

Strategies and Heuristics Used by the UMBCTAC Agent in the third Trading Agent Competition *

Li Ding, Tim Finin, Yongmei Shi, Youyong Zou, Zhongli Ding, and Rong Pan
Computer Science and Electrical Engineering Department
University of Maryland, Baltimore County
{dingli1, finin, yshi1, yzou1, zding1, panrong1}@csee.umbc.edu

Abstract

The UMBCTAC agent was one of the top ranked agents in the third international Trading Agent Competition (TAC'02). This paper describes and evaluates the key heuristics used by UMBCTAC, including the early bird heuristic, the balance heuristic, and the separation heuristic. We developed a simple gain-risk model to search safe and profitable allocations for hotel rooms and airline tickets. We also used a novel probabilistic approach to dynamically allocate entertainment tickets and bid in entertainment auctions. We conclude with a description of ongoing and planned work.

1 Introduction

The trading Agent Competition (TAC) is a market simulation game proposed by Wellman and Wurman [1999] with the first competition held in the summer of 2000 [Stone and Greenwald, 2001]. The second and third TACs [Wellman *et al.*, 2002a; Greenwald, 2003], which were held in the subsequent years, maintained research issues in simultaneous interrelated auction context, and had minor modifications for further research. The fourth competition initiates new research issues in supply chain management context [Raghu *et al.*, 2002] and keeps the original TAC framework under the name “TAC Classic”.

TAC Classic focused on automated strategies for software trading agent. A trading agent assembles a round-trip travel package for each of its customers by trading goods in multiple concurrent and interrelated auctions. TAC runs in client/server mode: a game server generates eight customers for each trading agent, and runs twenty-eight simultaneous auction instances which supply travel goods. On the client side, a human implemented trading agent acts on behalf of its customers, ordering airline tickets, hotel rooms and buying/selling entertainment tickets. The payout comes from a known utility function. This game was highlighted by its three types of auction mechanisms: eight continuous one-sided auctions on air-

line ticket (supply is unlimited during the game, and prices tend to rise over time), eight standard English ascending multi-unit auctions on hotel room (with auctions closing in random order); twelve standard continuous double auctions on entertainment ticket (both buying and selling are allowed during game). The trading agent needs to allocate and buy its customers' travel packages within limited time. The performance of a trading agent is evaluated by the profit obtained according to utility function. Details about TAC Classic are described by Wellman *et al.* [2001] and Eriksson and Janson [2002].

The rest of paper is organized like following: second section discusses the heuristics used by the UMBCTAC agent; section three describes price estimation techniques; section four and five elaborate the hotel/airline allocation strategy and the entertainment allocation/bidding strategy respectively; Section six concludes our accomplishments and suggests future work.

2 The heuristics

In TAC'02, “the most successful agents were primarily heuristic-based and domain-specific” [Greenwald, 2003]. The originally NP-complete optimization problem became more tractable when we used the domain-specific heuristics. The UMBCTAC agent is designed with the goal of being *simple* and *safe*. The agent should be *simple* because the complex situation and limited game history do not allow us to derive a comprehensive solution without over-fitting, but also because the real-time context requires fast response. The agent should be *safe* because the risk of allocating more resource has extremely high penalty, but also because the uncertain context make the price prediction unreliable. The overarching idea which guided our design was to maintain a balance between optimizing for a good solution and a safe solution. We found three heuristics to be useful – the *early bird heuristic*, the *balance heuristic* and the *separation heuristic*. We will describe each in turn and discuss its value.

2.1 Early bird and cautious bidder

Resource allocation is the most important part of a trading agent's strategy since subsequent bidding actions greatly rely on it. There are two candidate heuristics –

* This work was partially supported by DARPA contract F30602-97-1-0215 and Fujitsu Labs of America.

