CS886 – Assignment 2

Stefan Buettcher
#20161244

1. The Trading Agent Competition

The first Trading Agent Competition (TAC) took place at the International Conference on Multi-Agent Systems (ICMAS) 2000 in Boston and attracted 22 participating teams from academia and industry. It was the first e-commerce trading competition in which autonomous agents were pit against each other to acquire certain goods for their customers.

In the first TAC, every participating group had to develop a simulated travel agent that had to buy three different types of goods for its clients: airline tickets for flights from the virtual city of TACTown to Boston, hotel rooms, and tickets for certain entertainment events. These travel goods were traded in 28 simultaneous auctions.

The agents had to arrange a travel program for their customers within a given time frame of 5 days. Their goal was to buy the goods needed to fulfill their clients' wishes while spending as little money as possible. After the auctions had ended, the agents had to compile travel packages for all of their customers, i.e. distribute the purchased items among their customers, with respect individual preferences.

In every instance of the game, 8 agents were competing, and each of them had to serve 8 customers. A game instance lasted 15 minutes, and there were several thousand instances during the seeding rounds, which were used to select the 8 finalists.

1.1 Auctions at the first TAC

In order for the clients to be able to travel to Boston at all, an agent had to buy *flight tickets*. In total, there were 8 flight ticket auctions for flights from TACTown to Boston and back (flights to Boston for days 1-4 and return flights for days 2-5). The flight auctions were continuously clearing with prices between \$150 and \$600 and performing a random walk: every 40 seconds, the price changed by a value X uniformly selected from $[-10\$, +10\$]$. There was an infinite supply of airline tickets.

Accomodations were sold at *hotel room auctions* in which the agents could bid for rooms in the more expensive Grandhotel or in the less expensive Le Fleabag Inn. In each of these auctions, 16 rooms for a particular night in one of the two hotels could be purchased. The rooms were sold in open-cry, sixteenth-price auctions, which means that each of the top 16 bidders got a room, and all of them had to pay the sixteenth-highest price. Agents were allowed to bid for more than one room at the same time. A special feature of the hotel room auctions was that they could close suddenly after random periods of inactivity.

The third type of goods, *entertainment tickets*, came in three different flavors: for each of the four nights, the agents could buy tickets for Baseball, classical concert, or musical. In these auctions, the agents acted both as vendors and as sellers. Each agent was given an initial amount of entertainment tickets, and the agents could trade among each other.

1.2 Performance Evaluation / An Agent's Utility

As mentioned earlier, the agents in the competition had the goal of fulfilling their clients' wishes and doing so with as little money as possible. More formally, the performance $P_{A,C}$ of an agent A with respect to a particular client C is

$$P_{A,C} = U_{A,C} - cost_{A,C},$$

where $U_{A,C}$ is the utility of the travel package that the agent has compiled for the client C and $cost_{A,C}$ is the total cost of that package. The utility of a given travel package is

$$U_{A,C} = 1000 - penalty_{flight} + bonus_{hotel} + bonus_{entertainment},$$

where the penalty and bonus values depend on the client's preferences regarding the intended flight dates, the client's valuation of a room in the Grand Hotel (vs. Le Fleabag Inn) and the valuation of a certain entertainment event. If a travel package is infeasible, its utility is $U_{A,C} = 0$. There are several constellations that could make a packet infeasible, e.g. if the flight to Boston is later than the return flight to TACTown, or if the client has to change his/her hotel during the stay in Boston.

The total score of an agent in a particular instance of the game is the sum of the $P_{A,C}$ values with respect to each of the clients the agent has to serve in that game instance.

1.3 Later Development

While the Trading Agent Competition was novel when it was carried out for the first time in 2000, things did not remain unchanged during the subsequent competitions. One of the major problems for the agents in the TAC-2000 was the allocation of the purchased items, i.e. the distribution of the travel goods among the customers with the goal of maximizing the total utility. While this is an NP-complete problem (details are given by Rothkopf et al. [RPH1998]), it turned out that it could be solved for the problem instances that the agents had to deal with in the competition. In fact, two of the competing agents were able to compute the optimal allocation in every game instance. Therefore, new challenges were developed, some of which are presented in this section.

