Dec14, 2002 TR CS-02-15

UMBCTAC: A Balanced Bidding Agent

Li Ding[1,2], Yongmei Shi[1], Zhongli Ding[1], RongPan[1] , Tim Finin

Computer Science and Electrical Engineering Department
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, Maryland 21250

[1] Email: dingli1, yshi1, zding1, panrong1, finin@csee.umbc.edu
[2] Supported in part by DAPAR contract F30602-00-2-0591

UMBCTAC: A Balanced bidding Agent[3,4]

Li Ding, Yongmei Shi, Zhongli Ding, and Rong Pan, Tim Finin

ABSTRACT

UMBCTAC is one of the top ranking agents in the 3rd International Trading Agent Competition (TAC). A TAC game has multiple auctions running on different but interrelated resources simultaneously, and 8 trading agents will compete with each other for optimal result – making maximum profit.

The spirit of simplicity and balance is used as a guideline to solve the dynamical optimization problem in TAC game. The hotel/airline auctions and the entertainment ticket auctions are handled separately by applying early bird heuristic. A gain-risk model is used to select a good, safe resource allocation for the clients in hotel/airline auctions. A fast and simple probabilistic approach is used to handle entertainment ticket auctions.

Keywords: TAC, balance heuristic, early bird heuristic

1 Background

1.1 TAC and UMBCTAC overview

The TAC game simulates a travel market which has multiple related resources: airline tickets, entertainment tickets and hotel rooms. Resources are exchanged through different types of auctions. The auctions are running simultaneously. The players, the trading agents, try to build their clients travel packages and then make profits by doing so. The agents compete with each other and the one with maximum profit wins.

The TAC game was proposed by [Wellman99]. The first TAC game was held in 2000. Then the game rules were improved in TAC01 and TAC02. After TAC02, researchers agree to explore more challenges beyond current game rules. So the current TAC game is titled as TAC Classic and TAC03 will use a supply chain proposal [Raghu02].

The UMBCTAC attended all the three past TAC and ranked 4th in TAC00, 9th in TAC01, and 4th again in TAC02[5]. The UMBCTAC in TAC02 used a heuristic approach. The heuristics are learned from the game history statistics and the common sense.

---

[3] Supported by DAML

[4] Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. E-mail: dingli1, yshi1, zding1, panrong1, finin@csee.umbc.edu. Please send all correspondence to Li Ding.

[5] The UMBCTAC for TAC00 and TAC01 was developed by another UMBC student, Youyong Zou, yzou1@umbc.edu.

1.2 TAC Classic

We will first explain the game rules of TAC02 (or TAC Classic) before proceeding to next section. A TAC Classic game has goods, agents, and auctions. The goal for the trading agents is to trade goods in auctions, to assemble travel packages for their customers and to make maximum profit finally.

There are three types of goods: airline tickets, hotel rooms, and entertainment tickets. There are 2 types of airline ticket: fly-in which available on days 1-4, and fly-out which is available on days 2-5. There are 2 hotels in Tampa: TT (good hotel) and SS (cheap hotel). Each hotel has 16 rooms available on days 1-4. There are also tickets for 3 types of entertainment (agitator wrestling, amusement park and museum) on days 1-4.

A travel package from TACTown to Tampa should contain round-trip airline tickets and corresponding hotel room reservations. A client should spend at least one night in Tampa, and he/she can't change hotel during the trip. Bonus can be gained if the client's preferences are satisfied, for example, the client will give some extra bonus if he/she gets some entertainment tickets.

There are three types of roles: (1) client, who wants a round trip from TAC Town to Tampa. He/She has personal traveling preference; (2) trading agent, who purchases travel packages for its 8 clients, buys and sells goods in auctions, and therefore makes profit; (3) auction robot, who hosts auctions. Trading agents communicate with auction robot to query the auction quotes and to submit bids.

There are 28 auctions running in three different types of goods. (1) 8 airline auctions are continuous one-sided auctions. There are unlimited supply till the end of game, and the price tends to rise over time; (2) 8 hotel auctions are standard English ascending multi-unit auction, and the only difference is that they close at random time in game; (3) 12 Entertainment Ticket auctions are standard continuous double auctions (like the stock market) that close when the game ends.