early bird and *deliberate buyer* (they are called “open-loop” and “closed-loop” by Stone *et al.* [2002] respectively). A trading agent using the *early bird heuristic* decides on a resource allocation at the very beginning of game and does not change it. This heuristic was identified as contributing to LivingAgents’ [Fritschi and Dorer, 2002] success as the top scorer in TAC’01. The heuristic relies on *perfect prediction assumption*, which means that a trading agent can correctly predict the “exact” clearing price for each auction at the beginning of a game. The assumption guarantees the optimality of static resource allocation. Moreover, once the resource allocation has been settled, the trading agent can focus on implementing a bidding strategy to produce best profit. However, the assumption is not always true in TAC games because of the game’s intrinsic uncertainty. Moreover a trading agent could suffer significant losses if its static allocation ordered many goods in auctions which have a very high clearing price. An alternative is the *deliberate buyer heuristic* -- an agent continually modifies its resource allocation and bidding actions according to the change of context. Theoretically, this heuristic can produce better allocation since it collects run-time game information. However, it incurs the cost of delayed decisions, such as the rising airline ticket prices (see Figure one), and missing out on good deals.

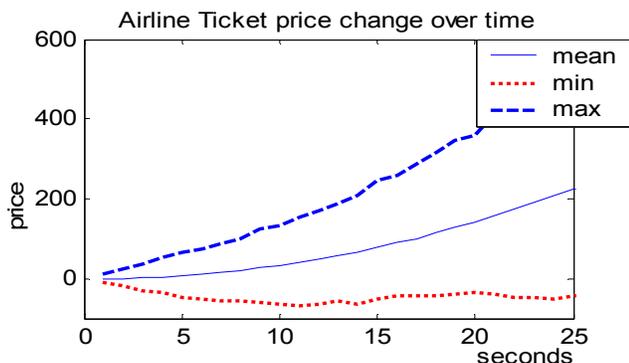


Figure 1: Airline ticket price increases exponentially over time, and its variance is quite large (this figure is based on the result of 10,000 controlled experiments, which set start price as \$0).

It is interesting to compare the top two scorers in TAC’01, Living-Agents and ATTAC, that employed the above two heuristics respectively. Stone *et al.* [2002] and Wellman *et al.* [2002] compared both and concluded that: (1) their performance is affected by the variance in hotel room auction clearing price; (2) their performance is affected by the ensemble of game participants; (3) the deliberate buyer has better theoretical performance, but its practical performance is sensitive to its implementation. In TAC’02, the top scorers mixed the two heuristics: (1) compose travel plans early, buy most airline tickets but delay purchasing “risky” airline tickets to allow reallocating resource later. (e.g. ATTAC [Stone *et al.*, 2002] and Whitebear [Vetsikas Selman, 2003]); (2) switch among different heuristics according to the prediction of competitiveness of game context (e.g. SouthamptonTAC

[He and Jennings, 2003]); (3) use early bird heuristic with safety consideration in hotel/airline auction, and use cautious bidder heuristic in entertainment auction (e.g. UMBCTAC [Ding *et al.*, 2002]). The success of these approaches is rooted not only by the ability of predict accurately, but also by the ability of avoid/handle risk, especially not buying hotel rooms in very high price.

2.2 The balance heuristic

The TAC game provides an interrelated and uncertain context for the trading agents: the utility function imposes tight relations among the goods; the agent constitution of a game directly affects the auction price; and the random closing order of hotel auctions increases the uncertainty in resource supply. We need a good resource allocation method with good performance in spite of the incomplete and uncertain context.

It is interesting to study the correlations of three economic terms: demand, price, and supply. Demand is the quantity of goods the buyers wants, supply is the quantity the sellers wants to sell, and price is either the market selling or buying price. In one-sided auction, where the supply is fixed, when the buyers’ demand is more than supply, the sell price rises until enough buyers quit or auction closes. In double auction, the three terms affect one another. When supply can’t satisfy demand, the selling price rises. When the selling price is high enough, more people might want to sell and thus increase the supply. As soon as supply overwhelms demand, the selling price will drop, attracting greater demand. Therefore, the supply might very likely be less than the demand again. Such casual relations dominate the market dynamics.

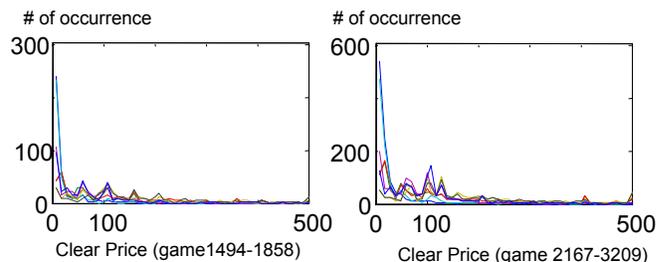


Figure 2: Clear price distributions of eight hotel auction in TAC’02 qualify round (left one) and seeding round (right one). The \$500 entry corresponds to all clear prices larger than \$500.

