An agent for online auctions: bidding and bundling goods for multiple clients

By

Chi-Lun Chau

A thesis

Submitted In Partial Fulfillment of the Requirements

For the degree of master of philosophy

Division of Information Engineering

The Chinese University of Hong Kong

Mar 2003

Abstract

Bidding and bundling profitable goods for multiple clients from online auctions is a complicated task. First, goods are sold under various mechanisms like English or sealed-bid auction simultaneously. An agent has to handle multiple trading mechanisms (auction protocols) concurrently. Second, client preferences are over bundles. An agent has to consider interacting relation of goods inside the bundles. Third, an agent has to allocate goods to multiple clients in the most profitable way. Several goods inside a bundle are not currently held but can be acquired later. The agent faces a complicated allocation problem which involves both held and to be acquired goods items. Moreover, an agent faces a game playing problem as opponent's strategies are unknown.

This thesis proposes a general agent model to bid and bundle goods for multiple clients from online auctions. The agent model decomposes the whole problem into several sub-problems, which can be handled by a set of mechanisms. The mechanisms include referencing historical clearing prices, increasing marginal costs for uncertain goods, bidding uncertain goods at low prices and considering bid winning probabilities. This thesis also studies interactions between agents with binary decision: being aggressive or adaptive. The results show that the minority has an advantage in a market under binary decision. For example, aggressive agents perform better when there are too many adaptive agents.

In Trading Agent Competition (TAC '02), we implemented an agent called "CUHK" to participate in the tournament and it was one of the finalists. The agent was derived from our agent model. This thesis presents the results obtained from both TAC '02 competition and our controlled environment. The TAC '02 competition result indicates that our agent model performs well even compared to world-leading research groups. The results from controlled environment show that our agent mechanisms are effective in both competitive and non-competitive markets.

摘要

在網上拍賣中，爲不同顧客合拼出有盈利的組合並出價投標是一項複雜的工作。首先，物品遵循不同的機制進行拍賣，例如傳統的英式競投和暗標式競投。其次，顧客的評價基於整套組合而非單一物品。因此，代理在投標時必須考慮物品在組合中的相互作用和關係。第三，代理必須根據顧客的組合評價，以最理想的方式分配物品。因爲當中包括已擁有的及未取得的物品，所以代理需要面對相當複雜的分配問題。此外，代理無法知道競爭對手的策略，所以它也要面對不同的對策性問題。

這篇論文提出了一個代理的設計模型，爲不同顧客合拼出有盈利的組合並出價投標。這個設計模型首先把整體問題分解爲多個子問題，再運用不同的技巧和手法進行處理。這些由我們研究出來的技巧和手法包括：參考物品的歷史結算價格、考慮物品的取勝概率、對價格不確定的物品採用上升的邊際成本以及進行低價競投等。這篇論文對不同策略的相互作用也作出了分析和研究，結果顯示在兩擇其一的條件下，市場有利於少數群體。

在二零零二年度交易代理比賽 (Trading Agent Competition 2002) 中，我們根據以上設計模型完成了代理，參與賽事並進入決賽。本文將陳述我們在比賽及控制環境下所取得的結果。實驗証明我們的設計模型在眾多競爭對手組合中亦表現理想。

Acknowledgement

Gratitude is expressed to my supervisor, Professor Wai-yin Ng, who has given me direction, support and guidance in this research. This thesis is completed with his professional advice on technical contents and thesis writing.

Thanks are also given to my co-supervisor, Prof. Ho-fung Leung, who has been continuously providing me encouragement and enlightenment on this research. The informative advice and intriguing discussions provided me a lot of insights in conducting this research.

Finally, I wish to say "Thank you" to all fellow researchers and classmates who have provided me confidence and help in this research and thesis.

Table of Contents

List of Figures

Definitions

Online auction:	An auction takes place in an electronic market, in which clients are participating via communicating channels such as the Internet
Online bidding agent:	A "software robot" assists its client(s) to participate in online auction(s). It is capable of autonomous actions including bidding, pricing or monitoring online auctions
Goods:	Objects traded in online auctions
Item:	A single unit of goods
BID:	The current bid quote. In general, it refers to the highest bid among all participants at current time
ASK:	The current asked quote. In general, it refers to the lowest ask among all participants at current time
Clearing price:	The trading price of an item when BID matches ASK
Single-sided auction:	An auction where participants are either buyer or seller, but not both at the same time
Double-sided auction:	An auction where a participant may be both buyer and seller at the same time
Marginal cost:	Increase in total cost for an additional goods item
Sunk cost:	Cost spent, that is not related to future decisions
Opportunity cost:	The highest valued option foregone for a decision

Chapter 1 – Introduction

1.1 Background

A search for online auction houses in Google[1] will probably return more than a thousand links. Online auctions become so popular since they can generate a huge amount of profit for auction houses. The total revenue generated by online auctions was about USD 8.4 billion in 2001 and predicted to reach USD 48.5 billion in 2006 (Surmacz, 2001). The increasing popularity of Internet brought numerous customers from worldwide to participate in online auctions. Besides, online auctions can also operate with existence of autonomous online agents, which is a kind of "software robot" to assist clients in bidding goods and monitoring auctions.

Unlike traditional auctions, participants in online auctions are from everywhere and are invisible to each others. More interestingly, participants do not always have good auction knowledge and bidding skills. They can aggressively bid their desired goods at high bidding prices. Alternatively, other participants may always respond to market changes and decide what the best bids are. Various participant combinations can lead to many possible outcomes. An online bidding agent faces a *game playing problem* since opponents' strategies are unknown.

The trading mechanism of an auction is called *auction protocol*. In online auctions, the most commonly used auction protocols include: English, Dutch, Vickery (Vickery, 1961), first-sealed bid and second-sealed bid auctions. *Simultaneous auctions or*

[1] www.google.com

parallel auctions refer to various auctions running at the same time. An agent faces a *multiple auction protocols problem* when it participates in simultaneous auctions using various auction protocols.

A *bundle* refers to a combination of goods. For example, a desktop computer is a bundle while CPU, RAM and hard disk are its components. An agent may have a client whose preference is over bundles. The agent faces a *bundle completion problem* when component goods are traded in various auctions. Goods inside a bundle are interacting, either *complementary or substitutable* (Greenwald & Boyan, 2001b). Let *u(G)* be a function which maps a type of goods G to an utility value. Complementary goods are goods with super-additive value, $u(A\overline{B})+u(\overline{A}B)<u(AB)$. Substitutable goods are goods with sub-additive value, $u(A\overline{B})+u(\overline{A}B)>u(AB)$. An agent has to decide the most profitable bundle for its client. At a particular time, the agent holds several components goods but not all of them. Since clearing prices of several goods are unknown, there is uncertainty in completing the bundles.

An *allocation* describes how an agent distributes goods to multiple clients. Each client has his/her private values on bundles. Allocating same goods to one client may generate more profit than allocating that to another client. The agent faces an *allocation problem* if it serves more than one client. At a particular time, several component goods are not currently held but could be acquired later. One possible reason is that corresponding auctions are still running. The allocating problem in simultaneous auctions is complicated because it involves both held and not held goods.

1.2 Testing Environment

The Trading Agent Competition (TAC) is chosen as our testing environment[2]. The (TAC) has been organized since 2000 which aims at providing a realistic and multifaceted benchmarking problem in e-market, with simple rules and interfaces (Wellman et al., 2001). In simple words, a TAC agent is going to construct travel packages for multiple clients. A travel package is a bundle of substitutable or complementary goods, which are traded in simultaneous auctions using various auction protocols. A TAC agent communicates to the TAC server via TCP/IP, obtaining marking information and submitting bids. As clearing prices of several goods are unknown, the most profitable travel packages are not determined until all auctions close. A TAC agent should be able to reason under a degree of uncertainty. The key challenges in TAC include clearing price prediction, value assessment of goods, allocation of goods, strategically bidding, risk management, machine learning and game playing. In TAC '02, we implemented an agent called "CUHK" to participate in the tournament. Although the agent made mistakes in the final round, it ranked the 8^{th} over 19 teams from world-leading research groups. The agent would rank 4^{th} position if no mistakes were made.

1.2.1 Game Overview

The whole TAC game consists of many TAC game instances. Each game instance lasts for 12 minutes and involves 8 trading agents. In a game instance, each TAC agent represents 8 clients who are planning to travel Tampa from TAC town within a national 5 day period (day 1-5). As a result, there are altogether 64 clients in a game

[2] For full detail, see http://www.sics.se (TAC '02) and http://auction2.eecs.umich.edu (TAC '01)

instance. A travel package is a bundle consists of inbound and outbound return flight tickets, hotel room reservations and entertainment tickets. The trip duration should be at least one day. No hotel reservation and entertainment event is needed on return day. Consequently, there is no outbound ticket at day 1, no inbound ticket at day 5, no hotel reservation at day 5 and no entertainment ticket at day 5. The flight tickets, hotel room reservations and entertainment tickets are traded using 3 different market mechanisms.

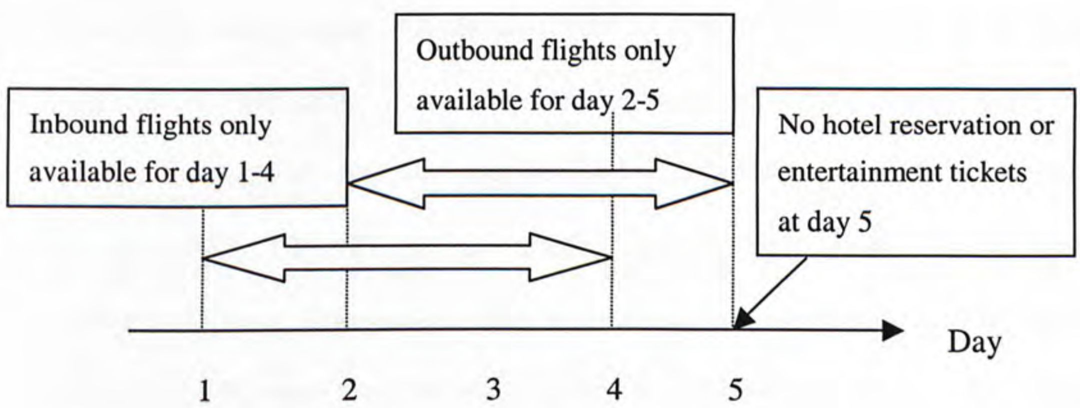


Figure 1: The TAC world

1.2.2 Auctions

- **Flights:** TACAir is the only airline providing flight tickets between TAC and Tampa. There are altogether eight markets for flight tickets, one market each direction per day (IN1, IN2... OUT5). All markets close at the end of a game instance. The market mechanism used is "take-it or leave-it" and flight tickets supply is unlimited. Tickets are traded simultaneously during a game instance. The trading agent acquires flight tickets by providing bids to TACAir. Transaction clears immediately once a bid is higher than the current flight ticket price, ASK. Otherwise, the bid is carried forward to the next round. The market can be considered as a special kind of "auction". The flight prices in each market are independent, having initial values between 250 and 400. The flight prices change by Δ every 24-32 seconds, where Δ is a random variable drawn from -10 to 90. In general, Δ follows a random walk but has an increasing trend. Resale of flight ticket is not allowed in TAC.

- **Hotels:** Two hotels are available in Tampa: Tampa Tower (TT) and Shoreline Shanties (SS). TT is a better hotel as it has a cleaner, more comfortable and more convenient environment. All clients prefer to stay in TT than SS for the same cost. Room reservations are sold at multi-unit simultaneous ascending auctions[3]. There are altogether eight auctions for hotel reservations, one auction each hotel per day (SS1, TT1...TT4). Each trading agent submits bids with its target quantity of goods. Transactions clear once at the end of an auction. The top 16 bids win the 16 room reservations at the 16th highest bidding price. In TAC game setting, one random auction for hotel reservations will be selected to close per minute since the end of 4th minute. All auctions for hotel reservations closed before the last minute (the 12th minute). The TAC server provides updated ASK and BID quotes to trading agents only once per minute. The ASK is the current 16th highest bidding price while BID is the current 17th highest bidding price. Other information such as higher bids is unknown for trading agents. A valid bid should satisfy 2 conditions: (i) offers to buy at least one unit at a price of ASK+1 or greater and (ii) offers buying quantity equals to or more than HQW (the agent's winning quantity if the auction closes now). Withdrawal of hotel bids and resale of hotel reservations are not allowed in TAC.

[3] The auction for hotel auctions is described as "standard English ascending multi-unit" in TAC '02 official page.

- **Entertainments:** There are 3 different kinds of entertainment events: Alligator wrestling (AW), Amusement park (AP) and Museum (MU). There are altogether 12 auctions for entertainment tickets, one each kind per day (AW1, AP1...MU4). The market mechanism used is continuous double auction (CDA). All 12 auctions close at the end of a game instance. Each trading agent has 12 tickets randomly assigned at the beginning of a game instance. The agents trade their entertainment tickets by providing bid and ask. Transactions clear immediately when one bid is over another ask. Unmatched bids become standing bids and are carried to the next round. The TAC server provides updated ASK and BID quotes simultaneously. The ASK is the current lowest ask and BID is the current highest bid. Both withdrawal of bids and resale of tickets are allowed. The market is much like a "stock market".