In 2001, the duration of each round of the game was reduced from 15 to 12 minutes. Furthermore, the rules of the flight auctions were changed so that the ticket prices performed a random walk that was determined by changing the price every 30-40 seconds by a value X uniformly selected from the interval $[-10, x(t)]$, where

$$x(t) = 10 + \frac{t \cdot x_a}{720},$$

t is the time (in seconds), and x_a is an auction-specific parameter, uniformly chosen from the interval $[0, 80]$. The rationale behind this change was to let the expected ticket price grow during the auction, whereas the expected price of a ticket stayed constant in 2000 (because X was chosen from $[-10, 10]$), which allowed the agents to remain inactive until the very end of each round. [WGS2003]

The sudden closing of the hotel room auctions was refined, too, because the agents successfully prevented the auctions from closing in 2000 by increasing the bids in \$1 steps. In 2001, after 4 minutes, every minute a randomly selected hotel auction closed, encouraging early bidding. Since the allocation problem was supposedly well understood by 2001, the agents did not have to compute the optimal allocation any more. Instead, the server reported it after the game.

In 2003, the Trading Agent Competition was split into two tracks: The original TAC was relabelled to *TAC Classic*, and a new *TAC SCM* (Supply Chain Management) was introduced. The general goal of the new TAC SCM and its rules are described by Arunachalam and Sadeh [ARA2003].

2. Key Issues in the Trading Agent Competition

In TAC-2000, the major issue, which determined the outcome of the competition, was the solution of the allocation/completion problem. In general, the agent that had the better allocation strategy won the game. Among the TAC-2000 finalists, there was only one exception from this rule.

Rank	Agent	Optimal allocation
1.	ATTac	13/13
2.	RoxyBot	13/13
3.	Aster	9/13
4.	UmbcTac	7/13
5.	ALTA	2/13
6.	DAIhard	1/13
7.	RiskPro	1/13
8.	T1	7/13

Table 1. Allocation quality of the 8 finalists in TAC-2000. [STO2001]

A second issue in 2000 was to try to place bids as late as possible in order to have as many possibilities as possible during the final seconds of the game. This goal could be achieved easily because

- the expected price of the flight was constant,
- and the agents could prevent the hotel auctions from closing early by increasing their bids in very tiny steps.

Since there was no necessity for most of the agents to reveal their true valuation of the goods offered until the very end of the game, the prediction of future prices did not play an important role in 2000.

This changed in 2001, when the expected price of flight tickets was increasing over time and the hotel room auctions were subject to early closing in a random order. Now, agents had to decide if it was better to buy flights ticket early (at a lower price) or to delay the purchase of the tickets in order to obtain additional information on the hotel rooms they could probably obtain, causing the risk of higher flight ticket prices (due to increasing expected prices).

Furthermore, since the measures taken by the organizers to encourage the participants to make their agents bid for hotel rooms early were so aggressive, much information about other agents' valuations of certain rooms was available that could be used in order to predict the future price of the rooms.

Finally, the problem of computing the optimal allocation after the end of the game was no longer present because the server solved the task, but still the agents had to solve the more general completion problem throughout the competition to determine what items to bid for and at what prices.

All in all, the main problem in TAC-2001 was the prediction of the auctions' closing prices, since with the knowledge of the closing prices, an optimal strategy can be easily developed: Determine the optimal allocation G^*, given the closing prices, and bid high for all items $I \in G^*$.

3. ATTac-2001, Top Scorer in TAC-2001

This section presents ATTac-2001, the successor of 2000's winning agent ATTac. The allocation problem that arised in the Trading Agent Competition was solved by ATTac-2001 using the same mixed-integer LP approach that had already successfully been employed by ATTac-2000. Details on their solution to the problem can be found in [SLS2001].

