Thalis: A Flexible Trading Agent

Maria Fasli
University of Essex
Department of Computer Science
Wivenhoe Park, Colchester CO4 3SQ, UK
mfasli@essex.ac.uk

Nikolaos Poursanidis
University of Essex
Department of Computer Science
Wivenhoe Park, Colchester CO4 3SQ, UK
npours@essex.ac.uk

ABSTRACT

With the advent of the Internet, trading in electronic market places has become common practice for an increasing number of businesses and individuals. The Trading Agent Competition (TAC) is an open-invitation forum designed to encourage research into electronic markets and trading agents. The emphasis of TAC-02 and previous competitions has been on developing a successful strategy for maximizing profit in a constrained environment. In this paper we discuss the challenges of TAC which features a complex benchmark e-market problem and we present the strategy of the trading agent Thalis that participated in the competition successfully obtaining the third place among the finalists.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

Keywords

trading agents, auctions, e-commerce

1. INTRODUCTION

Agent-based and multi-agent systems have become increasingly popular as a means of conceptualising and implementing a wide range of applications. More recently, with the advent of the Internet trading in electronic market places has become common practice for an increasing number of businesses and individuals and as a result there has been a mounting interest regarding the utilisation of agents in such applications. Agent technology, being particularly suited for information rich environments, can be applied in most of the stages involved in searching and buying products and services [7]. For instance, semi-autonomous and personalised software agents can be used in the negotiation stage in order to automate the trading process. This has several potential benefits such as reduced costs and greater efficiency as well as savings in time and effort. One of the most efficient ways of negotiating for goods and services is via auctions. Although constructing an agent to take part in an auction for a single good is relatively simple, developing an agent to participate in simultaneous auctions offering complementary and substitutable goods is a complex task. This is the form of the problem tackled by the Trading Agent Competition (TAC) which features artificial trading agents competing against each other in a market-based scenario [3, 4]. In this paper we discuss the challenges of TAC which features a complex benchmark e-market problem and we present the strategy of the trading agent Thalis that participated successfully in the competition achieving the third place among the eight finalist agents. The structure of the paper is as follows. In the next section an overview of the game is provided. Next we discuss the challenges that the competition presents. The rest of the paper is devoted in the analysis of the bidding strategy of Thalis. The paper ends with the conclusions.

2. GAME DESCRIPTION

The Trading Agent Competition is an open-invitation forum designed to encourage research into electronic markets and trading agents. The emphasis of TAC-02 and previous competitions has been on developing a successful strategy for maximizing profit in a constrained environment [3, 5]. TAC-02 was organised by the Swedish Institute for Computer Science (SICS). Taking over from the University of Michigan [9], developers from SICS had to build a new TAC server for running the competition in accordance with the TAC game specification. 19 out of the 26 teams originally expressing interest participated in the event . TAC-02 was organised into two stages: the first stage included the qualifying and seeding round games and the second stage the semi-final and final games as well as the workshop that was held in Edmonton Canada during the AAAI-02 Conference.

TAC trading agents operate within a travel agent scenario buying and selling commodities in order to satisfy their clients' preferences. All commodities are traded simultaneously in electronic auctions which are operated by the TAC server. A TAC game lasts 12 minutes and involves eight agents competing against each other. The agent's objective is to create travel packages from TACtown to Tampa during a notional 5-day period for eight clients. A feasible travel package for a client requires an in-flight reservation as well as an out-flight one and a hotel reservation for the duration of the trip in the same hotel. In addition, an entertainment package may be assembled comprising of tickets to one or more of three available entertainment events: alligator wrestling, amusement park and museum. Each of the agent's clients has its own individual preferences over the various aspects of the trip which are provided to the agent in the beginning of every game. The preferences comprise:

- Arrival and departure days (i.e. arrival on day 1 and departure on day 3)

• Good Hotel Premium (GHP): A measure of the preference for a good hotel over a bad hotel (i.e. an additional cost the client is willing to pay for good hotel)

• Entertainment Ticket Premiums (ETP): A measure of the preference for every type of entertainment (i.e. an additional cost the client is willing to pay for each type of entertainment)

Moreover, each travel agent receives an endowment of 12 entertainment tickets which are divided as follows:

• Four tickets of a particular type on day 1 or day 4.

• Four tickets of a particular type on day 2 or day 3.

• Two tickets of a particular type (different from above) on day 1 or day 4.

• Two tickets of a particular type (different from above) on day 2 or day 3.

There are two types of airline tickets, flights to Tampa (in-flights) and flights from Tampa (out-flights). There are four auctions for the in-flight tickets (days 1-4) and four for the out-flight tickets (days 2-5). Flight auctions are single seller, continuous auctions which open as soon as the game begins and close at the end of the game. They clear continuously and prices are set according to a stochastic function with the opening price being between 250 and 400 cost units. After the first clearing the auction price changes every 24 to 32 seconds and is kept within the range of 150 to 800 cost units. The supply of available seats on these flights is unlimited. Agents may submit buy bids, but not sell bids. Price quotes are issued immediately in response to new bids specifying the ask price, which is simply the price of the current sell bid. Any buy bids that are matched (i.e. their prices are equal or higher than the current ask price) are transacted immediately at the ask price while all others remain as standing bids until matched or withdrawn.