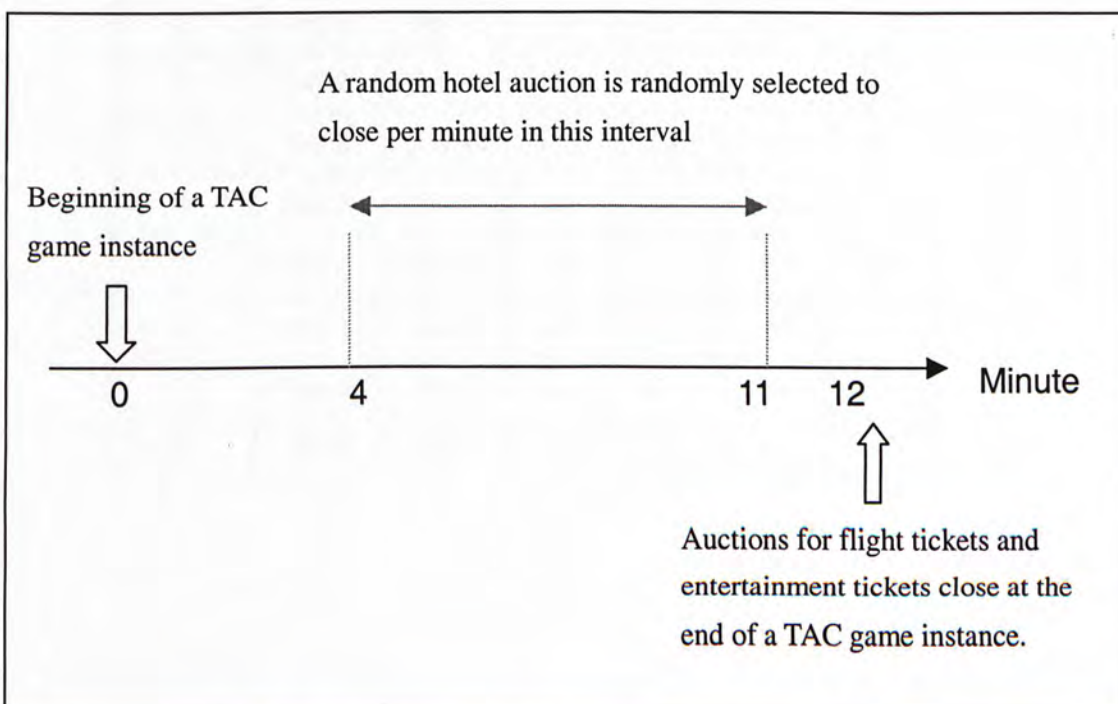


Figure 2: The timeline of a TAC game instance

1.2.3 Utility and Scores

In TAC, a travel package is described as *feasible* if it contains (i) an inbound ticket, (ii) an outbound return ticket and (iii) room reservations in same hotel within the whole tour.

Each client has his/her (i) preferred arrival (day 1-4) and return date (day 2-5), (ii) reward to stay in better hotel TT (50-150) and (iii) rewards to enjoy various entertainment events (0-200 each). Those clients' preferences are randomly generated and vary from client to client. (See table below for an example)

Table 1: The client preferences of our CUHK agent in game 452 at tac4.sisc.se.[4]

Client	PAD	PDD	HV	AWV	APV	MUV
1	Day 1	Day 2	76	65	129	170
2	Day 2	Day 4	119	9	168	43
3	Day 3	Day 4	121	26	68	97
4	Day 3	Day 5	117	54	160	97
5	Day 2	Day 3	50	35	49	102
6	Day 3	Day 4	109	69	166	182
7	Day 3	Day 5	121	104	93	139
8	Day 2	Day 3	51	75	72	0

[4] PAD and PDD stand for preferred arrival and departure date. HV stands for rewards to stay in better hotel. AWV, APV and MUV stand for the rewards to enjoy entertainment events AW, AP and MU respectively.

The client utility function, which measured in dollars, is defined as:

$$Utility = 1000 - travel\ penalty + hotel\ bonus + fun\ bonus$$

where

travel penalty $= 100 * (|AA - PAD| + |AD - PDD|)$[5]

hotel bonus = the client's reward to stay in hotel TT

fun bonus = sum of rewards from all entertainment events

If the client's allocated travel package is not at his/her preferred dates, there is a penalty of 100 per day. The client's hotel bonus is per travel package, not per night. For each client, duplicated tickets of same event counted once only within a trip. There should be no more than one event per night in a travel package. The client utility is zero if no feasible travel package is allocated to him/her. The score of a trading agent is the sum of all clients' utilities minus total expenditure (See Table 2 for an example).

Table 2: The result of the CUHK agent for game 452 at tac4.sics.se.

Client	AD	DD	Hotel	Entertainment	Utility	Cost	Score
1	Day 1	Day 2	TT	AP1	1205	520.90	684.09
2	Day 2	Day 3	TT	AP2	1187	920.07	266.92
3	Day 3	Day 4	TT	MU3	1218	802.61	415.38
4	Day 4	Day 5	TT	AP4	1177	756.51	420.48
5	Day 2	Day 3	SS	MU2	1102	895.35	206.64
6	Day 3	Day 4	TT	MU3	1291	802.61	488.38
7	Day 3	Day 5	TT	AW3, MU4	1364	630.21	733.78
8	Day 2	Day 3	SS	AP2	1072	806.60	265.39
				Sum	9616	6134.88	3481.11
				Other costs (unused goods, transaction losses, etc)		48.77	
				Total	9616	6183.66	3432.34

[5] The AD and DD stand for the actual arrival date and return date in allocated travel package

Further details of market game could be found at the TAC pages.[6] There is a multiple auction protocols problem because flight tickets, hotel reservations and entertainment tickets are traded under various auction protocols. There is a bundle completion problem because all goods are interacting. Flight tickets and hotel reservations are complementary because they are essential components for a feasible travel package. Entertainment tickets at the same day are substitutable goods because reward is counted once only. There is allocation problem because each TAC agent serves eight clients whose preferences are private. In TAC, each agent needs to compete with seven unknown opponents. Therefore, a TAC agent faces a game playing problem because opponents' strategies are unknown.

1.3 Thesis Contribution and Organization

This thesis studies how an online bidding agent can effectively serve multiple clients, where bidding goods under various auction protocols and bundling goods items to profitable bundles are necessities. The problem we study in this thesis involves features of all multiple auction protocols problem, bundle completion problem, allocation problem and game playing problem. In this thesis, we proposed a generic agent model using a divide-and-conquer approach to tackle the whole problem. The divided sub-problems then can be handled by a set of mechanisms proposed by us. This thesis also studied the agent's interactions in binary decision case. The experimental results show that the minority has an advantage in a market.

---

6 http://tac.eecs.umich.edu (TAC '01) and http://www.sics.se/tac (TAC '02)

The remainder of the thesis is organized as follow. Chapter 2 discusses previous research in related areas. Chapter 3 describes a theoretical model for agents in online auctions. Chapter 4 introduces our agent architecture and explains the mechanisms in each component. Chapter 5 presents the TAC '02 competition results and evaluates our design with empirical data under controlled environment. Chapter 6 concludes the thesis and possible explores future research.

Chapter 2 – Related Work

2.1 Traditional auction theory

The study of autonomous bidding agent was few until online auctions have become popular. Although there were a number of literatures in the traditional auction theory (Klemperer, 1999), they focused on auctions that trade only one unit of goods with a preset auction protocol. Past traditional auctions were held in physical locations and required bidders to participate in person. As a result of geographical and time limitations, bidders did not participate in multiple auctions simultaneously. There was no *bundle completion problem* or *multiple auction protocols problem* (see Chapter 1). Online auctions revolutionize the auction style because they allow bidders to participate in multiple auctions running simultaneously.

Some assumptions in traditional auction theory do not hold in online auctions (Bapna et al., 2001). First, the risk attributes of a participant cannot be assumed as neutral. Second, a participant's valuation is not as simple as an independent random variable from a given distribution. The effect of external auctions should be considered, especially when goods are interacting (complementary or substitutable). Thus, bidding strategies in traditional auction theory are not fully applicable to online auctions.

2.2 Technologies related to online auctions

Auction sites sometimes provide simple bidding agents for their clients. For example, the Proxy Bidding in eBay.com is a simple bidding agent which keeps increasing the bid just over the current price. Then, it stops increasing the bid once the current price is over the customer's pre-defined valuation. Alternatively, another agent may keep waiting and then bid over the current price just before an auction closes. Those simple bidding agents work only for single item, for single auction protocol and within their auction sites. The key advantage is their *autonomy* in bidding and monitoring auctions. In other words, simple bidding agents cannot be used for multiple auction protocols problem or bundle completion problem.

To assist customers bidding in online auctions, some technologies were developed in recent years. The *auction search engines* are designed for collecting and comparing prices from online product or service providers. In simultaneous auctions, price monitoring is time-consuming. Given customer's target goods, auction search engines return a list of open auctions with their current prices, remaining time and auction protocols[7]. Customers can compare prices easily and save time from finding auctions of desired goods. Yet, auction search engines do not assist customers in bidding and allocating goods. The customers have to analyze collected prices and make decisions by themselves; or they can ask another agent to analyze for them. *Pricebots* is another technology specific for selling goods online. Pricebots are autonomous selling agents, which are designed to maximize selling profit using various price-setting methods (Greenwald & Kephart, 1999). Auction search engines and pricebots partially solve the multiple auction protocols problem because they represent price information of

[7] ww.bidxs.com, www.bidfind.com

various auction protocols in a universal format.

2.3 Recent researches on online auctions

Some previous researches concentrated on bidding a single item from simultaneous auctions using various auction protocols (Anthony et al., 2001; Ito et al., 2000). As those researches concentrated on bidding a single item, there is no complementary relationship among various goods. Those researches solved only the multiple auction protocols problem, but not the bundle completion problem and allocation problem.

Recently, there have been an increasing number of researches about bidding and bundling interacting goods from simultaneous auctions using various auction protocols. TAC is a typical example. Over past TAC tournaments, the organizations[8] were successful to motivate researchers working on a common problem. A lot of studies done by the researchers were published. One example is the fantastic "*priceline*" structure which was investigated by RoxyBot team in TAC '00 (Greenwald and Boyan, 2001a).

[8] TAC '00 and TAC '01 were organized by University of Michigan; while TAC '02 was organized by SICS.

The priceline data structure is the core part of Roxybot design. The major advantage of priceline is its flexibility. The priceline represents marginal costs[9] of items in a universal format for both single-sided and double-sided auctions, and both limited and unlimited supply/demand. The marginal costs are represented as real numbers. The priceline is a very useful structure for the problem we study in this thesis. First, it solves multiple auction protocols problem because it uses a universal representation for all auction protocols. Second, it reduces the complexity of bundle completion problem and allocation problem because an agent only need to reason under a set of real numbers. On the other hand, the priceline works with known clearing prices only. An additional prediction mechanism is needed when clearing prices are unknown.

Another example is the study of efficient allocation mechanisms for multiple clients. The top two agents in TAC '00 investigated how to adopt heuristic search and linear programming (LP) to resolve the resources allocation problem (Greenwald & Boyan, 2001b; Stone at el., 2001). Those two methods have been widely used and further improved by researchers in later TAC tournaments[10]. The performances of those allocation strategies are both heavily depends on accurate price prediction.

Although there is no dominant strategy in TAC problem; meaningful works have been done by researchers to tackle multiple auction protocols problem, bundle completion problem and allocation problem. The participants in TAC '00 aimed at developing an efficient algorithm to resolve the NP-complete[11] allocation problem. In TAC '01, the main focus was switched to cost prediction because the game rules changed. The

[9] The marginal cost of an item refers the amount we will pay for using that item in an allocation.
[10] Examples included ATTac, TACSMAN, PainInNEC, Southampton (TAC '01), PackaTAC and Walverine (TAC '02).
[11] NP-complete problem cannot be solved in a polynomial time.

auctions for hotel reservations became uncertain because they close in random order and their ASK/BID quotes were updated only once per minute. In TAC '02, there was no specialized focusing area. The research area was extended to include machine learning, multi-agent system, game playing, observable Markov decision process and much more. The previous TAC reports summarized and compared agents' strategies in TAC '00 (Stone & Greenwald, 2000), TAC '01 (Wellman, Greenwald, Stone & Wurman, 2002) and TAC '02 (Greenwald, 2002).

2.3.1 Priceline (proposed by Amy Greenwald)

For each goods g, there is one priceline $\vec{p}_g$. The priceline $\vec{p}_g$ is a vector $< p_{g1}, p_{g2}, p_{g3}, \ldots >$ where p_{gn} is marginal cost of the nth item. First, we assume all held items of goods g_1 are not allowed for resale. Suppose an agent currently holds 3 items of goods g_1 and it wants 2 more items. Based on a given prediction mechanism, the clearing price for two additional items are predicted to be 10 and 20. In this case, the priceline for g_1 is given by $\vec{p}_{g_1}$=<0, 0, 0, 10, 20, ∞, ∞, ∞...>. The first three entries are zero because they are *sunk cost*. Sunk cost refers to the cost we already paid for non-reusable goods. The first three items can be allocated without any cost. For the 4th and 5th items, their entries equal the predicted prices because the agent needs to acquire them from others. The subsequent infinity entries indicate that those items are not useful for the agent.

Now we relax the constraint in resale such that an agent is allowed to resell all held items. Suppose other agents are willing to buy two items of g_1 with prices 2 and 5. Then, the priceline of g_1 become $\vec{p}_{g_1}$=<0, **2, 5**, 10, 20, ∞, ∞, ∞...>. It is because the 2nd and 3rd items have opportunity costs now. Opportunity cost of an item refers to the value of that item in another highest-value alternative. If an agent does not allocate those items, it can gain extra utility by reselling them to other agents.

2.3.2 ATTac: Integer Linear Programming (ILP)

We summarize the ILP employed by ATTac in TAC '00. Here we describe the original version. There were other modified versions in later TAC tournaments.