Figure two shows the clear price distributions of eight hotel auctions in TAC’02. Since hotel auction is one-side auction with fixed supply, the sooner the demand is reduced to no more than supply, the lower the clearing price will be. The peaks in the curves reflexes the “give-up point”, i.e. some agent would give up bidding higher at that price, and thereby reduce the demand. The flat parts of the distribution are caused by the random close order of hotel auction. It is obvious that the less the demand is, the lower the clear price will be.

The balance heuristic requires a trading agent to keep balance between profit and safety, i.e., its resource allocation should be profitable as well as safe. The *profit* consideration chooses the most profitable (estimated) resource allocation. The *safety* consideration chooses the resource allocation with less risk, i.e., restrict demand to be within average supply (overall supply divided by the number of participant agents). The balance heuristic is very important since UMBCTAC has fixed its hotel/airline allocation at the very beginning and it will definitely buy such goods regardless of the price.

The outcome of balance heuristic is straightforward. First, since the trading agent will not intentionally break the overall balance between demand and supply, the TAC game more easily remains in a “normal” state with final clearing prices close to their historical average. Second, even when the balance is broken, a trading agent suffers less than average loss. Finally, while a trading agent may not be the outstanding profit maker, its performance will be a statistically above-average. In the TAC’02 record, the UMBCTAC agent ranked second in the qualification round (120 games), third in the seeding round (440 games)¹, and fourth in the finals (32 games).

2.3 The separation heuristic

In the TAC game, there are three types of goods that together affect the final profit. If all these factors are considered with the full dynamics of the TAC game, the computational cost will be too expensive and the corresponding delay will be unacceptable. The separation heuristic is used to solve the complexity problem by simplifying the resource allocation process: handle loosely related auctions separately.

Table 1: A customer’s 20 possible travel schedules. (The number in AD, DD column corresponds to a weekday, e.g. 1 means Monday.)

ID	AD	DD	Hotel	ID	AD	DD	Hotel
1	1	2	SS	11	1	2	TT
2	2	3	SS	12	2	3	TT
3	3	4	SS	13	3	4	TT
4	5	6	SS	14	5	6	TT
5	1	3	SS	15	1	3	TT
6	2	4	SS	16	2	4	TT
7	3	5	SS	17	3	5	TT
8	1	4	SS	18	1	4	TT
9	2	5	SS	19	2	5	TT
10	1	5	SS	20	1	5	TT

Firstly, the UMBCTAC agent separates the hotel/airline auctions from the entertainment auctions. This heuristic came from following observations. (a) Separation can greatly reduce search complexity. A customer has ten legal choices for a travel date. Since the customer can’t change hotel in Tampa, he has two choices for hotel type, i.e., either a good hotel (denoted by TT) or a cheap

¹ UMBCTAC had very bad results in six of the 440 games because of network failure.

hotel (denoted by SS). Therefore a customer has altogether 20 possible legal travel schedules as shown in Table one (note that “not go to Tampa” is also possible but not included). However, his choices increase greatly when considering the allocation of entertainment ticket. For example, a customer who spends three nights in Tampa can have any of 60 (i.e., 5*4*3) possible entertainment ticket allocations for his trip. (b) The travel schedule is dominated by hotel/airline allocation, and trading agents rarely extend trips just for more entertainment bonus. The origin of this separation heuristic can be traced back to Greenwald and Royan [2001].

Secondly, the UMBCTAC agent separates the entertainment auctions and handles each independently. This heuristic evolved from observations in the continuous double auction (CDA) [Friedman and Rust, 1993; Wurman *et al.*, 1998; Smith *et al.*, 2002] in TAC: (a) no globally optimal allocation – not only the inherent randomness of price/supply in CDA but also the possibility that trading agent changes its resource allocation -- could greatly affect the global entertainment ticket supply. (b) Fast response was preferred. A good deal can only be caught by the first agent who takes action.