2 Heuristics

UMBCTAC is designed under the spirit of simplicity, robustness, and safety. It is well known that maximum profit always accompany with high risk, and high risk always cause leads to large loss. So the UMBCTAC try to find a solution which is both good and safe.

2.1 The early bird Heuristic

2.1.1 Motivation

In TAC game, a trading agent needs to consider two problems: resource allocation and bidding action. The resource allocation is very important in that it determines the resource allocation to be bid and the future bidding actions fully rely on it. From the aspect of when to make decision, there are two strategies: fix resource allocation at the very beginning, or developing it during the game.

2.1.2 The early bird Heuristic

The early bird heuristic is simple: a trading agent decides its resource allocation at the very beginning of game and does not change thereafter. All allocated resources are definitely needed by the agent.

When use this heuristic, the trading agent needs to predict the some future prices so as to evaluate the goodness of travel packages. Therefore, a *perfect prediction* assumption is made, and it assumes that the prediction always successes.

2.1.3 Evaluation

The trading agent can benefit a lot from the early bird heuristic. After the agent fixed its resources allocation at the very beginning, its bidding actions concentrate on low price, and it doesn't have to worry about the cost of switch plan. While this heuristic simplifies the complexity of future bidding actions, it also reduces the flexibility. The perfect prediction assumption is not necessarily always true. And the agent might loss a lot only because it can't change its resource allocation in abnormal cases.

Another choice is the cautious bidder heuristic: an agent modifies its resource allocation and bidding actions according to the change of game state and owned resources. This heuristic is more adaptive and able to predict accurately. However, it has to pay the cost of delayed decision, such as the rise of airline ticket price (see Figure 1), missing good deals, and the cost for unused goods.

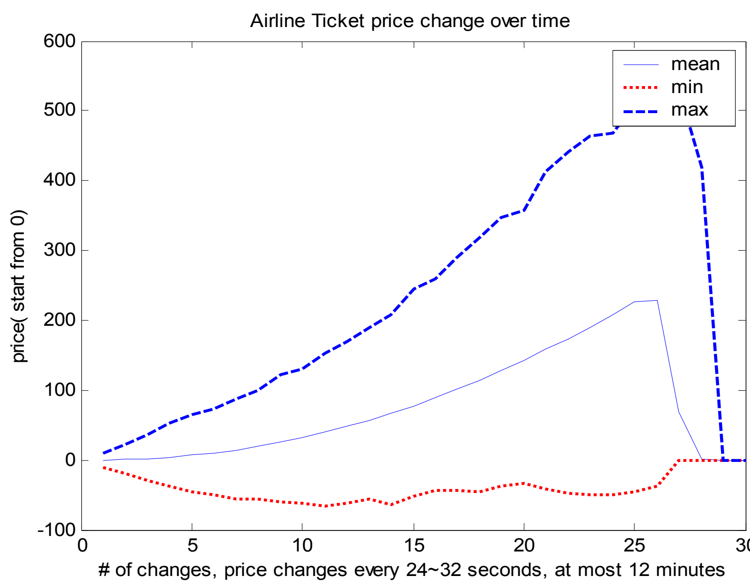


Figure 1 Change of airline ticket price over time [6]

2.1.4 Future work

If all trading agents use the early bird heuristic, then the game will be no more interesting than a study on bidding strategies for fixed demands. Since the early bird strategy fixes the resource allocation and all allocated resources are critical to form a

[6] Assume that the start price is $0, and the statistics is based on 10000 controlled experiments. It shows that the airline ticket price grows faster and faster over time, and the price variation is quite large. The drop at the end curve is caused by the different length of price change period: change price every 24~32 seconds.

travel package, the trading agent can either buy the resources at any price or leave one travel package fail. So we need some alternatives.

A good heuristic can take advantage from both early bird and cautious bidder heuristic. One approach bids more hotel rooms at the same time and allowing being overbidden. It changes its bid actions during the game according to its owned hotel rooms. It can buy more hotel rooms at low prices and provides the clients more flexibility. It has good performance when the demand is no more than supply, but it often boosts the average prices. Another approach delays part of its airline ticket purchases. So when it is overbidden in hotel auction, it can shift the corresponding travel plan to a shorter one. It has good performance when abnormal situation happens, i.e. too many clients want stay in Tampa at the same day, but it also introduces extra cost for delayed airline ticket purchase.