There are two hotels available: the "good hotel" Tampa Towers (TT) and the "not so good" Shoreline Shanties (SS). Sixteen (16) rooms of each category of hotel are available for days 1-4. Hotel auctions are English type, ascending multi-unit auctions which start at the beginning of the game. Every hotel auction clears and matches active bids only once: at its closing time. Four minutes after the start of the game one random hotel auction clears and closes. One random hotel auction closes every minute thereafter. Price quotes are generated once per minute so as to reveal the current ask and bid prices and the amount of hotel nights won so far by every agent. When an auction clears the 16 highest bidders win the rooms at the 16-th highest price. Bid withdrawal and reservation resale are not allowed.

Entertainment tickets are traded in continuous double auctions that start when the game begins and close when the game ends. Tickets are available from days 1 to 4, and they are traded in twelve auctions in total. Agents can act as both buyers and sellers in these auctions and the ask/bid prices are based only on their bid levels.

The satisfaction of each client is measured by means of a utility function, whose value expresses the proximity of the travel package assembled, against the preferences. The top scoring agent is the one that achieves the highest sum of the individual client utilities while minimizing the expenses for the goods bought. The client utility, measured in dollars, for a feasible travel package is computed by the formula:

$$CU = 1000 - TravelPenalty + HotelBonus + FunBonus \quad (1)$$

where

$$TravelPenalty = 100 * (|AA - PA| + |AD - PD|)^1 \quad (2)$$

$HotelBonus = GHP * QH$(QH=1 iff hotel=TT, 0 otherwise) (3)

$$FunBonus = EV1 + EV2 + EV3 \quad (4)$$

If the travel package is not feasible the client is assigned zero utility. At the end of a game the TAC server attempts to construct an optimal or a near-optimal allocation of the goods which have been purchased by the agent so as to create feasible travel packages while maximizing the utility of the agent's clients. The final score achieved by every agent is calculated as:

$$Score = Sum(CU) - (Cost) - (OverSellPenalty) \quad (5)$$

$$Cost = Sum(Purchases) - Sum(EntertTicketSales) \quad (6)$$

$$OverSellPenalty = Count(TicketsOverSold) * 200 \quad (7)$$

$TicketsOversold$ represents the number of entertainment tickets that the agent sold exceeding the available quantity that it originally had or bought. A more detailed description of the game can be found in [1].

3. CHALLENGES

There are several challenging issues in designing and implementing a successful bidding strategy for the TAC market scenario. Given the current ticket holdings (if any), the current market prices (or maybe the estimated prices) and the client preferences the agent is required to find the most profitable combination of tickets that need to be traded in order to complete travel packages. The *completion problem* [6] is a hard one since the number of the available tickets is constrained by the availability in the market. An additional problem arises since accurate price estimation is very difficult, if not impossible, because of the limited information the game provides, that is the current asking prices. The agent knows nothing about the other agents' bids or ticket demand while the game is running. Standard machine learning techniques are very hard to apply since the agent cannot observe the bidding patterns of the other participants.

Having already decided what tickets and how many are needed, the problem is to decide when and how much to bid. The agent may also consider bidding for tickets not currently included in the optimum package list to maintain flexibility later in the game.

The key resources are the hotel reservations since they are vital for the construction of feasible travel packages. Their limited availability makes them the most sought-after goods. Failing to acquire a particular hotel reservation may jeopardize the whole package. Things get even worse when the rest of the tickets have already been secured. Hotel auctions are ascending-price ones, and consequently once the price goes up there is no way back. An obvious strategy would be for the agent to place its bids at a high price, say $1000. The agent will most probably win the rooms at a price equal to or lower than the bid price and would not have to monitor the hotel auctions to update its bids. If the bid is rejected then the ask price would be much higher and therefore the reservation would not be profitable for the agent. Such a strategy has the drawback that the initial decision cannot be changed since bid withdrawal is not allowed.

The limited availability of hotel rooms creates a further dilemma: since flight prices will most likely increase with time, the expenses will be minimum if all wanted tickets are

[1] AA and AD are the actual arrival and departure days respectively.

purchased at the beginning of the game. This means committing early on packages, and thus having less flexibility to move a client's trip later in the game. The agent may fail to reserve some of the hotel rooms needed to assemble the travel packages, and additional flight tickets may have to be purchased to allow for changes. This entails that some of the initially purchased flights will remain unused while the expenses will increase. On the other hand, flights for each client could be purchased when all the necessary hotel rooms have been reserved, but the flight prices may be significantly higher, again leading to increased expenditure.