As a divide-and-conquer method, the separation heuristic simplifies and accelerates decision procedure, but also suffers from local optima, which can’t guarantee global optimality of resource allocation.

3 Estimating score for travel packages

In TAC, a travel package is scored through utility function (see also Table two) that is listed in the game specification on TAC web site.

Table 2: The utility function (from TAC game description)

<p>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</p>
--

A customer’s preference is generated by game server when a game starts, including preferred arrival date (PA), preferred departure date (PD), bonus for booked good hotel (HP), and bonus for obtained entertainment tickets (AW, AP, and MU). If the travel schedule, including actual arrival date (AA), actual departure date (AD), and hotel assignment (TT?), is determined, we will know the *travel_penalty* and *hotel_bonus*. However, we still need to estimate the *fun_bonus* and *cost*. Since we buy airline tickets at the beginning of game, and airline ticket price is always available, so we only need to estimate *hotel_cost* and *fun_cost*.

3.1 Estimating the fun_bonus and fun_cost

For a trading agent, if all customers have their travel schedule fixed (i.e. AA, AD are known) and the enter-

tainment tickets in hand don't change, it is easy to use a LP solver to find the best entertainment allocation. However, entertainment tickets are traded in double auctions with undetermined supply, and a trading agent might re-schedule any of its customers' travel packages during game. So the UMBCTAC agent uses a probabilistic method to solve such resource allocation problem as described in section five.

The UMBCTAC uses a simple formula (see Equation one) to estimate the entertainment profit, defined as (fun_bonus - fun_cost). We use bonus(C,E) to denote the bonus a customer C offered over entertainment E. We use fun_profit(C,E,D) to denote the profit a trading agent can make from customer C over entertainment E on day D. Note that day D should be within the customer's travel schedule. We also use a threshold T to determine if the customer's bonus is sufficient to let the trading agent obtain corresponding entertainment ticket from auction. HasTicket(E,D) means the trading agent has the ticket for entertainment E on day D in hand.

Equation 1: fun profit

$$\text{fun_profit}(C, E, D) = \begin{cases} \text{bonus}(C, E) & \text{if } (\text{bonus}(C, E) > T) \text{ and hasTicket}(E, D) \\ \text{bonus}(C, E) / 2 & \text{if } (\text{bonus}(C, E) > T) \text{ and not hasTicket}(E, D) \\ 0 & \text{if } \text{bonus}(C, E) \leq T \end{cases}$$

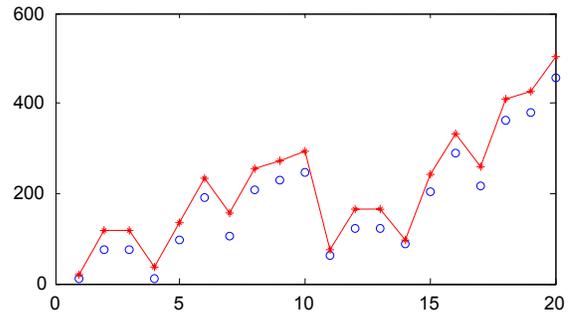
The trading agent partially counts the bonus without having tickets in hand because its offered buy bid is always good enough to obtain the desired tickets in entertainment auction. To ensure the seller never makes more profit than the buyer (since the seller is also a competitor), a trading agent's buy bid shouldn't larger than half of the bonus. The final entertainment profit for a customer is the best combination of fun_profit(C,E,D).

3.2 Estimating hotel cost

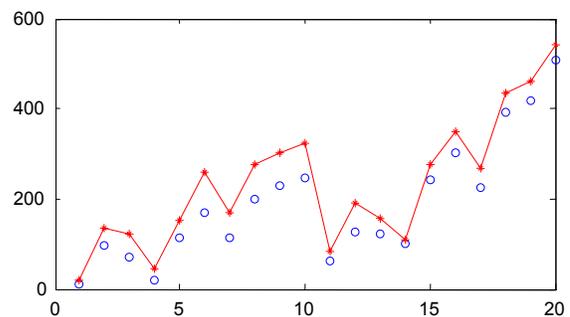
Hotel cost is very important for resource allocation decision and is also hard to predict, even when we know the game history. Stone *et al.* [2002] discussed some approaches predicting hotel cost. We chose simple statistical average, mean and median, to predict the clearing price of hotel rooms. Our approach predicts the clearing price for each of the 20 possible travel schedules (see section 2.3 for details). Note that each travel schedule has one unique type of hotel room allocation, e.g., a travel schedule which as AD=(1,3,TT) means that we need to book a room in hotel TT at Monday and Tuesday night.