Definitions

c:	A client symbol, from 1 to 8
f:	A feasible travel package symbol, with ● AD(f): arrival day, from 1 to 4 ● DD(f): departure day, from 2 to 5 ● H(f): the hotel, either TT or SS
e:	An entertainment ticket, with ● T(e): type of event, either AW or AP or MU ● D(e): day of the event, from 1 to 4
r:	A resource, either IN or OUT or TT or SS
P(c,f):	An indicator whether client c is allocated with feasible travel package f, altogether 160 variables
E(c,e):	An indicator whether client c is allocated with entertainment ticket e, altogether 96 variables
$B_r(d)$:	The target buying quantity of resource r on day d
$o_r(d)$:	The current holding quantity of resources r
$p_r(d)$:	The current price (asked) of resources r
$u_P(c,f)$:	The utility gained by allocating client c with feasible travel package f
$u_E(c,e)$:	The utility gained by allocating client c with entertainment ticket e

Objective function

$$\sum_{c,f} u_P(c,f)P(c,f) + \sum_{c,e} u_E(c,e)E(c,e) - \sum_{d\in\{2,3,4,5\}} p_{OUT}(d)B_{OUT}(d) - \sum_{d\in\{2,3,4,5\}, r\in\{IN,TT,SS\}} p_r(d)B_r(d)$$

The first term indicates the total utility gained from allocated feasible travel packages. The second term indicates the total utility gained from allocated entertainment tickets. The third term indicates the cost for acquiring additional outbound tickets. The fourth term indicates the cost for acquiring additional inbound tickets and hotel reservations.

Constraints

1. $\forall c, \sum_c P(c,f) \le 1$, no more than one travel package should be allocated to one client.
2. $\forall d \in \{1,2,3,4\}, \sum_c \sum_{f|AD(f)=d} P(c,f) \le o_{IN}(d) + B_{IN}(d)$, the number of inbound flight tickets should be sufficient for the allocation.
3. $\forall d \in \{2,3,4,5\}, \sum_c \sum_{f|DD(f)=d} P(c,f) \le o_{OUT}(d) + B_{OUT}(d)$, the number of outbound flight tickets should be sufficient for the allocation.
4. $\forall d \in \{1,2,3,4\}\ and\ h \in \{SS,TT\}, \sum_c \sum_{f|H(f)=h\ AND\ AD(f)\le d<DD(f)} P(c,f) \le o_h(d) + B_h(d)$, the number of hotel reservations should be sufficient for the allocation.
5. $\forall e, \sum_c E(c,e) \le o_E(e)$, the entertainment tickets allocated should be held by the agent.
6. $\forall c \forall e, \sum_{AD(f)\le D(e)<DD(f)} P(c,f) \ge E(c,e)$, entertainment tickets should be within the travel package tour.
7. $\forall c\ and\ d \in \{1,2,3,4\}, \sum_{e|D(e)=d} E(c,e) \le 1$, no more than one entertainment ticket should be allocated to a client at the same day
8. $\forall c\ and\ t \in \{AP, AW, MU\}, \sum_{e|T(e)=t} E(c,e) \le 1$, duplicate tickets of the same entertainment event should not be allocated to a client
9. The variables should be integers.

There are altogether 188 constraints. The ATTac could resolve the optimal allocation within 1 second using a software package "LPsolve" with a 650MHz Pentium machine.

2.3.3 RoxyBot: Beam search

The Beam search is an approximate approach. The search processes from level to level.[12] In each level, scores of all nodes are calculated based on a given heuristic function *f(x)*. Only the best *N* nodes are selected to expand in each level. The parameter *N* is called "*beam width*". The whole process is without backtracking. The Roxybot divides a search into two stages: 1) travel package feasibility and 2) fun bonus. In each stage, the feasible travel packages (or entertainment tickets) are allocated to clients one by one. The search depth[13] of each stage is 8. Therefore, the total depth of two stages is 16 and the number of nodes is bounded by 16*N*.

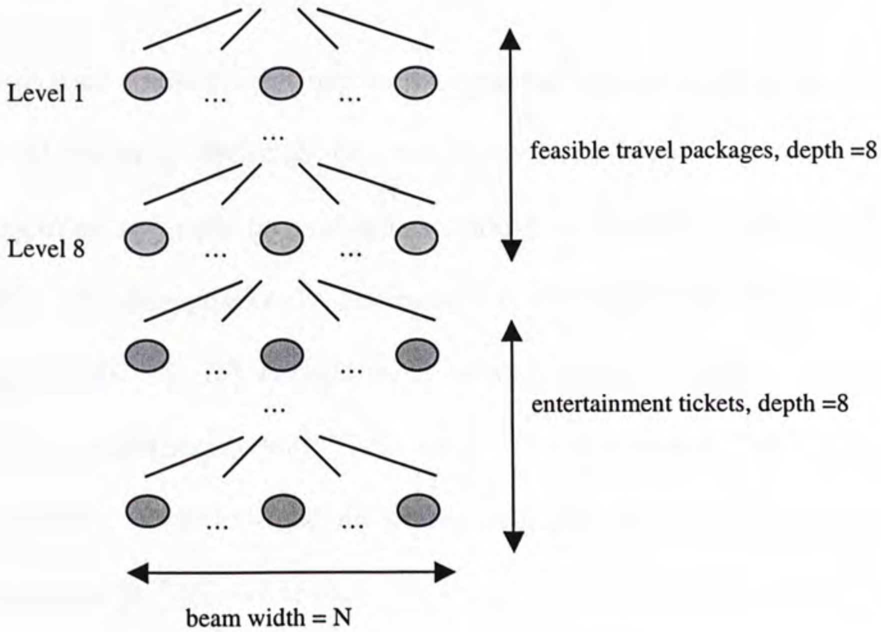


Figure 3: The Roxybot's beam search tree (TAC '00)

---

[12] A level refers to the order of generation in a search tree. The children of the starting node are in level one. The children generated by the level one node are in level two.

[13] Search depth refers to the number of levels.

Chapter 3 – Theoretical model for agents in online auctions

This chapter describes a theoretical model for agents in online auctions. First, we introduce a high-level planning for trading agents. Then, we formulate the bundle completion problem for multiple clients under various auction protocols into a mathematical model (as well as TAC problem).

3.1 High-level planning

Although each trading agent has its own mechanism; most of them adopt a general high-level planning (Wellman, Greenwald, Stone & Wurman, 2002). The high-level planning of an agent can be modeled as working cycles with different cycle times. For example, "SouthamptonTAC" reviewed its allocation adaptively for each game instance in TAC '01. It had cycle times between 6 and 30 seconds, based on the time used for communicating with TAC server (He & Jennings, 2002). Another agent, "livingagents", never changed its strategy and bid all desired items only once per game instance in TAC '01 (Fritschi & Dorer, 2002). In another view, "livingagents" took the whole game instance as one cycle and performed all tasks within that cycle. This high level planning is able to model both adaptive and non-adaptive trading agents. As the model is general, we believe it is also applicable to other market scenarios. The overview of the high-level planning is given in the table below.

Table 3: An agent's high-level planning - cycle

High-level planning
1 Gather market information from online auctions
2 Estimate marginal cost of each single item
3 Estimate the most profitable target, T
4 Generate a target list of items based on T
5 Bid the items in target list using different bidding strategies

In TAC '02, our developed agent "CUHK" adopted working cycles of one minute because the ASK/BID quotes of hotel rooms were updated once per minute. The uncertainty of hotel rooms is the highest among all goods in TAC scenario due to random auction closing times, in addition to fluctuating. To be an adaptive trading agent in TAC, the planning should be reviewed at least once per minute. The CUHK agent decided to review its planning only once per minute to avoid changing allocation too frequent and becoming over-reactive.

3.2 Mathematical model

Greenwald and Boyan (Greenwald and Boyan, 2001b) formulated a model that deals with static prices:

G: Set of all goods, $G = \{1,2,...,|G|\}$

$\vec{q}$: A bundle, which is in form of vector $<q_1, q_2, q_3, q_4 \ldots q_{|G|}>$ where q_g represents the quantity of goods g inside the bundle

Q: Set of all bundles

$u(\vec{q})$: Utility function of bundle $\vec{q}$

$u: Q \rightarrow R+$

Util(S): The total utility of a subset of bundles $S \subseteq Q$, i.e. $Util(S) = \sum_{\vec{q} \in S} u(\vec{q})$

Used(S,g): The quantity of goods g in subset $S \subseteq Q$, i.e. $Used(S,g) = \sum_{\vec{q} \in S} q_g$

$\vec{p}_g$: The priceline[14] of goods g, which is a vector $<p_{g1}, p_{g2}, p_{g3}, \ldots>$ where p_{gn} indicates the marginal cost of n^{th} item of goods g

P: A priceline profile, $P = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_{|G|}\}$

$\mathcal{P}$: Set of all priceline profiles

Cost(S): The total expenditure, i.e. $Cost(S) = \sum_{g \in G} \sum_{n=1}^{Used(S,g)} p_{gn}$

The bundle completion problem, *Completion(P,Q,u)*, is given by:

$$S^* = \arg\max_{S \subseteq Q}(Util(S) - Cost(S))$$

In this model, a universal utility function is defined for all bundles. It implies that there is a public value for all bundles. In many market scenarios (like TAC), each client has his/her private values over bundles. Hence, we revise the above model to incorporate 1) multiple clients with private values over bundles and 2) allocation goods items to those clients.

S: An allocation, which is a sequence of bundles $<\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_{|C|}>$ where $\vec{q}_c$ is the bundle allocated to client c.

$u_c(\vec{q})$: Utility function of client c, u_c: $Q \rightarrow N+$

$\mathcal{U}$: A utility profile for all clients, $\mathcal{U} = \{u_1(\vec{q}), u_2(\vec{q}) \ldots u_{|C|}(\vec{q})\}$

Util(S): The total utility of an allocation S, i.e. $Util(S)) = \sum_{i=c}^{|C|} u_c(\vec{q}_c)$

Cost(S, P): The total expenditure, i.e. $Cost(S,P) = \sum_{g \in G} \sum_{n=1}^{Used(S,g)} p_{gn}$

[14] The priceline described above already combined both buying and selling sides. We have skipped the detail in the original paper about how to combine both sides.

In the model above, the bundle completion problem and allocation problem become a function of static predicted marginal costs (see Chapter 1). The function ignores the effects of actions performed by an agent (i.e. bids). Prices are dynamic and keep changing even an agent does nothing. Furthermore, a bid submitted by an agent can affect both agent's internal state and external market. First, a bid can possibly change agent's holding, which is represented as a sunk cost or opportunity cost (see Definition) ~~in the priceline~~. Second, a bid submitted by an agent influences the market. For example, the bid can possibly induce competitors to increment their bids indirectly. Thus, a better model should consider how an agent's bids affect its holding and market in a long term.

<u>Additional Definitions</u>

b_i:	A bid (or ask) submitted at cycle i, which is a function $\tau_b : \mathcal{P} \rightarrow \mathcal{P}$
B:	Set of all bids, including no new bid
P_i:	A priceline profile at cycle i, $P_i \in \mathcal{P}$
$\mathcal{P}_F$:	Set of all final priceline profiles (i.e. all markets closed), $\mathcal{P}_F \subseteq \mathcal{P}$

Suppose all competitors' strategies are known, the effect of a bid on the priceline can be determined.

$$(P_0) \xrightarrow{b_0} (P_1) \xrightarrow{b_1} \cdots \xrightarrow{b_n} (P_F)$$

Figure 4: Effect of a bid on the priceline

$P_F \subseteq \mathcal{P}_F$ represents a final priceline profile after all markets closed. For a closed market, entries in the priceline are either 0 or ∞ (sunk cost or unavailable). Define $\vec{b}$ as a bid sequence $<b_0, b_1, b_2, \ldots b_n>$ which represents bids submitted by an agent. At cycle i, an agent performs different actions (i.e. bids) will result in various final priceline profiles.

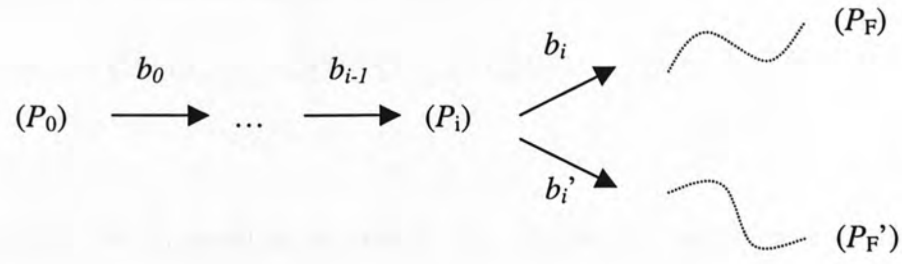


Figure 5: Different bids result in various outcomes

The agent aims at maximizing the final profit. At cycle i, the bundle completion problem, $Completion(P,Q,\mathcal{U})$, is given by:

$$\left(S_i^*, P_{F_i}^*\right) = \arg \max_{(S,P_F)\in Q^{|C|}\times\mathcal{P}_F} (Util(S) - Cost(S,P_F))$$

And, an optimal bid sequence at cycle i is given by:

$$\vec{b}_i^* =< b_{i_1}, b_{i_2}, \ldots b_{i_k} > \quad where \quad b_{i_1}, b_{i_2}, \ldots b_{i_k} \in B \; and \; \tau_{b_{i_k}}\left(\tau_{b_{i_{k-1}}}\left(\ldots\tau_{b_{i_1}}(P_i)\ldots\right)\right) = P_{F_i}^*$$

The optimal bid sequence is not unique because various bid sequences can lead to a single result. Consider an aggressive TAC agent bids all hotel reservations at a high bidding price while others are non-competitive agents. Once its bidding price is higher than the clearing price, the aggressive agent wins hotel reservations it wants.