Although the entertainment tickets are not as important as the other resources, the additional bonus that can be obtained can distinguish a good agent from a not so good one. These are the only goods that can be both bought and sold. Sometimes it is more profitable to sell a ticket than to assign it to a customer, a fact that adds to the complexity of the bidding strategy. Last but not least, the agent has to take into consideration network disruptions and delays and make sure that its bids arrive on time.

4. THALIS

The design of Thalis[2] was based on a multithread, modular model that ensures both independence and sufficient interoperation between the different modules. The implementation language was Java. More details on the design and implementation of Thalis can be found in [8].

4.1 Initial Allocation and Optimization

At the start of the game Thalis performs the initial allocation taking into consideration the flight auctions opening prices as well as an estimation of the closing prices for hotel auctions that are generated via the use of statistics from previous games. Based on these two sets of information the optimizer calculates the cost of all possible solutions and selects the optimum one for every client. In order to avoid solutions that might result into excessive demand on a specific day and thus in excessive cost, the system applies an after-optimization process that attempts to reduce the number of reservations requested on particular nights. This is called the night reduction scheme.

The system uses an optimization process in order to continuously evaluate the client preferences against the current situation in the game and designate the optimum solution. The optimizer takes into consideration a number of parameters in order to output the optimum solution for every client:

- Client preferences
- Auction data, including status, last ask price levels and estimated price increase
- Coefficients, regarding the utility calculation for the case of good hotel solution and the penalty applied when selecting a solution that deviates from the original client preferences.
- Statistical data weight factor

The system makes use of historical data in order to estimate the hotel auction closing prices. A weight factor is used depending on the flow of the games. This closing price estimation affects only the night reduction scheme (see below) and the initial bid levels, while the remaining price estimation is based on the estimated ask price increase. The number of past games considered is a parameter in the system. The already purchased goods including flights and hotel nights for every client are also taken into consideration.

[2]Thallis the Militian (643 B.C.-548 BC) Greek Mathematician.

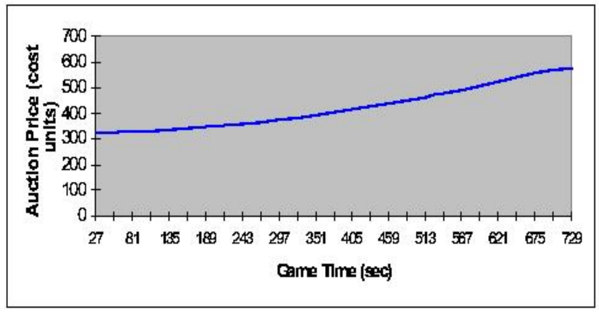

Figure 1: Flight Auction Prices Distribution.

The optimizer creates an array of all possible solutions for every client (irrelevant of its preferences). Possible solutions include every valid combination of arrival day, departure day, type of hotel and finally an array of all possible combinations of entertainment tickets for the duration of the trip. The utility value of any possible travel solution is calculated according to the following equation:

$$U = 1000 - (TravelPenalty) + (HotelBonus) * HotelBonusCoef \quad (8)$$

Where $HotelBonusCoef$ is the weight coefficient for the utility calculation in case of a good hotel solution. The optimizer calculates the cost of every possible solution based on the estimated ask price (last ask price increased by the estimated last ask price increase) of the relevant auction if the item has not been bought yet. After the purchase of the item, the relevant cost is set to 0. The system designates the optimum solution for every client as the one that maximizes the difference of its utility compared to its cost:

$$SolutionScore = U - StillToPayCost \quad (9)$$

Every entertainment package generated for each client has its own utility which is calculated based on the client preferences (Equation (4)). Once a solution for a client has been chosen, bid requests are issued to all the auctions involved in the solution. The system allocates purchased goods based on the requests received. Instead of keeping a chronological order (FCFS - first come first served), an ordering of the clients' pending requests by expected revenue is required in hotel and entertainment auctions. For entertainment auction requests, clients are arranged by decreasing fun bonus preferences so the client that has the highest bonus is allocated purchased tickets first. For hotel auctions, pending requests are sorted by decreasing good hotel bonus for good hotel auctions and increasing bonus for bad ones.

4.2 Flight Auctions

Although the server applies a random strategy in calculating the initial level of flight prices and their subsequent variation, as it can be clearly seen in Figure 1 prices tend to generally increase with time (Note that data plotted in this figure include all 1040 games of the seeding rounds). Although, a good strategy would involve bidding for flights as soon as possible after game initialization, this means committing early on packages and as a consequence having less flexibility to move a client's trip later in the game. In order to tackle this problem, a limit on the number of flights