The most important improvement over its predecessor ATTac-2000 was ATTac-2001's ability to predict auctions' closing prices. As mentioned before, if an agent knows the closing prices of all 28 simultaneous auctions, it can compute the optimal allocation of travel goods, buy them, and achieve maximal score. Unfortunately, the eventual prices of the travel goods are unknown and therefore have to be estimated. A realistic assumption is that the accuracy of a price prediction algorithm increases as more information is gathered. Thus, closing prices can be predicted better as more time elapses. However, a problem regarding the flight tickets and the hotel rooms immediately arises: Should the agent wait longer in order to obtain more information on that the price prediction could be based, or should it buy a certain good right away and thus avoid a higher price?

ATTac's answer to this problem was as following: At any point in time, let G^* be the optimal allocation of goods to the customers with respect to set of items the agent has already bought and to the expected prices of all items not owned so far. Then, disregarding entertainment tickets for the moment, ATTac's global strategy is:

1. Right at the beginning of the game, compute G^* and buy all flight tickets $f \in G^*$ for which the expected cost of postponing the purchase in bigger than the expected benefit of postponing it.
2. 1 minute before each hotel auction closes, compute G^*. Deal with the flight tickets as in (1). Bid marginal values for hotel rooms, with respect to currently owned items and expected hotel room purchases.
3. In the last minute of the game, buy all flights $f \in G^*$ that have not been purchased so far.

Of course, in order to follow this strategy, ATTac had to know the expected prices of all items not currently owned, the expected cost of postponing a purchase, the benefit of postponing it, and the expected hotel room purchases.

Price prediction

In order for ATTac's strategy to work, two types of prices have to be computed: flight ticket prices and hotel room prices. The prediction of flight ticket prices is fairly easy considering the fact that the ticket prices are adjusted every 40 seconds with respect to the rule:

$$x_{new} := x_{old} + X(t),$$

where $X(t)$ is randomly chosen from $[-10, x(t)]$, $x(t) = 10 + \frac{t \cdot x_a}{720}$, and t is the time. Given values $X(t)$, a confidence interval for x_a can be computed, which can be used to predict future ticket prices. For instance, if $X(240)$ is 23, it can be computed that the probability that $x_a \in [42, 76]$ is at least 83%. Obviously, as more $X(t)$ values become known to the agent, the accuracy of guessing x_a increases, and therefore the prediction of future flight prices increases as well.

The prediction of hotel prices is more difficult, since it seems to involve modelling the other agents in the system. However, the designers of ATTac-2001 chose a different approach that allowed them to predict hotel prices without having to model other agents. They regarded the closing price of a certain hotel room as a random variable that was to be estimated conditional on the current knowledge of the agent about the state of the game.

For the purpose of predicting the expected value of this random variable, ATTac gathered the following features from earlier games:

1. time remaining in the game;
2. current ask price / actual selling price of hotel rooms;
3. closing times of hotel auctions;
4. flight prices.

The reason for choosing features 1 and 2 is obvious. Regarding feature 3, it can be noted that the closing time of a certain hotel auction influences the expected closing price, since room prices are non-decreasing. Thus, it can be expected that the price is higher if the auction closes later. One reason to take flight ticket prices as a 4th feature is that, for instance, a very high price for return flights on the last day makes hotel rooms for the last night less attractive because the two goods are equivalent in the sense that neither of them can be used without the other.

Assuming that the price of hotel rooms can be determined by these features with sufficient accuracy, the agent used the data to train a predictor that can predict per-minute increases of hotel room prices, given a point in the feature space.

In the actual implementation, ATTac-2001 did not only predict expected values of hotel rooms, but was using entire price distributions instead. This was necessary because the agent did not know the closing times of the hotel auctions (feature 3), but only the probability distribution over the closing times. However, the mathematical techniques employed for the distribution-based approach are too complex to be presented here. An explanation of the method is given by Stone et al. [SSM2002].

