offer) it is effectively decommitted;
reservation price: no reserve price, minimum bid is \$0;

Entertainment Tickets

auction type: continuous double auction (CDA);
number of sellers: multiple seller;
number of auctions: there will be one auction for each event-night combination.
time of clear: continuous clears;
clear price: when a new bid is processed, the auction check whether the offered price would match the lowest (best) existing sell bid, and vice versa. The transaction price is the price that was put in first;
time of price quote: continuous quotes;
price quote: the ask quote is the price an agent would have to bid over in order to place a winning buy bid (the least of the sell bids). The bid quote is the price an agent would have to bid under in order to place a winning sell bid (the biggest

of the buy bids);

availability of goods: depends on the seller;

bid admittance conditions: any type of bid allowed;

bid withdrawal conditions: bids in the entertainment auctions can be withdrawn at any time if they have not transacted;

reservation price: no reserve price;