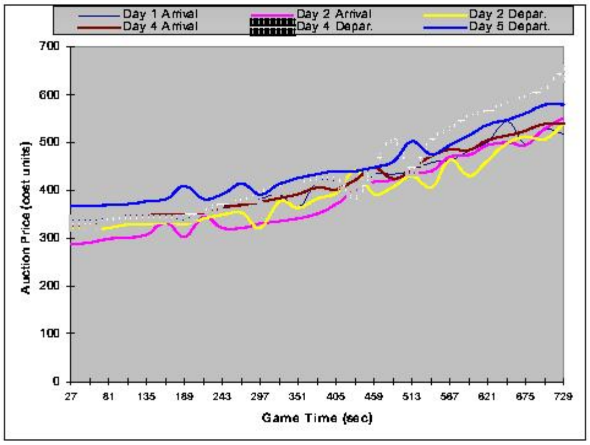

Figure 2: Flight Auctions Prices Distribution (Only 10 subsequent games considered)

purchased early on in the game was established. The distribution of flight auction prices was monitored on a small sample of 10 subsequent games. This was considered appropriate since in a large sample, small variations of the price may be hidden or balanced by the average function. A distribution based on 10 games is plotted in Figure 2. As it can be seen there, almost all flight auctions tend to maintain their initial price levels until game second 130. Prices remain relatively low until game second 300. After this the rates of increase are relatively high. Thalis' strategy in flight auctions can be summarised as follows:

• Flight requests generated after the night reduction scheme activation, are submitted to the auction manager. The exact time of bid submission is then dynamically decided upon satisfaction of at least one the following conditions:

- Auction price remains unchanged for more than four subsequent quotes
- Auction price has steadily positive or negative difference (compared to the opening price) for more than two subsequent quotes
- Auction price has increased by more than 5 cost units or has decreased by more than 10
- Game time exceeds minute five

• Flights that refer to client solutions that have either marginal revenue (i.e. below 90 cost units) or refer to four nights stay are not treated as above. Instead they are treated by a heuristic called DenyFlights. This heuristic allows the delay of the purchase of flights for a couple of clients per game. Thus, the relevant bid submission is prevented until one of the following conditions is fulfilled:

- Total increase in the prices of both flight auctions involved (arrival and departure) exceeds time and value limits stated in Table 1.
- Game time exceeds game minute five (5).

4.3 Hotel Auctions

Although the TAC game involves a great deal of uncertainty and unpredictability, the participants' bids tend to follow certain patterns and these patterns may stabilise over time. In Figure 3 average hotel prices have been plotted for all qualifying games and the first 100 seeding round games.

Game sec	Value Upper Threshold (cost units)
140	19
200	29
260	45

Table 1: Deny Flights heuristic time/value limits

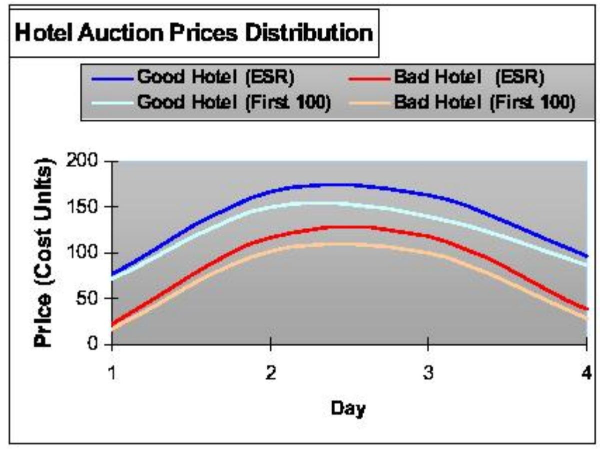

Figure 3: Hotel Auction Prices Distribution (ESR: End of Seeding Rounds, First 100: only the first 100 games of the seeding rounds are considered).

It is quite obvious that auctions tend to clear at an almost predetermined way that only varies slightly with the number of games. And in any case the relations between the prices of different auctions is kept constant. So for instance, it is clear that in both cases in Figure 3, day 1 good hotel is cheaper than day 2 good hotel. Even more specifically day 1 good hotel represents a price that is approximate to the 50% of the price of day 2 good hotel. The slight change in price levels that is observed in the flow of games reflects the changes in agents' strategies and in the client preferences.

The normal distribution of auction prices supports the use of statistics on clearing prices in order to obtain a rough estimation of future auction prices. However, instead of using all games in estimating the hotel prices, in practice we use only the ten (10) most recent games excluding the last game played. Thus, only the latest information available on the bid patterns of the other agents is taken into consideration. The last played game is excluded since i) if the prices are normally distributed, then it does not make any difference, and ii) if it involves excessively high prices this may mean that either some agent(s) changed its strategy or something unusual had happened. In either case excluding this last game should not make a huge difference to our agent's strategy. On the other hand if a change of strategy has taken place this will be taken into consideration as the games progress.