If an agent submits the bid b_{i_1} in cycle i, S_{i+1}^* will be equal to S_i^*. Otherwise, S_{i+1}^* will become less profitable than S_i^*. The reason is the best allocation S_i^* already becomes unobtainable. For example, an agent should acquire four items of goods g to obtain the best allocation. However, it won three items only because of a mistaken bidding strategy. As the best allocation is already unobtainable, a new (but less profitable) allocation should be formulated for replacement. The profitability is decreasing while allocation keeps changing continuously.

$$Util(S_i^*) - Cost(S_i^*, P_{F_i}*) \geq Util(S_{i+1}^*) - Cost(S_{i+1}^*, P_{F_{i+1}}*)$$

It is hopeless for an agent to determine the optimal bid sequence $\vec{b}_{i_i}^*$ since other agent's strategies are unknown. At cycle i, a trading agent can first formulate a target T_i whose profit approximates that of S_i^*, i.e. $Util(T_i) - Cost(T_i, P_{F_i}*) \cong Util(S_i^*) - Cost(S_i^*, P_{F_i}*)$. Afterwards, the agent can follows a bid sequence that 1) bids items of goods in target T_i and 2) guides S_j^* ($j > i$) towards T_i.

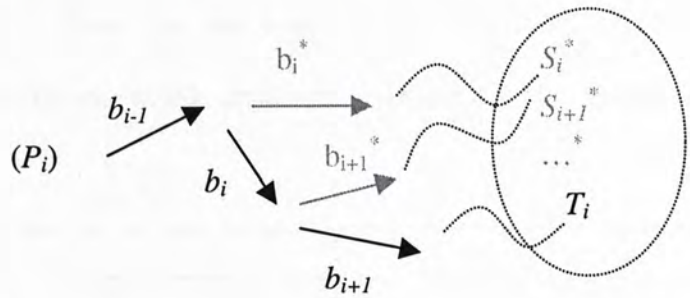


Figure 6: A bid sequence guides the best allocation to our formulated target

The idea is shown in graphical diagram above. Initially target T_i is already close to the best allocation S_i^*. The bid sequence continuously guides S_j^* ($j > i$) towards our target T_i. Although we are not able to determine S_i^*, a trading agent has the ability to guide S_i^* to its desired direction.

Chapter 4 – Agent Architecture and Mechanisms

This chapter provides all the detail of our proposed agent model. First, we introduce the agent architecture which adopts a divide-and-conquer approach. Second, we explain how our proposed mechanisms can handle the divided sub-tasks.

4.1 Architecture

Our agent model is composed of several components. The 3 key components are Cost Estimator (CE), Allocation and Acquisition Solver (AAS) and the Bidders. The figure below represents the whole architecture of our CUHK agent, which participated in TAC '02.

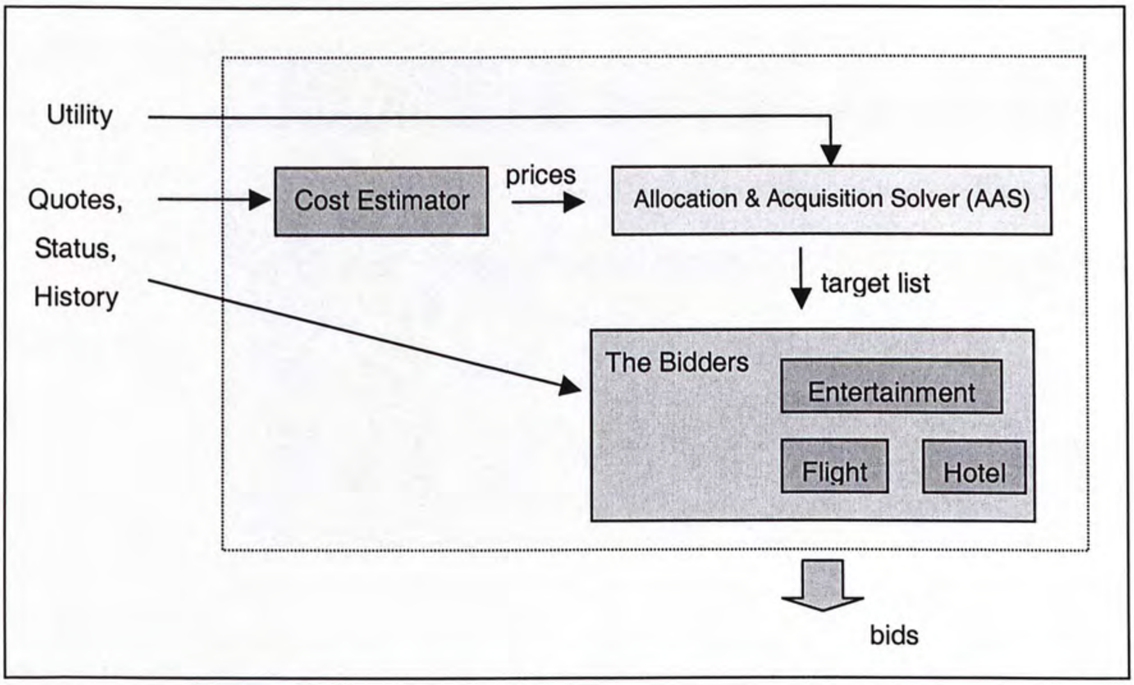


Figure 7: Architecture of our implemented agent, "CUHK"

The first component is Cost Estimator (CE). The main responsibility of CE is to predict marginal costs (see Definition) of all goods items and feed them into AAS. CE uses a data structure called "priceline" to represent estimated marginal costs in a common format. Different prediction methods are employed for various auction protocols. CE considers both sunk cost and opportunity cost because items can be held, possibly won[15] or to be acquired at any time. Another key responsibility of CE is to quantify market information such as demand, supply and auction statuses. The output of CE is a set of estimated marginal costs.

The second component is Allocation & Acquisition Solver (AAS). The responsibility of AAS is to decide what goods and how many items to buy (or sell). Given client's preferences and predicted marginal costs, AAS formulates a target allocation which it considers most profitable. As clearing prices of some goods are unknown; the real optimal is not well-defined. Thus, the "most profitable allocation" formulated by AAS is not the real optimal solution but a function of predicted marginal costs provided by CE. The function of marginal costs resolves the bundle completion problem and allocation problem altogether. In other words, AAS formulates a direction for an agent to follow. The output of AAS is a target list describing how many items of goods to buy or sell.

[15] A possibly won item refers to an item which an agent will probably win in near future.

The last component is the Bidders. The responsibility of the Bidders is to trade goods according to the target set by AAS at good price and good time. The Bidders are composed of a set of small autonomous software bidding robots (bid-bots). Although bid-bot has agent properties, we do not call that an "agent" in this thesis to prevent confusion. Each bid-bot trades one type of goods in an auction according to its auction protocol. The bid-bots decide how to trade items of goods in the most profitable way. They may not strictly follow the target set by AAS. The bid-bots can decrease the number of items to bid while prices are increasing dramatically. When it is not a good opportunity, a bid-bot can also wait and do not bid target items immediately. The bidding strategies used for various auction protocols will be covered in more detail in later section.

Table 4: High level planning of our agent model

While auctions are still running ● Update the status and current prices for all auctions ● Estimate marginal costs of all goods items (performed by CE) ● Determine an allocation with most profit obtainable (performed by AAS) ● Convert the allocation into a target list of goods items (performed by AAS) ● Bid the desired items by mechanisms tailor-made for the auction protocols (performed by the Bidders)

The above table shows how tasks are divided for components in high level planning described in last chapter. The detailed mechanisms for Cost Estimator (CE), Allocation and Acquisition Solver (AAS) and the Bidders are explained in following sections.

4.2 Cost Estimator (CE)

CE uses priceline data structure (Greenwald and Boyan, 2001a) to present all marginal costs in a universal format. At any time, an item of goods might be held, possibly held or to be acquired. In addition, various goods can be traded in different auction protocols. An agent requires a set of prediction mechanisms to deal with all possible cases.

4.2.1 Closed auction

Regardless of auction protocol, the marginal cost of an item is either zero or infinity when all auctions trading this type of goods were closed. The zero marginal cost indicates that it is a *sunk cost*. The item is already paid but it is not reusable for another purpose. The infinity marginal cost indicates that items are not available any more in the market. Suppose h is the number of items of goods g won in a closed auction.

For priceline $\vec{p}_g =< p_{g1} \cdots p_{gh} p_{g(h+1)} \cdots >$,

$$p_{gi} = \begin{cases} 0 & if \quad i \le h \\ \infty & otherwise \end{cases}$$

Consider an example in TAC scenario. Suppose auction for hotel reservation TT at day 3 (TT3) was already closed, and the CUHK agent won three reservations. Then, the priceline is given by $\vec{p}_{TT3}$=< *0, 0, 0*, ∞, ∞, ∞ ...>. The leading zeros for the first three items indicate that those items were already held. Since hotel reservations are not allowed for resale in TAC, there are sunk costs. The following infinity indicates that it is impossible to acquire those items any more. In TAC, there is only one auction for each type of hotel reservations. An agent cannot acquire items which auction for that type of goods was already closed.

$$\vec{p}_{TT3} = < \underbrace{0,\ 0,\ 0}_{\text{sunk cost}},\ \infty,\ \infty,\ \ldots\ldots >$$

4.2.2 Open "take-it or leave-it" market

In "take-it or leave-it" market with sufficient supply, an agent can acquire items at the current price instantaneously. There is no certainty in clearing price. The marginal cost of an item to be acquired is simply the current price, ASK. When items are not allowed for resale, the paid price is a sunk cost. The marginal cost of a paid and held item equals to zero. Suppose h is the number of items of goods g acquired in an open "take-it or leave-it" market.

For priceline $\vec{p}_g =< p_{g1} \cdots p_{gh} p_{g(h+1)} \cdots >$, $$p_{gi} = \begin{cases} 0 & \text{if} \quad i \le h \\ ASK & \text{otherwise} \end{cases}$$

Consider an example in TAC scenario. Suppose the CUHK agent holds two inbound flight tickets at day 2 (IN2) and the current ASK is 350. The priceline is given by $\vec{p}_{IN2}$=<0, 0, 350, 350...>. The two held flight tickets are not allowed for resale in TAC. On the other hand, the CUHK agent is able to acquire more flight tickets at 350 per ticket.

4.2.3 Open continuous double auction (CDA)

Continuous Double Auction refers to a market allowing an agent to buy and sell simultaneously. An agent could resell held items of goods to other agents in a double auction. The marginal cost of a held item is equal to the highest bid in market (BID). It is not a sunk cost because an agent can obtain additional profit by reselling that held item at price BID. Thus, there is an *opportunity cost* equals BID.

On the other hand, the marginal cost of an item to be acquired is equal to the lowest selling price in market (ASK). In double auction, an agent can acquire an item at price ASK immediately. We assume that the clearing prices are stable in an efficient double-sided trading market. Therefore, an agent can acquire multiple desired items at price ASK. Suppose h is the number of items of goods g acquired in an open continuous double auction.

For priceline $\vec{p}_g = < p_{g1} \cdots p_{gh} p_{g(h+1)} \cdots >$,

$$p_{gi} = \begin{cases} BID & if \quad i \leq h \\ ASK & if \quad i > h \end{cases}$$

Consider an example in TAC scenario. Suppose the auction for entertainment ticket AW at day 1 (AW1) has ASK of 50 and BID of 30. The CUHK agent currently holds 3 AW1 tickets. The priceline is given by $\vec{p}_{AW1}$=<30, *30, 30, 50, 50, 50 ....*>. Entertainment tickets are allowed for resale in TAC. The marginal costs of three held entertainment tickets are opportunity costs equals 30. If the tickets are not allocated to clients, an agent can obtain additional utility by reselling them at price 30. The marginal costs of entertainment tickets to be acquired equal 50. We assume those tickets can be acquired from other agents at price ASK.

$$\vec{p}_{AW1} = < \underbrace{30, 30, 30}_{\text{opportunity cost}}, 50, 50, \ldots\ldots >$$

The demand and supply are dynamic in double auction because it depends on participating agents. When a market has limited demand, an agent may not easily resell some held items to other agents. The situation is similar to a market in which resale is not allowed. The marginal costs of non-resalable items would become zero. (i.e. BID = 0) When a market has limited supply, an agent may not easily to acquire some desired items. The situation is similar to a closed auction where items are unavailable. The marginal costs of unavailable items would equal their highest obtainable profit. (i.e. ASK = highest obtainable profit)

Consider another example in TAC scenario. Suppose now no agent sells entertainment tickets in the auction for AW1. The CUHK agent considers the marginal cost of AW1 is equal to 200, which is the maximum utility obtainable of an entertainment ticket in TAC. An agent can generate more profit by selling the held tickets at 200 rather than allocating the tickets to clients. It is reasonable to assume other agents are willing to sell their held tickets at price 200.

$$\vec{p}_{AW1} = < 30, 30, 30, \underbrace{200, 200}_{\text{no supply}}, \ldots\ldots >$$

4.2.4 Open multi-unit ascending auction

Multi-unit ascending auction requires agents to offer bids higher than ASK. The transactions clear once at the end of an auction with price equals ASK. The top bids win those items of goods. An agent needs to reason under *uncertainty* since clearing prices are unknown.

One possible prediction method is considering marginal costs of all goods equal to their asking prices, ASK. Suppose ASK of goods g is 100, the priceline is given by $\bar{p}_g = <100, 100, 100 \ldots>$. In competitive markets, clearing prices of goods could be skyrocketing. The ASK is not so useful since it can change dramatically afterward. An agent may then acquire non-profitable items of goods due to the inaccurate predicted marginal costs.

4.4.2.1 Historical clearing prices

In a market with uncertainty, we believe an agent can improve accuracy of prediction by referencing a series of recent historical clearing prices. It is based on an assumption that collective behavior of opponents would not change dramatically. First, historical clearing prices acts as a measurement of opponents' aggressiveness. Although an individual opponent's strategy is unknown, the collective behaviors are observable through the historical clearing prices. Second, the finalized outcome of an agent's strategy is already reflected in recent historical clearing prices. The same strategy of an agent would result in consistent outcomes if collective behaviors of opponents do not change dramatically.