Both of the ideas try to determine most of wanted resources at the very beginning and leave the rest resources undecided. That will both add some flexibility and retain stability. The trick is how to choose the critical resources. Actually we can rank the resources with respect to flexibility. For example, we will bid for a long travel package, but leave the departure airline ticket open. So we can either change it to a shorter travel package when the hotel room cost too much, or keep it when the hotel room cost is cheap.

2.2 The Balance Heuristic

2.2.1 Motivation

The TAC game provides an interrelated and uncertain environment for the trading agents: the utility function imposes casual relations among the goods; the concurrently running trading agents affect each other; and the hotel auctions are closed in random order. How could a trading agent make a good resource allocation to achieve optimal performance upon such incomplete and uncertain context?

2.2.2 The balance heuristic

It is interesting to study the correlations among the three economic terms: Demand, Price, and Supply. In one-sided auction, the price always changes with the demand because the supply is fixed. For example, the price tends to increase fast when demand (number of bids) is higher than supply; and the demand tends to decrease when price is high. In double auction, the three factors affect each other. For example, buy price will be high when demand is larger than supply; then the supply will increase when buy price is high; then the demand is be less than supply; then the sell price drops; and then the demand might increase because of low sell price. These complex casual relations dominate the change of market state.

In order to satisfy the perfect prediction assumption, an agent needs to act to keep the balance between the demand and the supply in TAC game. This is the essence of the balance heuristic. As for the agent, it will keep its own resource allocation moderate.

2.2.3 Evaluation

There are two major benefits from the balance heuristic: (1) when the supply and the demand are balanced, the TAC game always results in normal state, in which the historical average price is good for predicting the future price; (2) the moderate allocation of resources also reduces the risk of losing a lot in abnormal state.

The average price is simple and robust. Suppose the value of demand is chosen from the set of {low, high}. Since there are 16 interrelated hotel auctions, we will have 2^{16} possible demand patterns. In stead of the exponential number of records for the full spectrum of distribution, the average only cost one record. Moreover, the statistical distribution is learned from noisy data: the data is collected from different combination of agent participants; the same agent may update its internal policies; and the initial client preferences are generated randomly too. All these noises make the distribution not so reliable. However, the average price is relatively robust to noise.

A moderate resource allocation means the agent only want no more than the average resources it can have. By doing so, this agent will not intentionally break the overall balance between demand and supply, and the system will be more easily to remain in normal state. Even when the balance is broken, the agent will not suffer much because its loss is less than average.

However, since the balance heuristic is simple and favors safety rather than profit, it might not be the best trading agent but an above average agent. And it needs large amount of games to obtain its statistical advantage. That's why its rank is 2nd in the qualification round (120 games), 3rd in the seeding round (440 games)[7], and 4th in the finals (32 games).

2.3 The Separation Heuristic

2.3.1 Motivation

In TAC game, there are three types of goods, and they together affect the final profit. However, it will be computational expensive or even impossible to consider them at the same time. Fortunately, if we only consider the combination of the hotel rooms and airline tickets, each client only has 20 legal travel plans[8]. But if we add the entertainment ticket to our consideration, the uncertain nature of double action will make the problem far more complex and undecided.

2.3.2 The separation heuristic

The separation heuristic is used to further simplify the resource allocation process: handle loosely connected auctions separately. The UMBCTAC firstly separate the hotel/airline auction (use early bird heuristic) and the entertainment auction (use cautious bidder heuristic). So the hotel/airline resource allocation finishes at the very beginning, and its result can be used in the dynamic resource allocation of the following entertainment auctions. Moreover, the UMBCTAC also handle the entertainment auctions individually for simplicity and speed.

---

[7] 6 of the 440 games have very bad result because of network failure.

[8] The duration which a client can stay in Tampa only has 10 legal choices: day1, day2, day3, day4, day12, day23, day34, day123, day234, and day1234 (day1 mean Monday, day2 means Tuesday, and so on). Since the clients can't change hotel while in Tampa, they have two choices on hotel type, good hotel or cheap hotel. Therefore, a client can have 20 possible legal travel plans. Note that not go to Tampa is a special legal travel plan which is not included in the 20 plans.