4.3.1 Night Reduction Scheme

As noted above demand for hotel reservations for the midweek days 2 and 3 is usually higher than that for days 1 and 4 (Figure 3). This leads to high auction prices for both hotels for those days. To avoid solutions that might result in excessive demand on a specific day and therefore in excessive cost, the system applies an after optimization process that attempts to reduce the number of hotel reservations requested on particular nights. This process works together with the optimizer (more or less like a guided local search algorithm) and:

- locates the days on which an excessive number of hotel reservations are requested (based on selected solutions). The decision is based on parameters stating the maximum preferred number of reservations per mid-week day (days 2 and 3) and per far-out day (days 1 and 4).
- applies a (parameter supplied) penalty so as to artificially increase the estimated closing price of the specific day's hotel auctions
- re-runs the optimizer so as to recalculate the cost of each possible solution for every client and select the optimum one
- applies the above process for as long as the conditions apply or more than 50 repetitions (threshold) on a specific day didn't result in a change. The termination limit (50 repetitions) is necessary because the night reduction algorithm, in contrast with the optimizer, may not terminate (just like a local search algorithm).

The night reduction scheme is activated in the form 5-3 (up to 5 reservations on day 1 and 4 and up to 3 reservations on days 2 and 3).

4.3.2 Bid Timing and Submission

All hotel auctions clear only once: when they close. At minute four one random hotel auction closes, and one random hotel auction closes every minute thereafter. By bidding before minute four the agent does not gain any significant advantage; on the contrary early bidding reveals the agent's intentions and the level of bids. Thus Thalis issues bids for the hotel auctions only after game minute four and these are issued every minute at a predetermined second (parameter $HotelBidTime$, default value 35). Although this means that Thalis misses the first auction that closes on minute 4, the information gained regarding the level of bids and consequently the other participants' intentions compensates for this.

4.3.3 DOSEF Heuristic

Since the sequence with which hotel auctions are closing is not predetermined, unavoidably there will be cases where bad hotel auctions will close first. In these cases bad hotel auctions usually close at a very low price while in the same game the same day good hotel auction usually closes at a much higher price. Due to these facts, all good hotel auction requests are replicated before the first bid to the bad hotel auctions. These replicated requests are to be used only when bad hotel auctions close first and at a low price. The whole idea of this heuristic is to keep all alternatives open (with the lowest risk possible) until after the first hotel auction closes (minute 4). After that the DOSEF (DOuble Shoot EFfort) heuristic is retracted in the sense that bids for each hotel auction are placed only for the auction's original requests.

4.4 Entertainment Auctions

Entertainment tickets are a very good way to increase the utility by buying tickets and reduce the cost of the agent's packages by selling owned tickets that are not very profitable in the clients' selected solutions. Nevertheless, special care has to be taken in order to submit sell and buy bids that will actually maximize the revenue of the agent without necessarily doing the same for the other participants. Both actions (buying and selling) are initiated after $EntsEarliestBid$ game second and terminate on the last second of the game (although, as we will see after the qualifying and seeding rounds this strategy was modified).

Agent	**Average nights reduction**	**Avg Score**
BigRed	7,08	572,94
TOMAhack	6,90	2501,12
Cuhk	5,38	2970,59
livingagents	5,08	3012,81
Thalis	4,64	2875,36
UMBCTAC	4,18	2897,73
Zepp	3,80	2050,42
kavayaH	3,43	2265,25
PackaTAC	3,39	2654,19
SouthamptonTAC	3,26	3006,98
Sics	2,53	2548,15
RoxyBot	2,50	2561,50
ATTac	2,21	3018,05
Walverine	1,30	2415,19
Tvad	1,14	2586,67
PainInNEC	1,05	2154,78
whitebear	0,43	2775,19
TniTac	0,07	2279,57
Harami	0,00	2035,73

Table 2: Average night reduction per agent

5. ANALYSIS AND EVALUATION

Thalis participated in the first stage of the competition achieving the 10th place among 26 agents in the qualifying rounds and 6th in the seeding rounds. More than 1000 games where played during the seeding rounds and every agent participated in 440 games. Since most of the agents participating had already stabilized their performance, it was deemed necessary to evaluate the strategy that had been used thus far as well as the results achieved in order to improve the performance of the agent. The logging module that had already been working with game logs in order to parse and store the final auction prices in the system database, was extended so as to be able to download and fully parse the game log files. The system database resulted in more 1,500,000 records and a size of 250 MBs.

5.1 The night reduction scheme

The main idea here was to evaluate the result of the available reduction schemes and compare them to select the most appropriate. A comparative query was constructed in order to get a picture of all the agents' relevant performance 2.

Although a direct relation between average number of nights and average score does not exist, still extreme values (very low or very high) should be avoided. Agents marked in italics in Table 2 seem to define two sets of ranges within which the average nights reduced should be. The possible reduction schemes are listed below:

- 5-3 schema: MaxNumEdgeDayNights =5 & MaxNumMidDayNights=3
- 5-4 schema: MaxNumEdgeDayNights =5 & MaxNumMid-