Figure three shows average clearing price with respect to a customer's 20 possible travel schedules. From that figure, we have following observations: (1) short travel schedules (stay in Tampa for one or two days), which demand less hotel room, cost less; (2) cheap hotel (SS) costs less; (3) the clearing price distribution of 20 travel schedules doesn't change much over time, i.e., Figures 3a and 3b have similarly shaped curves; (4) the median is always less than mean, i.e. more than half of games have clearing prices less than mean and the rest have very high

clearing prices. (5) While the *median* is too optimistic because it ignores the potential risk of very high clear price, the *mean* is somewhat pessimistic because it overlooks the average clearing price by counting outliers with extremely high clearing price.



(a) Hotel price based on 1000 seeding games(2167-3209)



(b) Hotel price based on 100 seeding games (3110-3209)

Figure 3: Average clear price w.r.t a customer's possible travel schedules (the data was collected from TAC'02 seeding round). X axis corresponds to 20 possible travel schedules' IDs (see table 1 for detail). Y axis is clear price. Stars connected by line denote mean value, and circles denote the median.

Since customer preferences are generated randomly, the composition of game participants dominates the game statistics. In addition, the internal design of participant agents also evolved during the TAC game. So the statistics from 100 recent games is more relevant than that of 1000 recent games. When estimating hotel price, the UMBCTAC agent favours short travel schedules, uses the median to predict the clearing price of short ones, and uses the mean for the others. The historic average is widely used in TAC'02 due to its simplicity. Other prediction approaches is discussed by Wellman *et al.* [2002b].

4 Hotel/Airline auction strategy

Because of the close relation between hotel and airline auctions, UMBCTAC handles them together according the heuristics in section two. The approach is simple: decide the hotel and airline allocation at the beginning of

game and do not change it. This allocation should be both safe and profitable.

To achieve this goal, UMBCTAC used the *Gain-Risk Model*, which has three important components: gain estimation, risk estimation and heuristic search. Gain refers to the sum of estimated score for the one resource allocation (see Section three). Risk refers to the probability of not being able to make profit by so doing.

4.1 Estimating risk

An allocation refers to assignment of goods (hotel rooms and airline tickets) to the trading agent’s customers. Once the trading agent has decided travel schedules for all its customers, the allocation is determined. The UMBCTAC agent uses thresholds and associated weights to quantify risk. For a given allocation c , we compute the risk of each hotel auction x , which is denoted by $Risk(c,x)$, and then we sum them as the overall risk, denoted by $Risk(c)$ (see also Equation two). We use $Alloc(c, x)$ to denote the number of rooms allocated in auction x . Each auction x has associated risk threshold $T(x)$ and risk weight $W(x)$.

Equation 2: Risk of allocation

$$Risk(c, x) = \begin{cases} [Alloc(c, x) - T(x)] \cdot W(x) & \text{if } Alloc(c, x) > T(x) \\ 0 & \text{otherwise} \end{cases}$$

$$Risk(c) = \sum Risk(c, x)$$

The threshold $T(x)$ and weight $W(x)$ are empirically determined constants that arose from the following considerations: (1) for each hotel auction, the average room supply is two. According to our balance heuristic, allocating more rooms will increase risk; (2) according to our observation in hotel price prediction (section 3.2), it is unwise to take the high risk of demand more than average supply since we can’t make profit in the long run by doing so; (3) long travel schedules result in high risks in multiple hotel auctions, and the corresponding historical hotel clearing price (median) is higher than the sum of its components’.