The separation heuristic comes from three observations. (1) None purchase dilemma. If each agent desires to make more profit than the other agent in entertainment auction, no purchase will happen. That is because the buyer will spend at most half of its entertainment bonus so that it can make more profit than the seller; ironically, the seller knows this too, and they will sell their goods only when they can make more profit than the buyer. Therefore, no purchase can get through! In real TAC games, however, not all agents are so rigid. (2) Big difference on the search space size. While there are only 20 possible choices for hotel/airline auctions per client, the choices increases rapidly when we consider the entertainment auctions. For example, suppose a client will spend 3 nights in Tampa, then he/she can have 5*4*3 possible choices. Moreover, since different client offers different bonus, we still need to consider how to assign the entertainment tickets to the 8 clients. (3) No optimal solution. Even we have an efficient algorithm that can find the "optimal" solution in entertainment auctions, there is no guarantee that the solution is really optimal. When the travel plan of individual client changes during the game, we need to recalculate entertainment ticket allocation for all clients. Moreover, the nature of double auction makes the supply and price of entertainment ticket unpredictable. So we can't say that an entertainment ticket allocation is the best during the game. (4) Fast response is preferred. When a good deal appears in double auction, only the fastest response can get the deal. In that case, a slow but perfect decision doesn't work.

2.3.3 Evaluation

The separation heuristic is indeed a divide-and-conquer method. Its significant benefits include: the search space for hotel/airline auction become very small and fits for exhaustive search; the search space for the entertainment ticket auction is also reduced because the client travel plans are fixed; and the decision delay in entertainment auction is greatly decreased. The disadvantages include non-optimal solution, rigid resource allocation, and possible stupid decision in double auction.

Linear programming (LP) is a popular approach in TAC because it can solve optimization problem efficiently. The LP method can find the "optimal" solution that conform the given constraints. However, not all LP approaches work well in TAC game, and there are still some problems. First of all, it is hard to define the constraints in LP due to the inherent incomplete domain knowledge of human designer. Secondly, the inputs of LP solver include estimations of price, and that will make the output uncertain too. Finally, LP algorithm is not designed for dynamical decision, and it is questionable to run a greedy algorithm where LP is used for each decision step.

3 Design issues

3.1 Estimate profit

According to the TAC game rules, the agent performance is evaluated by the score, i.e. the utility (see table below) minus the cost (expenditure in purchase). The best resource allocation yields the highest score...

Table 1 the Utility function (Adapted from TAC description)[9]

utility = 1000 - travel_penalty + hotel_bonus + fun_bonus

where

travel_penalty = 100*(|AA - PA| + |AD - PD|)

hotel_bonus = TT? * HP

fun_bonus = AW? * AW + AP? * AP + MU? * MU

cost = hotel_room_cost + airline_ticket_cost + fun_cost

score = utility – cost

As for the utility, once the client preferences are given, PA, PD, HP, AW, AP, and MU are fixed. The travel_penalty and hotel_conus will be fixed if we know AA, AD, and 'TT?'. However, no agent can guarantee its client can always get the entertainment tickets they want. So it is necessary to estimate the fun_bonus.

The cost of the travel package includes the airline ticket cost, hotel room cost, and the fun_cost (the money we spend for entertainment ticket). It is always desired to estimate the final price before we decide resource allocation. Note we should also count the expense on the unused but owned goods[10]. At the very beginning of game, we consider all the 20 legal travel packages (see also 8) in form of (arrival date, departure date, hotel). For each travel package, we know their travel_penalty, hotel_bonus, and the cost of airline ticket. So we only need to estimate the hotel cost and the fun_bonus.

3.1.1 Estimate the fun_bouns

Each trading agent is assigned some entertainment ticket initially, but no one knows the type and amount of entertainment tickets they will get from the auctions. In previous section, we know that no one can predict the "optimal" solution during the game because of the nature of double auction. However, fun_bouns can be partially estimated. UMBCTAC estimate fun_bouns for each client. The bonus is fully counted if the client offers high enough buy price plus the trading agent has the corresponding ticket in hand; The bonus is partially counted if the buy price is high enough, that is because the agent

[9] AA means actual arrival date. PA means preferred arrival date. AD means actual departure date. PD means preferred departure date. "TT?" is 1 when the good hotel is booked, and it is 0 otherwise. HP is client credit on living in good hotel. "AW?", "AP?" and "MU?" is 1 when corresponding entertainment ticket is obtained. Here AW refers to client credit on Alligator Wrestling; AP means client credit on amusement park; and MU client credit on means museum.