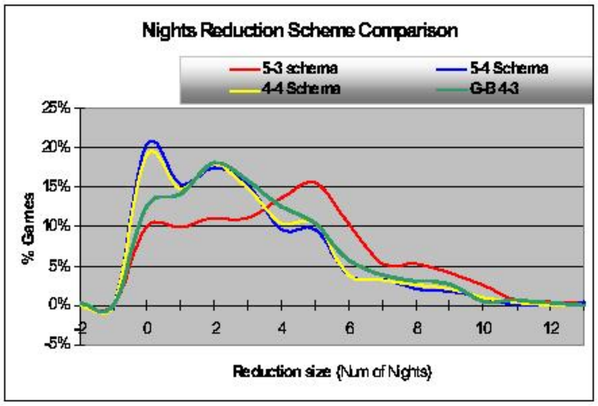

Figure 4: Nights Reduction Schemes Comparison

Reduction Scheme	Average Nights Reduction
5-3 schema	4.64
5-4 schema	2.72
4-4 schema	2.84
G-B 4-3 schema	3.22

Table 3: Average Nights Reduced per reduction schemes

DayNights=4
• 4-4 schema: MaxNumEdgeDayNights =4 & MaxNumMidDayNights=4
• G-B 4-3 schema (Good - bad hotel 4-3): Here the limitation was taken from the day level and applied to the auction level. So the maximum night restriction applies (for edge and mid-week days respectively), but not for the total number of night requests per day. Instead it applies for the number of night reservation requests per good and bad hotel auctions separately.

Using the same query, as above, and a simulation run for all games played by Thalis for every possible reduction scheme, the results can be seen in Figure 4. Note that the 5-3 scheme tends to achieve a higher degree of reduction in a large percentage of games (more than 15% of games had a 5 night reduction). On the other hand the 5-4 and 4-4 schemes tend to keep the majority of games at no reduction (0 nights reduced), while the G-B 4-3 scheme is somewhere in between. Based on these results the 4-4 scheme was selected to be used during the semi-final and final rounds. It provides a low average of night reduction and thus it will allow for higher scores (since the utility is going to remain high) especially if prices are kept at low levels. If the values of Table 2 are compared to the ones in Table 3, one can see that the selected scheme provided a value within the lower range of selected agents which by the way, includes the best three participants in the seeding rounds.

5.2 DenyFlights Heuristic

This heuristic allows the delay of the purchase of flights for a couple of clients per game. This is considered in cases in which the client solutions have marginal revenue (below 90 cost units) or involve a 4-night stay. Since it is not easy to evaluate how close to that target Thalis was during runtime, the simulation process was used again to reproduce all the games Thalis had played, under all possible reduction schemes and monitor the amount of clients that participated in the *DenyFlights* heuristic in every game. The results can be seen in Table 4 for all reduction schemes. It seems that whatever the reduction scheme the initial target was kept in average and so no changes were applied in this area.

Reduction Scheme	Average Nights Reduction
5-3 schema	0.98
5-4 schema	0.725
4-4 schema	0.778
G-B 4-3 schema	0.75

Table 4: DenyFlights average clients for the various reduction schemes.

Agent	Mean Num Of Flights
tvad	16,97
kavayaH	16,68
PackaTAC	16,51
SouthamptonTAC	16,29
whitebear	16,26
sics	16,24
Thalis	16,18
cuhk	16,14
Walverine	16,12
ATTac	16,06
PainInNEC	16,06
BigRed	16,06
tniTac	16,05
UMBCTAC	16,01
livingagents	16,00
harami	16,00
zepp	16,00
RoxyBot	15,82
TOMAhack	15,05

Table 5: Mean number of flights bought per game

5.3 Number of flights bought

The data collected from the seeding round games, showed that there are differences between agents regarding the number of flights bought per game. Still those differences are not so high as to require action. The results for all agents can be seen in descending order in Table 5.

5.4 Entertainment ticket transactions

One important factor in configuring the range of bids for both selling and buying entertainment tickets is that the premium for each such ticket varies between 0 and 199 cost units. Thus, if sales are to be profitable we should:
• Sell tickets at a price which is as high as possible, but not lower than the 50% of the premium.
• Buy tickets at the lowest possible price but never above the 50% of the premium.

As a consequence, when Thalis acts as a seller, the profit of the buyer agent should not be higher than the profit of Thalis, while when Thalis acts as buyer, the profit of the seller agent should not be higher than the profit of Thalis. Of course this is not always applicable in a real game since the current prices are not only configured by our agent, but the other agents as well. Still, it seems reasonable to incorporate

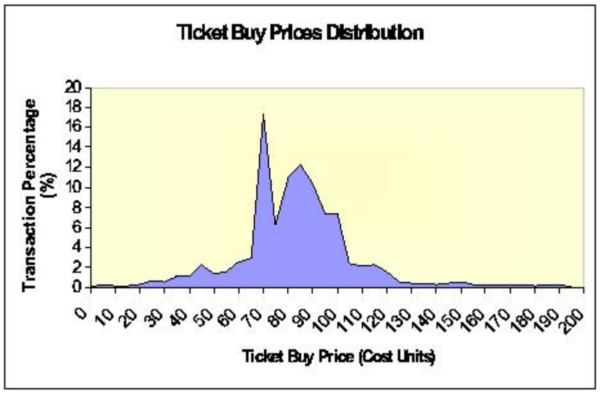

Figure 5: Ticket Buy Prices Distribution

Ent type	Average Buy Price	Total Trans.	Max Buy Price	Min Buy Price
1	80,792	12562	189,05	0
2	80,651	12410	189,05	0
3	82,098	12136	200	0

Table 6: Entertainment ticket prices during the seeding round

these tactics even at a lower level (with the limit being below the 50% of the premium). Average, min and max buy prices offered during the seeding round games are listed in Table 6. The average buy price is around 80 cost units for every kind of ticket which supports our suggestions above.