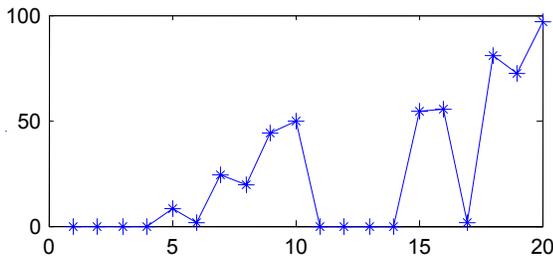


Figure 4: Median difference. X axis corresponds to 20 possible travel schedules. Y axis shows the difference. (This figure is based on game data 3110-3209 in TAC’02)

Figure four shows the difference between the actual median and the sum of one-day travel schedules’ medians; (4) days two and three have higher risk than days one and

four because they are always in higher demand by all 64 customers in a TAC game; (5) travel plans matching or subsumed by the preferred time frame are typically more profitable. We observe that risk increases only when the room allocation exceeds threshold $T(x)$. Furthermore, weight $W(x)$ is assigned to auction to scale the risk value according to the probability of having risk.

4.2 Heuristic search for best gain and risk

Since there are two goals (safety and profitability) to optimize, the core of the Gain-Risk model is a multiple criteria optimization problem [Steuer, 1986]. One possible solution is to use multiple objective linear programming (MOLP). The alternative solutions are classical AI search techniques, such as A* or beam search. The UMBCTAC agent runs a relatively simple heuristic search which has two stages.

In the first stage, we prune those “unfavorable” travel schedules. For each customer, we use the *favor-short-trip* and *change-trip-slightly* heuristics to select favored travel schedules among the 20 candidates. The favor-short-trip heuristic only selects the travel schedules that match or are subsumed by the customer’s travel preferences. The change-trip-slightly heuristic avoids introducing high travel penalties. In our practice, a customer normally has approximately three favorite choices (note that the number of favorite choices varies for different trip length: a one day trip has one choice, a two day trip has two of three choices, a three day trip has three choices and a four day trip has five choices).

In the second stage, an exhaustive brute force search is used to find the safest combination of travel schedules for all eight customers. The search is viable because the search space is reduced to approximately 3^8 after stage one. So the candidate combination with lowest risk will be selected (if multiple candidates have same risk value, then we choose the one with largest gain).

According to our experiments, the safest allocation reached in the second stage also has near-optimal profit (see Figure five). We have found, however, that this algorithm is robust to perfect prediction assumption only when it has reliable statistical average. Lack of examples will cause its bad performance.

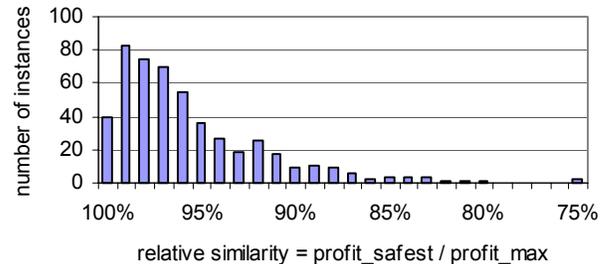


Figure 5: The distribution of relative similarity (based on 500 controlled samples). The average is 96%.

5 Entertainment auction strategy

The entertainment auctions are handled individually after the hotel and airline resources have been allocated.

5.1 Probabilistic resource allocation

Instead of globally assigning e-tickets (entertainment tickets) to customers, the UMBCTAC agent holds e-tickets and dynamically distributes them to its customers with certain probabilities. The algorithm in Table three both allocates e-tickets probabilistically and returns customers' *offer price*, which is the best price the customers can offer for buying a ticket from the auction.

Table 3: The probabilistic allocation algorithm

```
ProbabilisticAlloc (TicketOwned)
1. count = TicketOwned
2. get all clients who need e-ticket in that auction
3. candidates= sort the clients by their bonus
4. for each client in candidates in descending order
5.   offer_price = client.bonus
6.   if count is 0 then break;
7.   with probability of (1/trip_length) count--
8. end for
9. return offer_price
```

5.2 Probabilistic buy and sell

The UMBCTAC agent uses *desire probability* to represent the desirability of selling and buying. The ordering of selling outcomes with respect to their desirability is as follows. Selling with a high price is most preferred; selling with reasonable price is less desirable; not selling is acceptable; but selling with a low price is undesirable. The same idea applies to buying strategy.