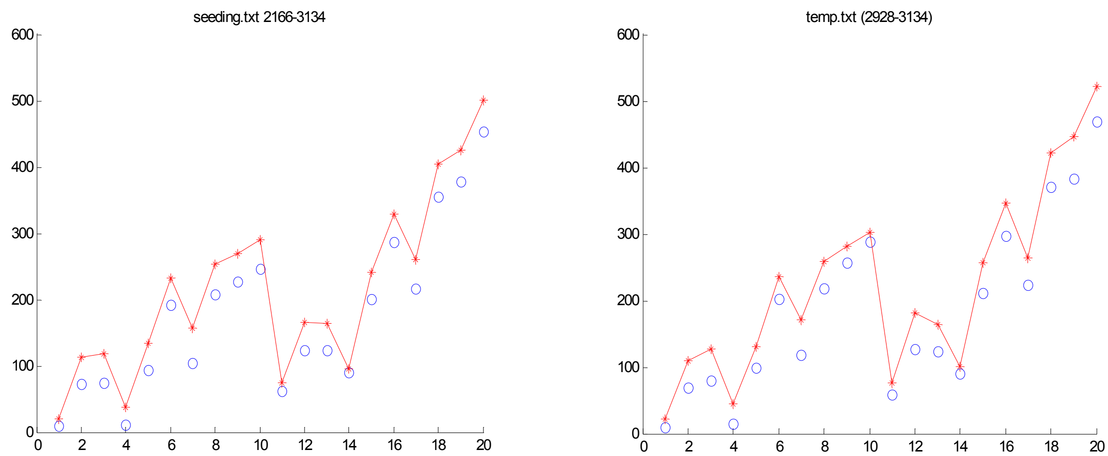
[10] Note there is possible another special cost, the fine for oversell the entertainment ticket.

will have a high probability to buy the ticket from auction; The bonus will be zero if the buy price is too low.

3.1.2 Estimate hotel cost

It is both important and hard to accurately estimate the price of hotel room. Game history serves as a good source for prediction. And there are quite a few choices: (1) we can randomly draw a price from all history prices on one auction. But the ticket price will seldom be the same as the future price; (2) median or mean from historical price can be used as an estimation of hotel price.

Figure 2 shows 1000 and 100 games statistics on the hotel cost for the 20 possible travel packages for a client[11]. The two sub-figures show that: (1) shorter hotel combinations cost less; (2) cheaper hotel cost less; (3) the distribution of hotel combination price is very similar for 100 games and 1000 games; (4) the median is a little bit optimistic while the mean is a little bit pessimistic.



(a) Hotel price based on 1000 game (b) Hotel price based on 100 game

Figure 2 the average hotel room price curve

Note that not all history data can be used for prediction. The participants of game do affect the game result. The antique history data does not work because the participant agents and the internal design of agents are different. So UMBCTAC only use 100 recent games' history data as learning resource.

3.2 *Hotel/Airline bidding policy*

Hotel and Airline auctions are closely interrelated, so the UMBCTAC consider them together with the guide of the early bidder heuristic and the balance heuristic. A Gain-Risk model is used to select a good combination of travel packages for all clients, i.e. a combination with low risk and good gain. Here, gain can be evaluated by the sum of estimated score of each travel package. Risk is the probability of we spend a lot in hotel

[11] All the data are collected from 2002 seeding round. X axis corresponds to 20 travel packages and Y axis corresponds to the statistical price. Mean is denoted by red solid line, and median is denoted by blue circles.

auction. The Gain-Risk model consists of three important components: how to estimate gain, how to estimate risk and how to search a good solution with both good gain and low risk.

3.2.1 Estimate gain

For a client, each of its 20 possible travel packages can be represented in form of a triple (arrival date, departure date, hotel). How to estimate the profit of individual plan is discussed in previous section.