The distribution of ticket buy/sell prices was plotted in the graph of Figure 5. As a result the buy and sell price ranges were configured accordingly:

- Buy tickets range: 30-101. The upper limit was extended above 50% of the premium in order to increase the possibility that whenever needed, a purchase would take place even at a slightly higher price.
- Sell tickets range: 81-125. The lower limit was extended slightly below 50% of the premium in order to increase the possibility that sell bids would result in a transaction.

Figure 6 represents graphically the price areas for buy and sell tickets. Since agents are interested in selling tickets in order to decrease their cost, it seems quite normal for them to get "anxious" as time goes by and tickets are not sold. This "anxiety" is translated in the tendency to sell at lower prices. After analysing all the seeding round games, the results proved that agents do tend to sell at lower prices as time goes by. Specifically this tendency is increased after the 6th game minute as seen in Table 7. As a result, a modification was implemented in the strategy regarding entertainment tickets. More specifically while selling tickets is allowed after minute 1, bids for buying tickets are only issued after game minute 6 so as to minimize the cost of buying tickets and increase the overall score.

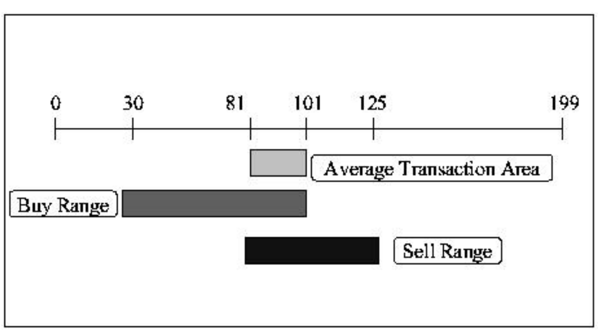

Figure 6: Entertainment Ticket Price Range

Time Range	Average Buy Price	Max Buy Price	Min Buy Price
<6th Min	90,81562	191,8193	17,1777
>=6th Min	73,3162	189,35	0

Table 7: Ticket Sell Price statistics before and after game minute 6

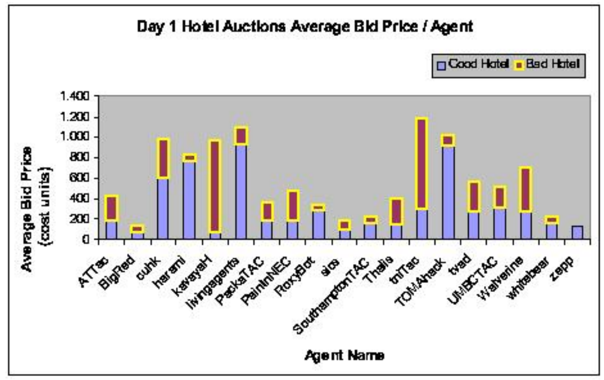

Figure 7: Day 1 hotel auctions

5.5 The high auction prices phenomenon

In many cases during the seeding rounds games, although a good result was produced by Thalis' strategy, the final score was quite low. This was as a consequence of auctions clearing at very high prices. The mean bid prices that agents offer for every different hotel auction per day have been plotted in Figures 7- 10. As can be observed, Thalis agent was