Given the number of owned tickets k , we defined desire probability as $P(k)=0.9^{L(k)}$, where $L(k)=3^k$. For example, when the UMBCTAC agent has fewer than two e-tickets it tends to buy, otherwise it tends to sell.

The UMBCTAC agent also uses a *price range* to provide additional control over the price convergence process in double auction. A *price convergence process* starts from the gap between buying and selling price, and then the two prices advance toward each other gradually and finally converge. The lower part of the price range can guarantee *minimum relative profit* (seller's profit should exceed the buyer's profit), and the higher part is the highest expected selling price, which will be post on the market as selling bid. The higher part is determined by the desire probability, relative game time (the percentage of game time passed) and customers' offer price.

5.3 The auction handler algorithm

For each auction, the UMBCTAC agent collects the number of owned tickets k , the market (buy/sell) price and the relative game time t . It can then compute its customers' offer price w , and thereby determines the price range, i.e., the buying price should always less than w ,

while the selling price should always greater than w . Moreover, it can be used to derive the desire probability. The auction handler algorithm is given in Table four.

Table 4: The auction handler algorithm

```
Handle-Entertainment-Auction
1.  $w = \text{ProbabilisticAlloc}(k)$ 
2. compute (low-buy, high-buy) price based on  $P(k)$ ,  $t$  and  $w$ 
3. with probability  $P(k)$ , we send a buy bid – either buy instantly if current ask price in auction falls between our acceptable range, or post the low price in the auction otherwise
4. compute (low-sell, high-sell) price based on  $P(k)$ ,  $t$ ,  $w$ 
5. with probability  $P(k)$ , sell ticket instantly if current bid price in auction falls between our acceptable range, or post a sell bid with high price otherwise.
```

In the TAC'02 finals, the UMBCTAC agent did not achieve good entertainment profits [Cheng *et al.*, 2003]. Our conclusions are: (1) the algorithm is too simple; (2) we always shortened customers' trips, and shorter travel schedules have less entertainment profit. In fact, the entertainment profit is affected by multiple factors: the hotel and airline allocations, the entertainment tickets allocations and the bidding algorithm.

6 Conclusions and future work

The UMBCTAC agent employs simple heuristics to achieve above average behavior. Its performance in TAC'02 conforms to our expectation: not the best but the statistically above-average player. We believe that domain specific heuristics are the keys to solving the complex optimization problem in TAC. It is not a coincidence that the top scorers took advantages of "risk analysis" in TAC'02. Moreover, the heuristics and the optimization problem co-evolve -- when the agents have improved their heuristic, the optimization problem evolves! In TAC'01, good "price prediction" led to optimal profit, and in TAC'02, good "risk analysis" and "entertainment exchange" led to optimal profit. Lanzi and Strada [2002] made the interesting observation that TAC has "pack of winners" rather than "a single significant winner". Isn't that because the winners took advantages from the "good" heuristics? We expect to find some theoretical ground for this phenomenon, and economic theories might be the most promising potential.

According to Greenwald [2003], the two major solutions used in TAC are heuristics [Greenwald and Royan, 2001] and integer linear programming [Stone *et al.*, 2001]. UMBCTAC belongs to the heuristic group and is basically a risk-preventing early bidder. However, its Gain-Risk model is still useful for the deliberative buyer because it relies less on the perfect prediction assumption. The formula for evaluating risk is not yet theoretically sound because it simply sums the individual risks where a multiplication might be more appropriate. The

safety probability of a travel package might better come from the multiplication of prior safety probabilities of each affiliated hotel auctions (note that $\text{risk} = 1 - \text{safety}$). Our future work will include improving methods for evaluating risk and introducing better search methods.

The probabilistic bidding approach worked fairly well in TAC'02 and we continued to improve it until the end the seeding round. It simulated the human decision process and provided a fast and reasonable near-optimal solution for resource allocation. In future work we will explore its theoretical underpinning.

Acknowledgments

Our research was partially supported by DARPA contract F30602-00-2-0591 and Fujitsu Laboratory of America. Professors Yun Peng, Anupam Joshi, and Tim Oates provided valuable help and advice.