In TAC, we use *median* of clearing prices in 10 most recently played games for representing outcomes of historical auctions. Median is used instead of mean to reduce effects from abrupt clearing prices. Some agents did not submit bids until the first hotel auction close (at the end of 4th minute). It results in a very low ASK during first few minutes. In several game instances, the ASK can reach as low as zero. Referencing historical clearing prices is useful for marginal cost prediction when ASK and BID are not accurate enough.

Most TAC agents have submitted their bids before the first hotel auction closed. Since hotel auction closing order is random, it is impossible for an agent to predict which hotel auction closes first. Consequently, an agent needs to bid all desired hotel reservations before the first hotel auction closed. After the first hotel auction closed, the ASK and BID possibly rise dramatically. At that time, ASK and BID become much useful because they reflect most opponents' valuations on hotel reservations.

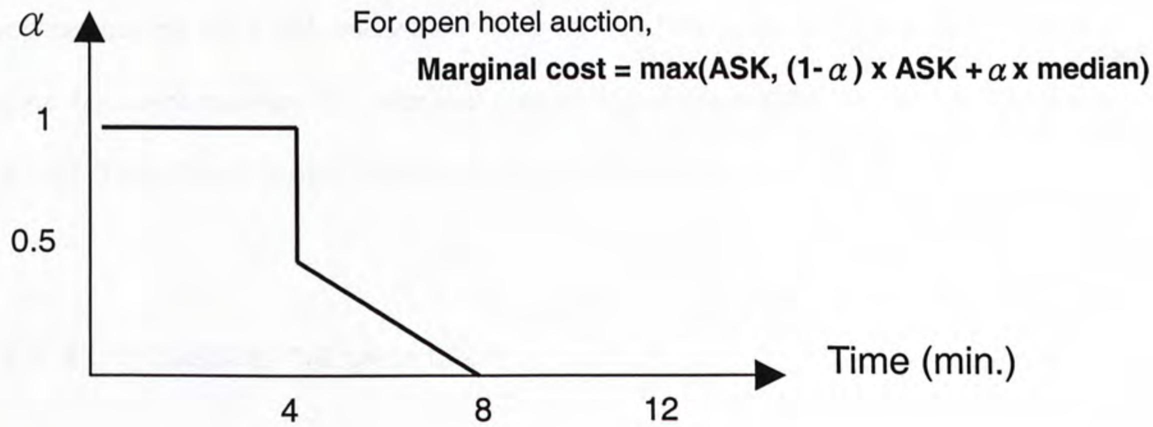


Figure 8: The marginal cost prediction which considers historical clearing prices

The CUHK agent uses a function of *ASK* and *median* to predict marginal costs in the middle of a game instance. A parameter α is used to represent the relative importance between ASK and median. Before the first hotel auction closes, the CUHK agent considers median as the marginal cost because ASK provides no useful information.

For priceline $\vec{p}_g =< p_{g1} ... p_{gi} ... >$,

$$p_{gi} = MC \text{ and } MC = \max(ASK, 1 - \alpha(t) \times ASK + \alpha(t) \times Median)$$

where *Median* is median of clearing prices of goods *g* in last 10 played games
$\alpha(t)$ is a user-defined weight function of time *t*

After the first hotel auction closes, the importance of median sharply drops to 50% in order to adapt the current market. After the eighth minute, the market is believed to be settled. Therefore, the CUHK agent considers ASK as the marginal cost of goods directly.

Suppose auction for hotel reservations SS2 has median of 80 and ASK of 20 at the end of the fourth minute. The marginal cost of SS2 is predicted to be 80 x 0.5 + 20 x 0.5 = 50. Thus, the priceline is given by $\vec{p}_{SS2}$=*<50, 50, 50,...>*.

4.4.2.2 Increasing marginal costs

When clearing prices of goods are uncertain, an agent's allocation should not heavily depend on a few types of goods. Otherwise, the penalty of an inaccurate prediction will be high. We propose a set of rules an agent should follow for predicting marginal cost of goods with unknown clearing prices.

Table 5: Rules for increasing marginal cost of uncertain goods

Rules	
1. $MC_{g,j} \geqq MC_{g,i}$	for $j \geqq i \geqq 1$
2. *If* $MC_{g,1} \geqq MC_{h,1}$, *then* $MC_{g,i} \geqq MC_{h,i}$	for $i \geqq 1$
where $MC_{g,i}$ refers to the marginal cost for i^{th} item of goods g	

The first rule implies that the marginal cost of goods g should be increasing. The second rule implies that the marginal cost of i^{th} item should be higher if that type of goods is predicted to be more expensive. The first rule causes an agent to bid fewer items. The second rule causes an agent to bid fewer items of goods which are predicted to be more expensive. The first and second rules altogether prevent an agent from depending on a few kinds of goods.

For priceline $\vec{p}_g = \langle p_{g1} \ldots p_{gi} \ldots \rangle$,

$$p_{gi} = MC_i \quad \text{and} \quad \begin{cases} MC_1 = MC \\ MC_i = f(MC, i) \end{cases}$$

where $f(x, n)$ is a user-defined function that satisfies the 2 rules above

Consider an example in TAC scenario. Define MC_{SS4} as the marginal cost of hotel reservation SS at day 4 based on the methods we described earlier (i.e. MC_{SS4}=50). Our CUHK agent uses MC_{SS4} as a *baseline* to calculate all marginal costs of goods SS4. The calculation is very straightforward. The marginal cost of i^{th} hotel reservation, $MC_{SS4,i}$, is equal to $MC_{SS4} \times i$. The priceline is given by $\vec{p}_{SS4}$=*<50, 100, 150, 200…>*.

First, the CUHK agent targets on fewer hotel reservations. It is understandable as the marginal costs of hotel reservations are increasing; bidding too many hotel reservations is very costly. Second, the CUHK agent targets on fewer hotel reservations which are predicted to be expensive. It is because the second reservation of an expensive hotel is already very expensive. When more hotel reservations are needed for an allocation, the CUHK agent prefers reservations with lower prices. Third, the CUHK agent does not heavily depend on one type of hotel reservations. Suppose MC_{SS3} is lower than MC_{TT3}. The CUHK prefers SS hotel for the first few reservations. Once the agent's allocation already contains a few hotel reservations of SS3, the marginal cost of an additional SS3 hotel reservation becomes higher than that of an additional TT3 hotel reservation.

$$\bar{p}_{SS2} = < 50, 100, 150, 200, \ldots. >$$

Marginal cost are increasing

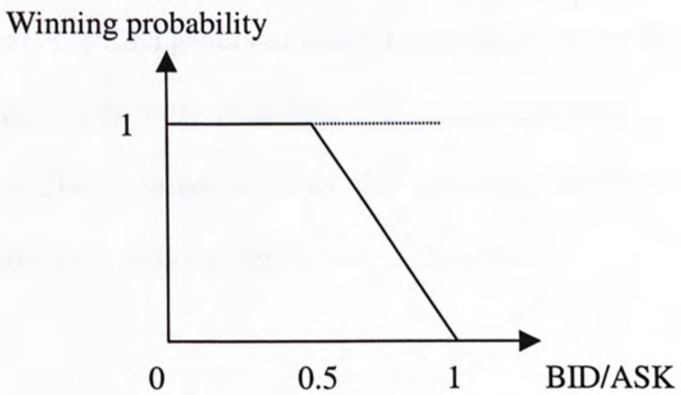
4.4.2.3 Bid winning probability

In a multi-unit ascending auction, usually an agent is not allowed to withdraw its bids. In other words, an agent makes a commitment to acquire its won items of goods at clearing price for each bid it made. An agent *possibly wins* items when their bids are over the current price ASK. The commitments are removed only when an agent no longer possibly wins the associated items, i.e. other agents place bids on top of their bids.

Under particular conditions, several commitments can be considered as fixed and not removable again. First, the commitments are unlikely removable when too many of items are can possibly be won. Second, it is unlikely that other agents will make new bids to compete with us when ASK become already so high. Since an agent should acquire those items later anyway, the marginal costs become sunk costs.

We define *bid winning probability* as how likely an agent wins the bid items. In TAC, our CUHK agent estimates the bid winning probability based on the BID to ASK ratio. When BID is close to ASK, we predict other agents will place bids over our bids soon. Thus, commitments of bids are likely to be removable. When ASK is far higher than BID, it is likely that there is no further demand for hotel reservations. Thus, commitments of bids are likely fixed.

Figure 9: The bid winning probability calculated from the BID to ASK ratio



For priceline $\vec{p}_g =< p_{g1} \ldots p_{gH}\, p_{g(H+1)} \ldots >$,

$$p_{gi} = \begin{cases} 0 & if \quad i \le H \\ MC_{(i-H)} & otherwise \end{cases}$$

where H is the number of items of goods g considered to be surely won

Consider an example in TAC scenario. Suppose our CUHK agent possibly wins 4 hotel reservations SS at day 4 (SS4) and the calculated bid winning probability is 0.5. Then, we consider the agent surely wins two hotel reservations (4 x 0.5). Suppose marginal cost of SS4 is predicted to be 50. The priceline is given by $\vec{p}_{SS4}$=***<0, 0,*** *50, 100, 150, 200 …>*.

4.3 Allocation and Acquisition Solver (AAS)

AAS is the most important component in our agent model. It proposes profitable bundles for clients and then generates a target bidding list of items. AAS takes a set of estimated marginal costs from CE as input. Accuracy and speed are the two key issues in AAS design. The accuracy denotes the optimality of AAS output allocation. Generally, a faster AAS provides less accurate allocation.

4.3.1 Un-coordinated VS coordinated aspiration

AAS solves both bundle completion problem and allocation problem by a function of marginal costs. There is no unique approach to design that function. A trading agent could adopt 1) un-coordinated or 2) coordinated aspiration to their clients. Uncoordinated aspiration implies that an agent aspire to maximize the profit of each client. "Livingagents", the top scoring agent in TAC '01, was taking un-coordinated aspiration (Fritschi and Dorer, 2002). It allocated the most profitable travel package for each single client without considering global profit. Coordinated aspiration implies that an agent aspire to maximize global profit of all clients. "ATTac", the runner-up in TAC '01, was taking coordinated aspiration (Stone et al, 2002). It took whole clients' utility and overall expenditure in consideration.

Coordinated aspiration is more reasonable because a trading agent aims at maximizing whole profit. Maximizing local profit for each client may not offer a global optimal to an agent. The coordinated aspiration is also more flexible because an item can be used for many possible allocations. In contrast, coordinated aspiration consumes more computational resources but is not guaranteed to work better than uncoordinated aspiration. A good example is "livingagent" obtaining the highest score in TAC '01. It is because prices of goods are dynamic and unknown to the agents. With coordinated aspiration, a small change in goods price may already affect whole allocation. The worst situation is when an agent keep modifying its allocation. With uncoordinated aspiration, an allocation does not change easily by a small change in prices. As a result, an agent becomes more stable and avoids frequent changes in allocation.

4.3.2 Optimal VS heuristic approach

There are 2 common approaches to find the best allocation: 1) optimal approach and 2) heuristic approach. The optimal approach aspires to find an optimal solution while the heuristic approach aspires to find an approximate solution. The optimal approach provides an optimal solution but it can be time-consuming. Linear programming (LP) and A* search are typical examples of optimal approach (Greenwald and Boyan, 2001b). The heuristic approach is fast but less accurate. Greedy search is one example.

The *optimality* described above assumes predicted marginal costs provided by CE are exactly the clearing prices of goods items. Since there is no perfect cost prediction method for goods with unknown clearing prices, even an optimal approach is not guaranteed to find the best allocation. On the other hand, a heuristic approach already provides a high-quality solution with faster speed.

4.3.3 An greedy approach with coordinated aspiration

In TAC, our CUHK agent adopts a greedy heuristic search to allocate items to clients. The figure below illustrates its allocation strategy.

Table 6: The allocation strategy of our CUHK agent

- INPUT
  - A set of estimated marginal costs
- MAIN PROCEDURE
  - Generate a random order of clients
  - Allocate provisional travel packages to clients
  - Calculate global profit
  - Repeat N times
- OUTPUT:
  - An allocation with highest global profit

The greedy search algorithm of the CUHK agent is modified from the Roxybot's Beam Search (Greenwald and Boyan, 2001b). We set the beam width equals one (i.e. best-first search). Given a client order, AAS allocates items client by client. The first client can pick any item to construct a travel package with the highest profit. Then, the second client picks any item from remaining items to construct his/her travel package. This process repeats until all clients have been considered. Notice that the items are not necessary held by a trading agent. Some allocated items can be acquired later.

As greedy search can lead to a suboptimal, the CUHK agent performs the greedy search over N random client orders. The allocation with highest global profit is then selected. The optimality improves when the number of trials increases. Experiments show that Beam Search with width equals one has averaged (or median) accuracy of 99% in TAC scenario (Greenwald and Boyan, 2001b; Stone et al., 2001).