3.2.2 Estimate risk

According to the balance heuristic, we consider the risk for each candidate combination of travel packages, which corresponds to a certain resource allocation. We simply sum up the difference between the resource allocation and the empirical threshold to get the risk. The empirical threshold is generated from following observations: (1) since there are 16 rooms in each hotel auction and there are 8 agents, an agent can have 2 rooms in an auction in average. More room allocation in an auction will cause higher risk; (2) the penalty of high risk is much higher than the profit. i.e. we shouldn't attempt high risk; (3) high risk is caused either by a trip with long duration or too much room allocation in a hotel auction. (4) Day 2, 3 have higher risk than day 1, 4; (5) in most time, a travel plan which matches or between the preferred arrival day and departure day has lower risk than others.(6) There are not much demands on hotel room in Day1,4.

So we use thresholds and weights to quantify the risk.

Risk = Sum (# of rooms exceeds threshold * weight)

For each hotel auction, we have a threshold for the maximum number of rooms we can allocate in that auction. Only the room allocation exceeds the threshold with cause risk. Moreover, a weight is assigned to the auction with respect to possibility of running into risk. The weight is higher for day 2, 3 than for day 1,4. Finally, we sum up the risks for each hotel auction to get the overall risk.

3.2.3 Search best gain-risk

We should also consider the travel plan selection. For each client, we know the some travel plans are good and safe, while the others not. So given the client preference, UMBCTAC only need to consider some of the 20 travels. Based on the above analysis, an algorithm is used to choose the best travel package combination.

Gain-risk-algorithm

1. Clients select favored travel package (FTP) which are considered sub-optimal.
2. Clients estimate profit for each FTP,
3. Clients submit their FTPs to Dealer in form of (FTP-id, estimated profit)
4. Dealer exhaustively searches in possible travel package combinations to find the one with lowest risk. If there are more than one candidates which have the same risk, the one with largest gain is preferred
5. Dealer submits bids aggressively to the TAC server

3.2.4 Evaluation and future work

The gain-risk model always outputs travel plan combination which needs two rooms in each hotel auction, especially in auctions on day2, 3. That is, every time we need fixed number of rooms. Can we change the search policy to "find the plan combination with the best gain while use two rooms in each auction"?

The balance heuristic does help the game to remain in normal state, but it can't guarantee. In abnormal situation, the prediction from normal case will be useless. Moreover, when we do not have any history data in current game setting, there is no way to predict. In order to avoid the mistake from estimation, let us consider the margin, which is the maximum profit we can achieve without consider the estimated cost, i.e. the score without the estimated hotel cost. Can we use the margin as gain?

The risk evaluation method, which simply sums up the individual risk, is not theoretically sound. We'd better estimate the risk for each auction, then compute the risk for each client by multiply the values, and then sum them up.

3.3 Entertainment bidding policy

Following the separation heuristic, the entertainment auction is handled individually. After we have settled down the optimal travel plan combination with algorithms in 3.2, we can dynamically bid in entertainment ticket auctions to achieve optimal e-ticket.

3.3.1 How bid change over time

A bid includes two parts: sell price—how much you want to ask for an e-ticket and buy price—how much you want to spend for an e-ticket.

There are several possible situations in the e-ticket (entertainment ticket) market: (1) no one wants tickets, then the sell price will decrease; (2) no one sells tickets, then the buy price will increase; (3) someone wants to sell and someone wants to buy but their prices do not match yet. In the third case, the price change can be modeled as several rounds, such that each round starts from large difference and ends up at a match (see Figure 3). The third case is very common in real life. To buy a ticket in a low price, we need to determine when to buy. Our approach use following rules:

- Use a function which can change over time.
- Use a random factor to increase the probability of achieving a match
- Use thresholds to avoid pay too much or sell in a too low price

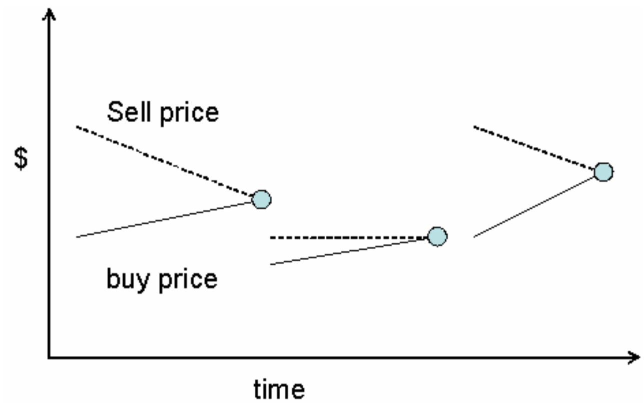


Figure 3 Price change in e-ticket auction

3.3.2 Determine bid action and bid price

A buy bid can be either higher than current sell price, which means we want to buy the e-ticket now, or lower, which means we only want to spend that amount to buy an e-ticket and we will wait for someone willing to sell at that price.