References

- [Cheng *et al.*, 2003] Shih-Fen Cheng, Evan Leung, Kevin M. Lochner, Kevin O'Malley, Daniel Reeves, and L. Julian Schwartzman, Walverine: A Walrasian trading agent. To appear in Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, 2003.
- [Ding *et al.* 2002] Li Ding, Yongmei Shi, Zhongli Ding, Rong Pan, Tim Finin, UMBCTAC: A Balanced Bidding Agent, UMBC Technical Report TR-02-15, 2002.
- [Eriksson and Janson, 2002] Joakim Eriksson and Sverker Janson. The Trading Agent Competition - TAC 2002. ERCIM News, 51, October 2002.
- [Friedman and Rust, 1993] Daniel Friedman and John Rust, editors. The Double Auction Market: Institutions, Theories, and Evidence. Proceedings Volume XIV in the Santa Fe Institute Studies in the Sciences of Complexity, 1993.
- [Fritschi and Dorer, 2002] Clemens Fritschi and Klaus Dorer. Agent-oriented software engineering for successful TAC participation. In First International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2002.
- [Greenwald and Royan, 2001] Amy Greenwald and Justin Boyan. Bidding algorithms for simultaneous auctions. In Proceedings of Third ACM Conference on E-Commerce, pages 115-124, Tampa, FL, 2001.
- [Greenwald, 2003] Amy Greenwald (ed.). The 2002 Trading Agent Competition: An Overview of Agent Strategies. AI Magazine, 24(1): 83-91, 2003
- [He and Jennings, 2003] M. He and N. R. Jennings, SouthamptonTAC: An adaptive autonomous trading agent. To appear in ACM Trans on Internet Technology, 2003
- [Lanzi and Strada, 2002] Pier Luca Lanzi and Alessandro Strada, A Statistical Analysis of the Trading Agent Competition 2001, In SIGecom Exchanges 3(2):1-8, 2002
- [Raghu *et al.*, 2002] Raghu Arunachalam, et al, The TAC Supply Chain Management Game (Draft version 0.5), 2002.
- [Smith *et al.*, 2002] Eric Smith, J. Doyne Farmer, László Gillemot, and Supriya Krishnamurthy, "Statistical Theory of the Continuous Double Auction", SFI Working Paper Abstract, 2002
- [Steuer, 1986] Steuer, R. Multiple criteria optimization: Theory, Computation and Application. Chichester, John Wiley & Sons: New York.
- [Stone *et al.*, 2001] Peter Stone, Michael L. Littman, Satinder Singh, Michael Kearns, ATTac-2000: An Adaptive Autonomous Bidding Agent, Journal of Artificial Intelligence Research 15:189-206, 2001
- [Stone and Greenwald, 2001] Peter Stone and Amy Greenwald (ed.). Autonomous bidding agents in the trading agent competition. IEEE Internet Computing, 5(2):52--60, 2001.
- [Stone *et al.*, 2002] Peter Stone, Robert E. Schapire, János Csirik, Michael L. Littman, David A. McAllester: ATTac-2001: A Learning, Autonomous Bidding Agent. AMEC 2002: 143-160
- [Vetsikas and Selman, 2003] Ioannis A. Vetsikas and Bart Selman. A principled study of the design tradeoffs for autonomous trading agents. To appear in Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, 2003
- [Wellman and Wurman, 1999] 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
- [Wellman *et al.* 2001] Michael P. Wellman, Peter R. Wurman, K O'Malley, R Bangera, S Lin, D Reeves, and William E. Walsh, Designing the Market Game for a Trading Agent Competition. IEEE Internet Computing, March/April (2001), 43-51
- [Wellman *et al.*, 2002a] Michael P. Wellman, Amy Greenwald, Peter Stone and Peter R. Wurman, The 2001 Trading Agent Competition, In Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-02), pages 935-941, Edmonton, 2002
- [Wellman *et al.*, 2002b] Michael P. Wellman, Daniel M. Reeves, Kevin M. Lochner, and Yevgeniy Vorobeychik. Price prediction in a trading agent competition. University of Michigan, 2002.
- [Wurman *et al.*, 1998] Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. Decision Support Systems, 24:17--27, 1998