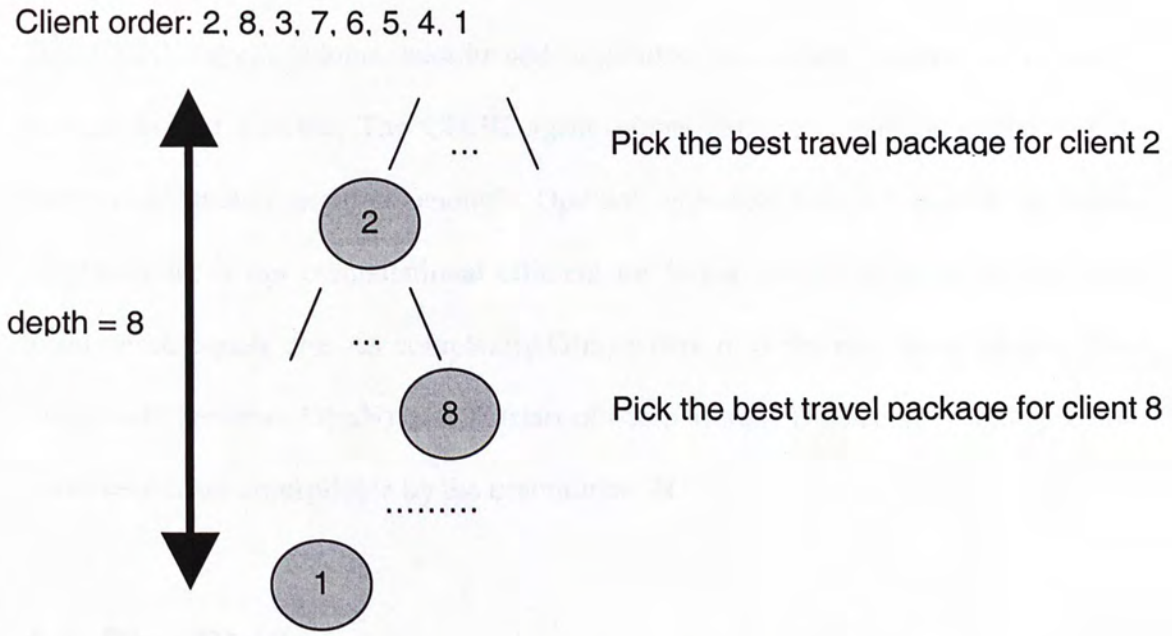


Figure 10: A greedy search with coordinated aspiration to clients

The CUHK agent divides the whole allocation process into 2 stages. In the first stage, only flight tickets and hotel reservations are allocated. It is because they are the essential elements for bundling feasible travel packages. In the second stage, the entertainment tickets are allocated.

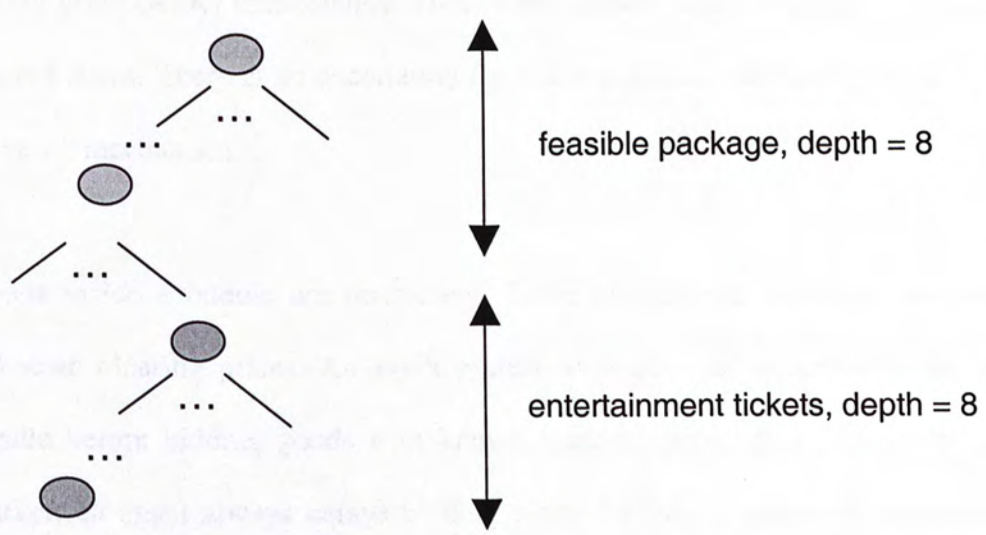


Figure 11: The 2-phase greedy search

The CUHK agent adopts coordinated aspiration to clients because it is more reasonable and flexible. The CUHK agent adopts heuristic approach because it is faster and already accurate enough. Optimal approach like A* search or linear programming is not computational efficient for larger problems. Beam search with beam width equals one has complexity O(m) where m is the number of clients. The complexity becomes O(mN) over N trials of beam search. The accuracy and speed of beam search are controllable by the trial number N.

4.4 The Bidders

The bid-bots, which are inside the Bidders, trade target items of goods set by AAS according to the prevailing auction protocols.

4.4.1 "Take-it or leave-it" market

In a "take-it or leave-it" market, an agent can acquire any quantity of goods by the asking price (ASK) immediately. Thus, a bid equals ASK is sufficient to acquire the desired items. There is no uncertainty for bidding goods in a market using "take-it or leave-it" mechanism.

Goods inside a bundle are interacting. Some of them are traded in auctions with unknown clearing prices. An agent prefers to acquire all uncertain goods inside a bundle before bidding goods with known clearing prices. In a "take-it or leave-it" market, an agent always delays bidding when the asking prices are predicted to be unchanged.

Sometimes, the asking price in a "take-it or leave-it" market may have an increasing trend. Then, an agent faces a dilemma either 1) bid items earlier for lower prices, or 2) bid items later for more flexible planning.

Consider an example in TAC scenario. Flight tickets are traded in markets using "take-it or leave-it" mechanism in TAC. Since TAC '01 tournament, the game rule was changed such that flight ticket price has an increasing trend. On the other hand, flight tickets and hotel reservations are interacting because both of them are essential for travel package feasibility.

To solve the dilemma, the CUHK agent bids flight tickets twice within a TAC game instance: the beginning and the last minute. At the beginning, the prices of flight tickets are expected to be lowest. For that reason, the CUHK agent bids all desired flight tickets in the target list given by AAS. At the last minute, all auctions for hotel reservation were closed. There is no further uncertainty associated with travel package feasibility. Thus, the most profitable travel packages for clients are determined. The, our CUHK agent bid all remains flight tickets for completing those profitable travel packages.

Table 7: The bidding strategy for flight tickets in TAC

1. Initialize • Bid all flight tickets needed for initial allocation *T* with *bid*s set to *ASK* 2. While some hotel auctions are running • Do nothing 3. Finalize • Bid remains flight tickets needed for the best allocation *T* with *bids* set to *ASK*

4.4.2 Multi-unit ascending auction

An agent need to reason under uncertainty when it bids goods in multi-unit ascending auction. The opponents' strategies are unknown but only limited items are available. Depends on agents' demands for goods, clearing prices sometimes can be low but sometimes can be skyrocketed. Worse, items become unavailable after all auctions selling that type of goods are closed. As goods are interacting inside a bundle, an agent fails to complete bundles for clients if it cannot acquire enough goods items. An agent faces both *game playing problem* and *bundle completion problem* in multi-unit ascending auction (see Chapter one).

Table 8: The bidding strategy for hotel reservations in TAC

1. Initialize
   - Bid hotel reservations using **Low price bidding**
2. Until the first hotel auctions close
   - Bid insufficient hotel reservations by incrementing *ASK*
3. While some hotel auctions are running
   - Bid hotel reservations using **Budget bidding**

4.4.2.1 Budget bidding

Suppose a bundle contains some goods traded in multi-unit ascending auctions. In a multi-unit ascending auction, the top bids win the items at the clearing price (i.e. the lowest winning price). We propose an agent's budget for bidding goods inside a bundle should be equal to its utility obtainable from that bundle. If the budget for completing a bundle is more than its utility obtainable, an agent can end up with a loss. If the budget is less than its utility obtainable, an agent has smaller chance in completing the bundle.

The CUHK agent starts budget bidding since one cycle prior to the first hotel auction closes. The bidding prices of hotel reservations are calculated based on the formula below.

For each client c, $Bid_{cg} = \left(u_c(\vec{p}) - \cos t_{flight}\right) / AA$

where

Bid_{cg} is bidding price used by client c for an item of goods g

$u_c(\vec{p})$ is utility of client c over travel package $\vec{p}$

$cost_{flight}$ is total cost for inbound and outbound tickets to be acquired later

AA is the number of remains hotel reservations required for travel package $\vec{p}$

We illustrate it by an example. In TAC, hotel reservations are traded in multi-unit ascending auction. Suppose AAS allocated a feasible travel package $\vec{p}$ to client 3 with utility of 800. The arrival day and return day of $\vec{p}$ are day 1 and day 4 respectively. Hotel SS is selected in $\vec{p}$ such that there is no hotel premium for client 3.

The utility obtainable is 800 if an agent can successfully complete travel package $\vec{p}$ for client 3. Assume inbound and outbound tickets are ready for $\vec{p}$ such that they cost nothing (sunk cost). The SS2 reservation, which auction was already closed, is ready for $\vec{p}$ as well. Hence, budget for bidding two remains hotel reservations, SS1 and SS3, is 800. Our CUHK agent divides the budget into two bids of 400 each.

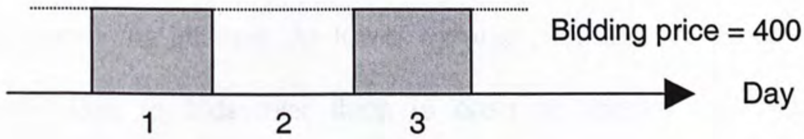


Figure 12: Budget bidding in multi-unit ascending auction

The budget did not include hotel auction SS2 because it is already paid (sunk cost). The budget is re-calculated after each cycle. Suppose auction for SS1 closed first with clearing price 300, the new bidding price for SS3 become 800.

Unlike aggressive agent which always bids at very high price, the CUHK agent employs divisions of utility as bidding prices. It prevents an agent from completing the travel package with a loss. Suppose bidding prices of SS1 and SS3 are set as 500, the profit become negative when sum of their clearing prices is over 800.

4.4.2.2 Low price bidding

Interacting goods inside a bundle can be traded in simultaneous multi-unit ascending auctions. Since clearing prices are unknown, some agents can underestimate market demands of those goods at the beginning. When the agents discover this mistake, most auctions may be already closed. As a result, the agents cannot change their allocation anymore from limited available goods. The agents may then compete for remain goods aggressively, causing the clearing prices become skyrocketing.

We propose that an agent can stabilize market by submitting a series of low bids to multi-unit ascending auction. As lower winning positions are occupied by our bids, competitors need to bids over them in order to acquire their desired items. It stimulates competitors to actively participate. When asking price (ASK) is pushed to a high level earlier, competitors recognize the demand for this type of goods is high. Hence, a few agents switch to acquire other goods to avoid a non-profitable bid war. Effectively, it prevents agents from fighting against each others aggressively. Since the market becomes more efficient and stable, clearing prices are more predictable. Additionally, an agent can win items at low clearing prices when the demand for that type of goods is low. The low price items provide more flexibility to an agent in constructing bundles for clients.

Now we describe how the CUHK agent performs low price bidding in TAC. For each kind of hotel reservation, the CUHK agent submits a series of low bids. The bid of i^{th} hotel reservation is $\lambda \times \left(1-\gamma^{\frac{1}{i}}\right)$ where λ is a controllable parameter defining expensiveness and γ is an accepting threshold. In our implementation, λ is set to be 100 and γ is set to be 0.5. The calculation is based on an inequality $\sqrt[i]{(1-\frac{bid}{\lambda})} \geq \gamma$.

The part $\sqrt[i]{(1-\frac{bid}{\lambda})}$ in inequality computes a *degree of inexpensiveness* (*DI*). A high value of *DI* means that the item is a bargain. If a bid has *DI* over the threshold γ, then an agent accepts that bid. *DI* is discounted exponentially for additional hotel reservations. Thus, the bids for additional reservations are decreasing.

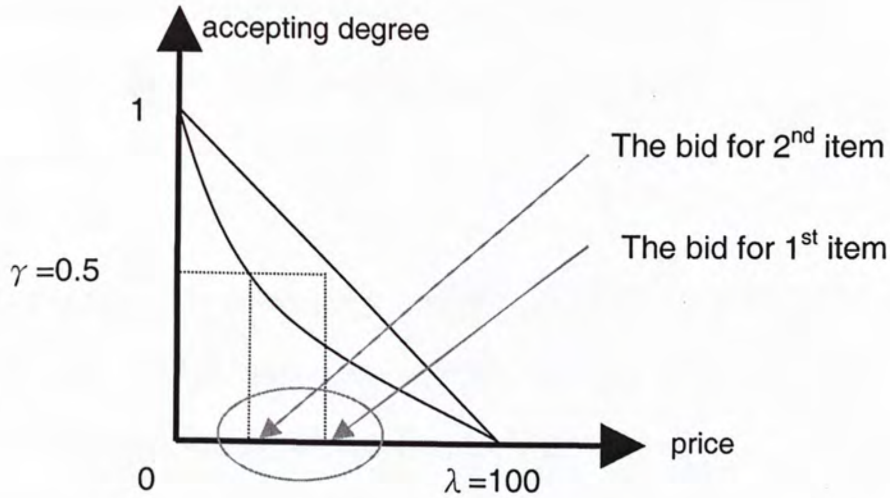


Figure 13: Low price bidding in multi-unit ascending auction

The CUHK agent submits this series of low bids at the beginning of a game instance. Afterwards, the CUHK agent continues to bid insufficient hotel reservations for the target set by AAS. The bids are equals to ASK x ω where ω is a controllable parameter.

4.4.3 Continuous double auction (CDA)

We propose an agent to use FL-strategy to trade goods in CDA. The FL-strategy was first proposed by He and Leung (He & Leung, 2001). The FL-strategy makes use of fuzzy reasoning and fuzzy rules to decide the optimal bid or ask in CDA. [16] Moreover, the fundamental FL-strategy can be enhanced by introducing an adaptive risk attitude.

The original study of FL-strategy focuses on trading single item without considering interacting goods. An agent can behave either buyer or seller, but not both. Besides,

[16] See Definition

the original FL-strategy assumes that an agent has well-defined valuations over all items of goods. These assumptions do not hold when client preferences are over bundles.

In this thesis, we developed a simplified version of FL-strategy and have integrated it into our CUHK agent implementation. In TAC, the CUHK agent bids the entertainment tickets once per cycle (minute) starting from the beginning of a game instance until the end of the instance.