Expected hotel room purchases

In order to follow the strategy described, ATTac-2001 had to know which hotel rooms it eventually would purchase. The agent fulfilled this requirement by ensuring that it bidded high enough for all hotel rooms it was bidding for, with respect to the predicted closing prices. Since there are 16 rooms sold per hotel auction (which are sixteenth-price auctions), the designers of ATTac-2001 assumed that, by bidding the predicted closing price, an agent can usually get 2 rooms (called the "fair share" for 8 agents and 16 rooms). In order to purchase more than 2 rooms, the agent has to increase the bidding. Under the supposition that biddings are increasing exponentially from the 16th-highest to the highest bid, the n-th-highest bid for a particular auction is

$$b_n = p \cdot c^{16-n},$$

where p is the expected closing price of the auction. ATTac's developers found out that $c = 1.35$ worked well. So, for instance, if ATTac wanted to purchase 3 rooms of the same type, its bid was $b_14 = p \cdot c^2$.

Expected cost/benefit of postponing flight purchases

The key decision problem in ATTac-2001 was: Under what circumstances should flight tickets be bought immediately, when should the commitment to a particular flight be postponed? This problem can be reduced to decision whether the expected benefit of postponing the purchase decision is bigger than the expected cost of the waiting. This decision is made by first computing the utility u_{buy} of the optimal allocation G^* under the assumption that the agent already owns a particular flight ticket f (bought at the current price) and then computing the expected utility u_{wait} under the assumption that the agent bought a different flight instead. The expected utility u_{wait} is computed by sampling possible future price over the probability distribution calculated by the hotel price predictor.

As a second step, the cost c_{wait} of postponing the decision by taking the average of the expected ticket costs over the next m minutes is computed. The initial value $m = 3$ was changed by the ATTac team to $m = 2$ during the TAC-2001 finals.

Knowing u_{buy}, u_{wait}, and c_{wait},

$$v = u_{buy} - u_{wait} + c_{wait}$$

can be computed, and the agent immediately buys the flight ticket in question if and only if $v > 0$.

Entertainment tickets

The trading of entertainment tickets remained almost unchanged from the TAC-2000 rules. The only modification in TAC-2001 was that instead of initially distributing the tickets among the agents in a uniform way, every agent was only given tickets of two different types per day in order to motivate the trading of tickets between the agents.

ATTac's general approach to the problem of trading tickets was very similar to its behavior regarding flights and hotels: The difference between the agent's utility for either having or not having a particular entertainment ticket is calculated and then compared to the current offers/bids for this ticket, and the buy/sell decision is based upon the outcome of this comparison. Because the entertainment tickets were not critical to the success in TAC-2001, the exact details are omitted.

4. Competition Results and Discussion

ATTac-2001 placed second in the 2001 Trading Agent Competition. However, in the 2001 finals, there were only 24 games, which caused the risk of unequally difficult client preferences between the different agents. According to Stone et al. [SSC2002], this was the case. After TAC-2001 was over, the TAC team at the University of Michigan analyzed the results of the TAC seeding rounds and developed a modified scoring function which takes into account these differences in client preferences (column "Handicapped" in Table 2). According to this scoring function, ATTac placed first in TAC-2001.

Agent	Score	Handicapped
livingagents	3670	4094
ATTac-2001	3622	4154
whitebear	3513	3931
Urlaub01	3421	3909
Retsina	3352	3812

Table 2. Scores of the top 5 agents in the TAC-2001 finals (according to [SSC2002]).

So, in general, the agent performed very well in the TAC-2001 competition. Especially, using essentially the same solution to the allocation/completion problem as in 2000, ATTac was able to compute optimal allocations of travel goods. Since, for performance reasons, it uses only an estimation of the value of the optimal allocation, this is not completely true. But the agent was able to do better than many other agents in the competition, which is surprising because the solution to the allocation problem should have been known to all TAC-2001 participants.

The price predictor, which is the critical part of the agent for the strategy chosen by ATTac-2001's designers, is able to adapt to a changing environment. This turned out to be necessary during the finals, since the predictor was initially trained using price samples from the seeding rounds, in which the hotel closing prices were substantially different from those in the finals. ATTac was able to adapt to this change, although it did so very slowly. Its scores began to increase in the 16th out of 24 games, which was too late to place first (i.e., first according to the original scoring function).

Another shortcoming of the agent is its inability to learn from other agents. In their retrospective, ATTac-2001's developers point out that the highest-scoring agent, Livingagents, pursued a parasitic strategy that mainly relied on the fact that the competing agents are trying to keep the hotel prices low (and hotel room auctions are 16th-price auctions) – without the "help" of the competing agents, it could have paid very high prices.