Since we already decided the travel package for each client, the rest is to assign e-tickets to them. This is not a static allocation problem because we can both buy and sell e-ticket, plus the supply and price of e-ticket keep on changing all the time. The UMBCTAC uses a probabilistic based approach: (1) no e-ticket needs to be assigned to a specific client, instead all clients get their e-tickets with some probabilities; (2) for the agent, the client request and the buy-bid from auction are treated as the same, so sometime the agent will sell the ticket instead of giving it to its client; (3) to avoid the non purchase dilemma, we set up a price range for both buy and sell actions; (4) the buy and sell actions are executed with certain probability.

For a certain agent, the lowest price its clients can offer in an auction which corresponds to (type of ticket, day) is determine by the number of clients who stay in Tampa on the same day, how long they will stay in Tampa, and their offered bonus.

The sell price range consists of a high price and a low price. If current buy price in auction is higher than low price, we can sell the ticket; if current buy price in auction is lower than low price, we do nothing; else we will put the high price in the auction as a sell bid and see if anyone can accept it. The buy price works the same as the sell price.

The price range (low, high) is used to achieve better performance. For example, we can delay a sale to get better price because some time the buyer might offer better price one minute later. We also can’t wait too long because someone else might sell the e-ticket before us.

The probabilistic buy and sell actions are used to simulate human decision. For example, people always don’t need to have more than 4 tickets. So UMBCTAC model the tendency of buy and sell in e-ticket auction as following: the tendency of buying an e-ticket is related to the number of tickets we currently hold. When we have less than 2 e-tickets, we might want to purchase some more tickets, or when we have more than 2 e-tickets, we might want to sell some tickets.

3.3.3 The UMBCTAC bidding strategy in e-ticket auction

The bidding algorithm is running individually in each e-ticket auction. Let k be the number of e-tickets the UMBCTAC already owns. We use a probability function p(k) to implement the probabilistic buy or sell.

$$P(k) = 0.9^{3^k}$$

Ticket owned	0	1	2	3
Probability P(k)	0.900	0.729	0.387	0.058

The bidding algorithm is defined as below:

Handle-e-ticket-auction (t)

w is the highest price which can be offer by the 8 clients for that auction.[12]

t is the percentage of time which has passed, it range from 0 to 1.

1. # initialize
2. compute k, P(k) based on current allocation
3. compute w according to the need of clients
4. # buy
5. compute (low-buy, high-buy) price based on P(k), t and w
6. with probability P(k), we send a buy bid – if current ask price in auction falls between our acceptable range, we buy it instantly, or we post the low price in the auction
7. # sell
8. compute (low-buy, high-buy) price based on P(k), t and w
9. with probability P(k), if current bid price in auction falls between our acceptable range, we sell the e-ticket instantly, otherwise we post a sell bid with high price

3.3.4 Evaluation and future work

This probabilistic bidding strategy works fairly well in TAC02 and we kept on improving it till the end the seeding round. Our approach is quite different from the other teams. It greatly simplifies the decision process.

We believe its success comes from the simulation of human decision. Future work will be done on building its theoretical foundations.

4 Conclusion

The UMBCTAC use simple heuristics to achieve average behavior. It has good statistical average performance, but it does not always achieve the best result in games. We still not fully understand why and how the heuristics work. We also need the theoretical bound for them. Future work can focus on theoretical explanation and efficient algorithms for dynamically searching a solution in uncertain context.

5 ACKNOWLEDGMENTS

Our thanks to Dr. Finin, Dr. Peng, and Dr. Oates for their helps and supports.

[12] The buy price should always less than w, while the sell price should always larger than w

6 Reference

[Wellman99] Michael P. Wellman and Peter R. Wurman. A trading agent competition for the research community. *IJCAI-99 Workshop on Agent-Mediated Electronic Commerce*, Stockholm, 1999

[Raghu02] Raghu Arunachalam, ect. The TAC Supply Chain Management Game (Draft version 0.5), 2002. http://www.sics.se/tac/