4.4.3.1 Review of fuzzy reasoning mechanism

Consider the following set of fuzzy firing rules from R1 to RN:

Table 9: Fuzzy firing rules

R1:	If x is A_1 and y is B_2, then z is c_1
R2:	If x is A_2 and y is B_2, then z is c_2
	……
RN:	If x is A_N and y is B_N, then z is c_N
	where A_i and B_i are fuzzy sets and c_i is a real number.

Suppose the following facts are true: x *is* x_0 and y *is* y_0. For each rule Ri, the *firing level* α_i is calculated as min $\{(A_i(x_0), B_i(y_0)\}$ where $A_i(x)$ and $B_i(y)$ are membership functions[17].

[17] Membership function returns a degree of membership of one variable in a fuzzy set.

The overall result z_0 equals the "weighted average" of all c_i.

$$z_0 = \sum_{1...N} \alpha_i c_i$$

The calculated result z_0 is not always a valid value in some scenarios. For example, some markets require agents increase their bids by a pre-set incremental value. A fuzzy number representation can be used to resolve this problem. In original FL-strategy, the result z_0 is extended to a triangular fuzzy number $\tilde{z}_0 = (m, \theta, \chi)$ whether m is the center, θ and χ are left and right spreads. The previous result z_0 become the centre of $\tilde{z}_0$ (i.e. $z_0=m$), which has highest degree of membership. Define D as a set of all valid values, which has degree of membership higher than a pre-set threshold π. Then, FL-strategy selects the value with the highest degree of membership inside D.

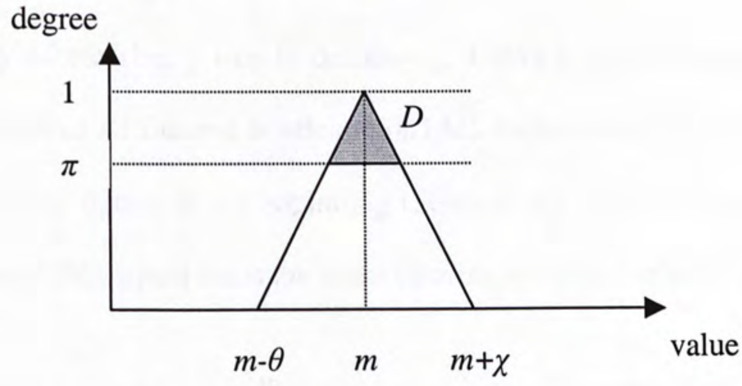


Figure 14: Triangular fuzzy number, $\tilde{z}_0 = (m, \theta, \chi)$

In TAC, there is no price stepping requirement for bid. Hence, the CUHK agent uses the real number z_0 for simplicity.

4.4.3.2 Fuzzy Reasoning in FL-strategy

Our simplified FL-strategy decides the optimal bid and ask based on ASK, BID and a *reference price (PR)*. The principle of using a reference price is straightforward. It is easier for an agent to decide bid and ask based on trading experiences. Unlike original FL-strategy, our simplified version does not consider valuations of items. First, it is hard to assess the values of items while goods are interacting. Second, the target goods items set by AAS are already believed to be profitable.

In the original FL-strategy, the clearing price in a pre-set number of transactions ago is taken as the reference price (*PR*).[18] It is based on an assumption that the supply and demand of market are relatively stable. When a market is very dynamic, it is better to use the latest clearing price as a reference point. Consider TAC as an example. The uncertainty of clearing prices is decreasing during a game instance in TAC. Some competitors, like ATTac and SouthamptonTAC, increase bid (decrease ask) steadily to keep optimistic option at the beginning (Stone et al., 2001; He and Jennings, 2002). Hence, our CUHK agent takes the latest clearing price as a reference price PR.

There are three possible cases[19] for the relation between BID, ASK and PR:

1. PR ≦ BID < ASK
2. BID < ASK ≦ PR
3. BID ≦ PR ≦ ASK

[18] For example, an agent can take the clearing price 3 transactions ago as PR. Assume the latest three clearing prices are {80, 100, 90}, then PR is 80.
[19] BID is always less than ASK. Otherwise, there is a transaction.

Same as original FL-strategy, simple fuzzy rules are used for the first two cases. Here we study the selling side first.

- When PR $\leqq$ BID < ASK,
  IF BID is *much_bigger* than PR
  THEN accept BID
  ELSE ask $\leftarrow$ (ASK – $\beta_{s,1}$)

When BID is higher than PR, it implies that the standing BID is in a relatively high level. When BID is much bigger than PR, an agent should sell goods immediately at that attractive price. Even if BID is not much bigger than PR, an agent can reduce *ask* by a great amount to attract buyers.

- When BID < ASK $\leqq$ PR
  IF ASK is *much_smaller* than PR
  THEN no new asked
  ELSE ask $\leftarrow$ (ASK – $\beta_{s,2}$)

When ASK is smaller than PR, it implies that the standing ASK is in a relatively low level. When ASK is much smaller than PR, a seller should not sell goods at that unfavorite price. Otherwise, it will benefit to other competitors. If ASK is not much smaller than PR, an agent can reduce *ask* by a little bit even that price is not so good. The reason is that the item is still a target item set by AAS.

The relation "*much_bigger*" and "*much_smaller*" above are fuzzy sets. Denote "x is *much_bigger* than y" as fuzzy set A. If membership function $A(x\text{-}y)$ return a degree higher than a pre-set threshold $\gamma_{s,1}$, an agent considers x is much bigger than y (selling side).

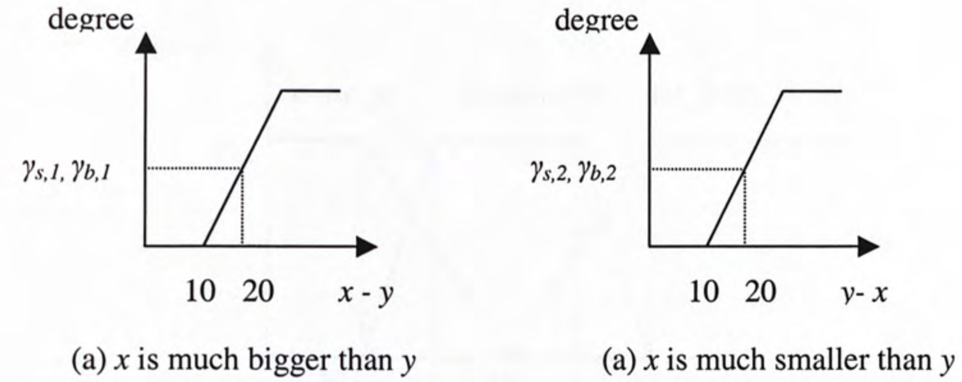


Figure 15: Fuzzy sets used in simple fuzzy rules

The rules for buying side are using the same understanding but in a reverse way.

- When BID < ASK $\leqq$ PR
  IF ASK is *much_smaller* than PR
  THEN accept ASK
  ELSE bid ← (BID + $\beta_{b,1}$)

- When PR $\leqq$ BID < ASK,
  IF BID is *much_bigge*r than PR
  THEN no new bid
  ELSE bid ← (BID + $\beta_{b,2}$)

The last case (BID $\leqq$ PR $\leqq$ ASK) is not as straightforward as previous 2 cases. When PR falls between BID and ASK, it is hard to tell whether the current market situation is a good opportunity. In this case, we employ the fuzzy firing rules and fuzzy reasoning mechanisms introduced above. We define three more fuzzy sets: "*close_to*", "*medium_to*" and "*far_from*".

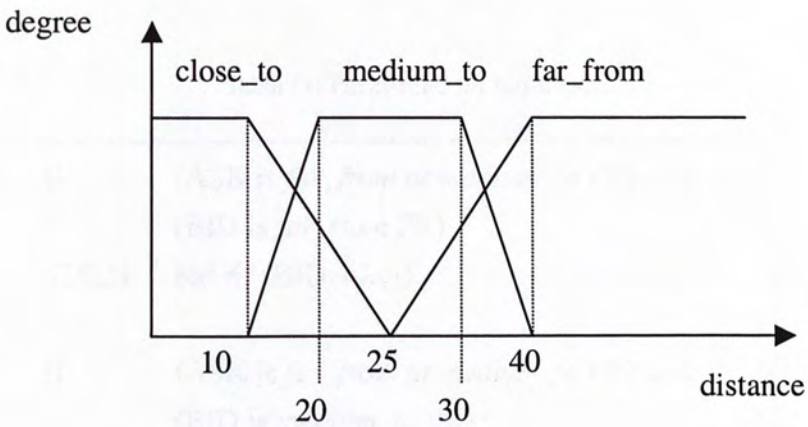


Figure 16: Fuzzy sets used in complicated fuzzy reasoning mechanisms

We study the selling side first:

Table 10: Fuzzy rules for selling side

- IF (BID is *far_from* or *medium_to* PR) and (ASK is *far_from* PR)
  THEN ask ← (ASK − $\lambda_{s,1}$)
- IF (BID is *far_from* or *medium_to* PR) and (ASK is *medium_to* PR)
  THEN ask ← (ASK − $\lambda_{s,2}$)
- IF (BID is *far_from* or *medium_to* PR) and (ASK is *close_to* PR)
  THEN ask ← (ASK − $\lambda_{s,3}$)
- IF BID is *close_to* PR
  THEN ask ← (PR + $\lambda_{s,4}$)

When BID is far from PR, an agent need to reduce *ask* to attract buyers. When ask become closer to PR, less profit is obtainable by the agent. Thus, the agent reduces *ask* by a smaller amount when *ask* become closer to PR.

Similarly, the fuzzy firing rules for buying side are as follow:

Table 11: Fuzzy rules for buyer side:

•	IF	(ASK is *far_from* or *medium_to* PR) and (BID is *far_from* PR)
	THEN	bid ← (BID + $\lambda_{b,1}$)
•	IF	(ASK is *far_from* or *medium_to* PR) and (BID is *medium_to* PR)
	THEN	bid ← (BID + $\lambda_{b,2}$)
•	IF	(ASK is *far_from* or *medium_to* PR) and (BID is *close_to* PR)
	THEN	bid ← (BID + $\lambda_{b,3}$)
•	IF	ASK is *close_to* PR
	THEN	bid ← (PR – $\lambda_{b,4}$)

In our simplified FL-strategy, an agent has a feature of *relaxed acceptable price*. The same feature can also be found in design of SouthamptonTAC (He and Jennings, 2002) in TAC '01. Suppose *ask* is calculated as 100 based on the fuzzy reasoning mechanisms above. An agent accepts BID when BID is already close to the calculated ideal *ask*.

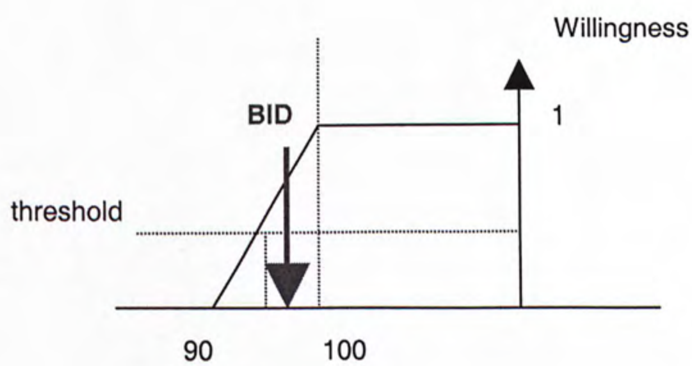


Figure 17: Relaxed acceptable price in FL-strategy

4.4.3.3 Adaptive Risk Attitude

The last feature of FL-strategy is the adaptive risk attitude. Define *RISK* as a risk attribute which is bounded by -1 and +1. In our simplified version, an agent has 2 separate attributes for buying side and selling side, $RISK_b$ and $RISK_s$. The risk attributes are adjusted based on previous transactions. The learning rules are as following[20]:

- IF The selling (buying) transaction *is_active*

  THEN $RISK_i = RISK_i\,(1 + \sigma * is\_active)$
- IF The selling (buying) transaction *is_inactive*

  THEN $RISK_i = RISK_i\,(1 - \sigma * is\_inactive)$

where $i = \{s,b\}$ and σ is the maximum step. Define *transaction rate* (TR) is ratio between total sold quantity and target selling quantity in previous cycle. Suppose target selling quantity was 10 and sold quantity was 4, TR equals 0.4. The relation *is_active* and *is_inactive* are fuzzy sets. If the degree of *is_active*(TR) is higher than a pre-set threshold π_{active}, an agent considers the transactions (in previous cycle) is active.

[20] Same learning rules are applied for $RISK_b$ and $RISK_s$.

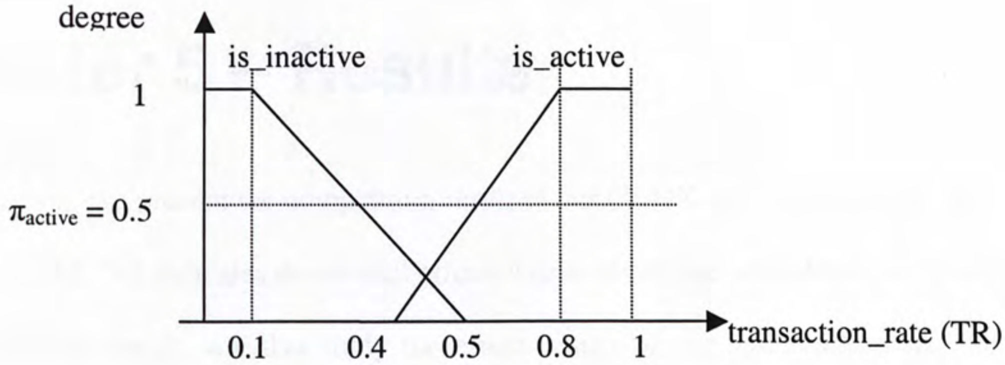


Figure 18: Fuzzy sets used in learning risk attributes

Finally, the parameters (β, γ, λ) are adjusted based on new calculated risk attributes. There is no unique approach to adjust those parameters. Basically, an agent reduces *ask* by a smaller amount when $RISK_S$ is higher. Similarly, an agent increase *bid* by a smaller amount when $RISK_b$ is higher. The following equations describe how the CUHK agent did in TAC '02.