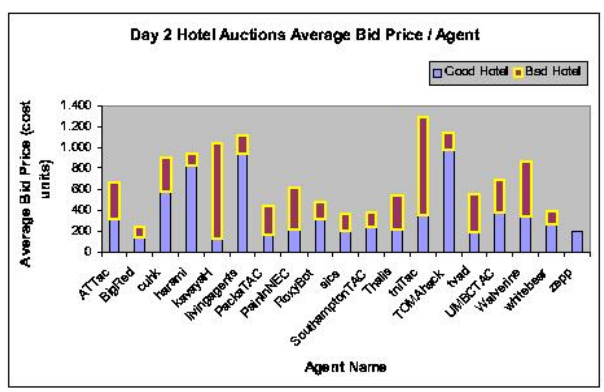

Figure 8: Day 2 hotel auctions

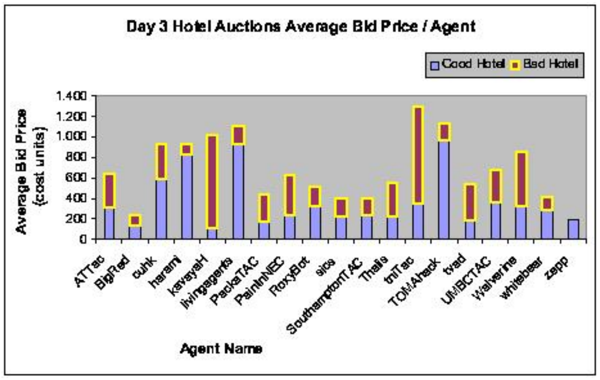

Figure 9: Day 3 hotel auctions

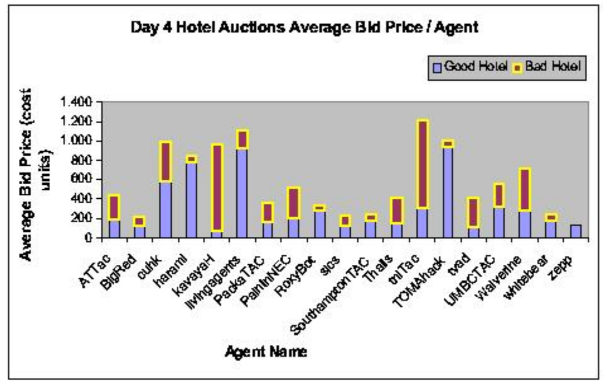

Figure 10: Day 4 hotel auctions

No	Agent	Avg Score
1	Whitebear (Cornell University)	3412.78
2	SouthamptonTAC (Southampton Univ.)	3385.46
3	*Thalis* (Essex University)	3246.27
4	UMBCTAC (UMBC)	3235.56
5	Walverine (University of Michigan)	3209.52
6	Livingagents (Livingagents AG)	3180.89
7	KavayaH (Oracle India Pvt Ltd)	3099.44
8	Cuhk (Chinese Univ. of Hong Kong)	3068.77

Table 8: TAC 2002 Final Results.

one of the few agents offering low prices in good hotel auctions. In bad hotel auctions our prices where a little bit higher compared to good hotel auctions, but still much lower than most of the other agents'. Thus the bad hotel price calculation was lowered so as to offer even lower prices in these auctions, especially in cases that the DOSEF heuristic was applied. More specifically the lowest bid limit for these auctions came down to 30 cost units from 45-50 that was before. After studying all games in which the high bid prices phenomenon occurred it was ascertained that in these cases most other participants were requesting only one hotel night and our agent was requesting one or more. After studying the other agents' bid strings in all relevant games the problem was revealed: the tactics applied now did not use High-Low bids[3] (except when applying DOSEF) while some of the others (the most successful in score) did. Since all other agents (participating in the same auction) requested only one night, they could not apply High-Low bids to hold the prices low. So the solution applied to solve this problem was to modify the bidding module so as to issue High-Low bids whenever more than one items are requested. Even further in cases where $Hwon$ value ensures that items requested are going to be transacted, the new low price was allowed to be even lower in order to stop prices from going up. This aspect of the strategy was named price stopper heuristic.

6. CONCLUDING REMARKS

In total nineteen teams, representing academic institutions, research centers and companies [2] participated in the competition. After incorporating all the above modifications, Thalis participated in the semi final group 2 games and was placed third in this group, thus qualifying for the finals. The main conclusion from those games was that most of the successful participants followed the same basic tactic as Thalis: they lowered the prices in order to achieve high scores. Those that did not follow this tactic were left out of the finals, even if their strategy had been successful during the course of the seeding rounds (e.g. ATTac). The final round of games were staged in two servers simultaneously and in total 32 games were played. The results are shown in Table 8. After successfully participating in the last two competitions [5] we hope we will be able to participate in next year's event whose market scenario is going to change from a travel market scenario to a supply chain one.

[3]High-Low bid: The bid includes one bundle with high price and one bundle with low price.

7. REFERENCES

[1] TAC game description. http://www.sics.se/tac/gamedescription.php.

[2] TAC list of entrants. http://www.sics.se/tac/tacparts.php.

[3] M. Boman. Trading agents. *AgentLink Newsletter*, 6:15–16, 2001.

[4] M. Fasli. Trading agent competition 2001. *AgentLink Newsletter*, 9:13–14, 2002.

[5] M. Fasli, I. Korres, M. Michalakopoulos, and G. Rallidis. Trading agents: Challenges and strategies. In *In Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, pages 63–67, 2002.

[6] A. Greenwald and J. Boyan. Bidding algorithms for simultaneous auctions: A case study. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 115–124, 2001.

[7] A.G. Guttman, R.H. Moukas and P. Maes. Agent mediated electronic commerce: a survey. *Knowledge Engineering Review*, 13(2):147–159, 1998.

[8] N. Poursanidis. Implementation of a software agent for the 2002 trading agent competition (TAC 2002). MSc Thesis. University of Essex. Department of Computer Science, September 2002.

[9] M.P. Wellman, P.R. Wurman, K. O'Malley, R. Bangera, S.-D. Lin, D.M. Reeves, and W.E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001.