This observation is correct: According to the developers of Livingagents 2001 [FDL2001], their agent determines the best journey for each of its 8 clients right at the beginning of every game instance, buys the flight tickets immediately, and tries to purchase the necessary hotel rooms for the clients. The calculation of the best journey is based on the current prices of flight tickets and the average hotel prices from previous games. Because, generally, the hotel prices were low during the finals, Livingagents did not lose much money for hotel rooms, while it saved much money by buying the flight tickets immediately and at lower prices than those ATTac-2001 paid for the same tickets.

This shows that ATTac was neither able to learn from other agents (from Livingagents, in this case), nor was it able to react to the behavior of the competing agents in an adequate way. If hotel closing prices stay low over a couple of games, this should cause a change in the agent's strategy.

Improvements

Because ATTac-2001's design is in general very good and the designers give a good discussion on the agent in their paper, I have few suggestions for improvement.

The first point concerns the agent's inability to adapt to a changing environment in a timely manner, as shown in the TAC-2001 finals. This inability is mainly due to the process of training the descriptors, which, according to [SSM2002], does not give a greater weight to experiences that are more recent than other experiences. This problem could be fixed by implementing different predictors, trained on different data, for example the first predictor trained on the most recent 4 games and the second trained on all older games, and assigning appropriate weights to their predictions.

Unfortunately, it is not clear if the data gathered in as few as 4 games is sufficient for training ATTac's price predictor. However, even if this approach turns out to be infeasible, it should be possible to modify the algorithm that is used to train the predictor in such a way that it gives a greater weight to more recent experiences.

The second suggestion is a meta suggestion related to the goal of placing first in the competition rather than serving the customers in the best possible way. If ATTac notices that one of the other agents in the competition follows a parasite strategy and is continuously scoring higher than ATTac, it should start taking countermeasures, for instance by increasing hotel prices in a way that harms the competing agent more than ATTac. One possible opportunity to do so is when both agents are bidding for the same hotel at the same night, but ATTac is bidding for less rooms than the other agent.

Similarly, if an agent is scoring better than ATTac, it should stop cooperating with the other agent in the entertainment auctions. However, it is not clear to me if the agents in TAC have enough information to make such decisions.

References

[ARA2003] R. Arunachalam, N. Sadeh. *The 2003 Supply Chain Management Trading Agent Competition.* AAMAS-04 Workshop on Trading Agent Design and Analysis, 2004.

[FDL2001] C. Fritschi, K. Dorer (Living Systems AG). *Livingagents strategy description.* http://tac.eecs.umich.edu/research_reports/livingagents_strategy.pdf, 2001.

[GRE2002] A. Greenwald. *The 2002 Trading Agent Competition: An Overview of Agent Strategies.* AI Magazine 24, 2003.

[RPH1998] M. Rothkopf, A. Pekec, R. Harstad. *Computationally Manageable Combinational Auctions.* Management Science 44, 1998.

[SLS2001] P. Stone, M. Littman, S. Singh, M. Kearns. *ATTac-2000: An adaptive autonomous bidding agent.* Journal of Artificial Intelligence Research, June 2001.

[STO2001] P. Stone, A. Greenwald. *The First International Trading Agent Competition: Autonomous Bidding Agents.* Journal of Electronic Commerce Research, 2001.

[SSC2002] P. Stone, R. Schapire, J. Csirik, M.Mittman, D. McAllester. *ATTac-2001: A learning, autonomous bidding agent.* Revised Papers from the Workshop on Agent Mediated Electronic Commerce IV, July 2002.

[SSM2002] R. Schapire, P. Stone, D. McAllester, M. Littman, J. Csirik. *Modeling auction price uncertainty using boosting-based conditional density estimation.* Proceedings of the Nineteenth International Conference on Machine Learning, 2002.

[WGS2003] M. Wellman, A. Greenwald, P. Stone, P. Wurman. *The 2001 Trading Agent Competition.* Electronic Markets 13, 2003.