For selling side:

1. $\beta_{s,1} = \beta_{s,1}$ (1 - 0.5*$RISK_s$); $\beta_{s,2} = \beta_{s,2}$ (1 - 0.5*$RISK_s$)
2. $\gamma_{s,1} = \gamma_{s,1}$ (1 + 0.5*$RISK_s$); $\gamma_{s,2} = \gamma_{s,2}$ (1 - 0.5*$RISK_s$)
3. $\lambda_{s,1} = \lambda_{s,1}$ (1 - 0.5*$RISK_s$); $\lambda_{s,2} = \lambda_{s,2}$ (1 - 0.5*$RISK_s$)
4. $\lambda_{s,3} = \lambda_{s,3}$ (1 - 0.5*$RISK_s$); $\lambda_{s,4} = \lambda_{s,4}$ (1 + 0.5*$RISK_s$)

For buying side:

1. $\beta_{b,1} = \beta_{b,1}$ (1 - 0.5*$RISK_b$); $\beta_{b,2} = \beta_{b,2}$ (1 - 0.5*$RISK_b$)
2. $\gamma_{b,1} = \gamma_{b,1}$ (1 - 0.5*$RISK_b$); $\gamma_{b,2} = \gamma_{b,2}$ (1 + 0.5*$RISK_b$)
3. $\lambda_{b,1} = \lambda_{b,1}$ (1 - 0.5*$RISK_b$); $\lambda_{b,2} = \lambda_{b,2}$ (1 - 0.5*$RISK_b$)
4. $\lambda_{b,3} = \lambda_{b,3}$ (1 - 0.5*$RISK_b$); $\lambda_{b,4} = \lambda_{b,4}$ (1 + 0.5*$RISK_b$)

Chapter 5 – Results

In this chapter, we present the competition result of our CUHK agent in TAC '02. The success of TAC '02 indicates the overall effectiveness of our agent model. Apart from the competition result, we also study the effectiveness of our agent model and its mechanisms under a controlled experiment environment.

The experiments are conducted in a relative small-scale market of eight agents. This market size is the standard of TAC '02 tournament. All analyses concluded draw from TAC '02 tournament are in this market scale. Besides, we also adopt the TAC standard in our controlled experiments. There are two reasons for this decision. First, it was hard to collect working agents of world-leading researchers to conduct experiments locally. Researchers may not want to disclose their agents' source codes and the binary executables may only work in TAC standard market. Second, results are more convincing when they are obtained in the standard TAC '02 tournament. Although we can implement other researchers' agents based on their publications, it can be very time-consuming and developed agents are not guaranteed to perform identically as the original version.

5.1 TAC '02 Competition

The latest TAC tournament was organized by Swedish Institute of Computer Science (SICS) from 17 June to 28 July, 2002. It attracted 19 teams from research labs and universities to participate[21]. The whole TAC tournament was arranged into 4 phases: (i) qualifying round, (ii) seeding round, (iii) semi-final round and (iv) final round. The purpose of qualifying round is selecting 16 out of 19 agents to participate semi-final round. The seeding round aims at dividing those 16 agents into two heats of semi-final round[22]. The top 4 agents and last 4 agents are placed into heat 1, while others 8 agents are placed into heat 2. After semi-final round, the 4 top scoring agents of each heat entry final round. The table below shows the finalists from 2000 to 2002.

Table 12: The finalists in previous TAC tournament

Position	TAC '00	TAC '01	TAC '02
1st	ATTac	Livingagents	whitebear
2nd	RoxyBot	ATTac	SouthamptonTAC
3rd	Aster	SouthamptonTAC	Thalis
4th	umbctac1	whitebear	UMBCTAC
5th	ALTA	Urlaub01	Walverine
6th	m_rajatish	Retsina	Livingagents
7th	RiskPro	CaiserSose	kavayaH
8th	T1	TacsMan	Cuhk

[21] The full participants list is available at http://www.sics.se/tac/tacparts.php.

[22] In TAC '02, the 8 top scoring agents in qualifying round had a place in semi-final directly. The other 11 agents compete for 8 remaining places.

5.1.1 Tournament result of our working agent

Our CUHK agent is fully implemented from our agent model. It placed one of 8 finalists in TAC '02, out of initial 19 teams. In qualifying round (120 games), the CUHK agent placed the 7th position with score 3040.30. In seeding round (440 games), it placed the 5th position with score 3055.42. In semifinal round (14 games), it placed 4th position with score 3353.54 in heat 2. In final round (32 games), it placed the 8th position with score 3247.83. The CUHK agent played around 500 games in total and obtained high positions in all rounds. From large number of games, it had played with many possible combinations of opponent agents. It is shown that our agent model performed well in all possible scenarios.

All 3 top scoring agents[23] in qualifying round obtain scores over 3300, which are 300 ahead of the CUHK agent. The score difference is pretty large. It is because the CUHK agent did not consider historical clearing prices before entering seeding round. Afterward, the CUHK agent became one of five agents obtaining score over 3000 in seeding round. The scores of top few agents were very close in seeding round. The top scoring agent, ATTac, had score 3131.25 which is only 3% ahead of our CUHK agent.

During final round, games were running in two separate TAC servers simultaneously[24]. Each server consists of 16 games. Participating teams needed to run two agents for both servers independently. Initially, we used different bidding strategies for two separate servers. The CUHK agent bid low price hotel reservations in one server but not in another server. After each server finished 9 games, we

---

[23] The 3 top scoring agents in qualifying round are "whitebear", "livingagents" and "UMBCTAC".
[24] The two servers are "tac.sics.se" and "tac4.sics.se".

observed that the CUHK agent which bid low price hotel reservations had a better performance. Hence, we decided to bid low price hotel reservations for 7 remains games in both servers.

Unfortunately, we made a mistake running both two agents in one server but no agent in another server (the 10th game). We stopped and restarted the agents immediately after we discovered this mistake. However, it was too late because two agents already acquired duplicate flight tickets in one server. Also, the prices of flight tickets in another server already rose a lot after we restarted the agent.

If we did not make this mistake, the CUHK agent would place the 4th position with score 3301.41. The new score is calculated by discarding scores of all finalists in the 10th game of both servers[25].

Table 13: The adjusted average scores of all agents after discarding the 10th game in both servers.

Position	Agent	tac.sics.se	tac4.sics.se	Overall
1	Whitebear	3379.1	3675.6	3527.3
2	SouthamptonTAC	3521.6	3351.7	3436.6
3	Thalis	3380.2	3312.8	3346.5
4	Cuhk	3226.5	3376.3	3301.4
5	UMBCTAC	3259.2	3322.7	3291.0
6	Walverine	3255.8	3295.7	3275.7
7	livingagents[26]	3516.7	3006.8	3261.8
8	kavayaH	2942.4	3418.2	3180.3

[25] Game 4589 in tac.sics.se and game 446 in tac4.sics.se.
[26] Livingagents missed 2 games in the server in tac4.siscs.se during the final round because of the timing problem.

5.1.2 Comparisons between CUHK, ATTac and Roxybot

In chapter two, we had discussed three recent researches on online auctions: Priceline, ATTac, and Roxybot. Priceline is a data structure storing predicted marginal costs of good items in vectors. It reduces complexity by converting the bundle completion problem into a function of marginal costs. In contrast, Priceline itself does not improve agent performance where cost prediction mechanisms do. ATTac and Roxybot participated in TAC '02 tournament and became two of the semi-finalists. The following table shows the scores of ATTac, Roxybot and our agent "CUHK" in qualifying and seeding rounds.

Table 14: Comparisons of CUHK, ATTac and Roxybot in qualifying and seeding rounds

	Qualifying round (120 games)	**Seeding round (440 games)**
CUHK	3040.30 (7^{th} position)	3055.42 (5^{th} position)
ATTac	1669.20 (23^{rd} position)	3131.25 (1^{st} position)
Roxybot	2900.84 (14^{th} position)	2855.27 (8^{th} position)

In the qualifying round, the CUHK significantly outperformed the other two agents. However, the performance of agents became closer in the seeding round. The main reason was that the agents were not fully implemented and still had bugs during the qualifying round. The participating agents were more completed and functioning during the seeding round. In the semi-final round (14 games), ATTac was assigned to participate in heat 1 while Roxybot and our agent "CUHK" were assigned to participate in heat 2. Finally, ATTac and Roxybot could not place the top four in their heats and thus not be able to enter final round.

Tournament results in the semi-final round and large number of samples in the seeding rounds indicated that our agent performed well when compared with groups like ATTac and Roxybot.

5.1.3 Low-price Bidding

From the participating experience in TAC '01, we believed an agent can stabilize the market by submitting a series of low bidding prices in multi-unit ascending auction. When the market become stable, clearing prices are more predictable.

We compare average scores of all agents in semi-final round under 2 cases: I) the CUHK agent did not bid low price hotel reservations and, II) the CUHK agent bid low price hotel reservations.[27]

Table 15: The average scores of the agents in heat 2 (with and without low price bidding)

Agent	I	II	Difference
PackaTAC	3065.67	3712.34	646.7
RoxyBot	2897.02	3816.23	919.2
TOMAhack	2510.76	3672.64	1161.9
Thalis	2920.61	3896.45	975.8
Walverine	3174.47	3567.47	393.0
cuhk	3039.48	3832.83	793.4
sics	3030.43	3435.89	405.5
whitebear	3162.17	3727.86	565.7

The scores of all agents were improved when the CUHK agent bid low price hotel reservations. As we discussed in Chapter 5, the low bids can stimulate competitors to participate actively. Hence, ASK was pushed to a high level earlier. Some agents

[27] There are 9 games for case I and 5 games for case II.

would switch to acquire other goods when they recognize demands of some goods were high. Effectively, market became stable because agents are not competing aggressively.

5.2 Controlled Environment

The success in TAC '02 tournament verified that our agent model and mechanisms work well in a wide range of situations. On the contrary, the competition results can only be for a reference purpose. The reason is that both TAC server and participating agents keep modifying during the 2-months TAC tournament. The competition result was not completely obtained from a controlled experiment environment (Stone et al., 2001). A statistic analysis also illustrated that no individual agent statistically outperforms another individual agent. On the other hand, the analysis stated that some groups of agents outperform other groups of agents (Lanzi et al., 2002).

5.2.1 Software platform

The TAC server of version 1.0b5 was selected as the test bed platform in this thesis. The software is downloadable from TAC '02 official page[28]. The previous versions either contain program bugs or misbehaved game rules. On the other hand, the later versions bundle with advanced features for competition purposes. As those advanced features are not useful for our experiments, we finally selected the simpler version 1.0b5.

[28] http://www.sics.se/tac/download_server.php

5.2.2 Aggressive agent vs. Adaptive agent

First, we study how agents are interacting in a market. We assume there are two types of agent: 1) aggressive agent and 2) adaptive agent. An *aggressive* agent decides its allocation once initially and then bid those desired goods at high bidding prices. An *adaptive* agent continuous reviews its allocation and avoids bidding expensive goods. We believed that the minority has an advantage in a market. When there is a large population of aggressive agents, adaptive agents performs better because they do not involve in bidding wars of aggressive agents. When there is a large population of adaptive agents, aggressive agents performs better because they always win their desired goods without strong competition.

In order to investigate how agents are interacting, we used two TAC agents to represent aggressive agent and adaptive agent. They were called L-agent (aggressive) and S-agent (adaptive).

L-agent (aggressive agent)

L-agent is a referencing "dummy" agent deployed by SICS team, bundled with the TAC server software package. The agent design was inspired by the top scoring agent "livingagents" in TAC '01. It is an aggressive agent with open-loop[29] strategy (Fritschi and Dorer, 2002).

L-agent selects the best travel packages with un-coordinated aspiration to clients (see Chapter 5, AAS). It takes initial flight ticket prices and average hotel clearing prices

[29] "Open loop" refers to a strategy without feedback and thus is not adaptive.

in the calculation.

```
foreach (arrivalDay = 1 to 4)
  foreach (departureDay = arrivalDay + 1 to 5)
     foreach (hotel SS and TT)
     {
       profit = 1000
                - inboundFlightPrice
                -  outboundFlightPrice
                -  TotalHotelPrice
                + PremiumForBetterHotel
                - TravelPenalty
                + EntertainmentBonus
     }
  }
}
```

The "EntertainmentBonus" is defined as overall fun bonus minus the total tickets cost. L-agent assumes the cost of a single entertainment ticket always equals 80. Suppose a 2-day travel package is allocated to a client. Then, the entertainment bonus to him equals (bestEntertainmentValue – 80) + (2ndBestEntertainmentValue – 80).

L-agent bid all desired flight tickets and desired hotel reservations at the beginning of a game instance. The bidding prices for flight tickets are always high enough for sure acquisition. Alternatively, L-agent bids hotel reservations aggressively with high bidding price 1000. The bids for flight tickets and hotel reservations never changed during a game. L-agent trades desired entertainment tickets opportunistically with fixed prices (60 for buying and 120 for selling).

S-agent (adaptive agent)

The skeleton of S-agent is identical to that of the CUHK agent. It contains components: Cost Estimator (CE), Allocation and acquisition solver (AAS) and the Bidders. S-agent always bids hotel reservations by increasing ASK with a fixed incremental price step.

Summary of features of S-agent:

- Predicts marginal costs of goods as how the CUHK agent does
- Consider increasing marginal cost as how the CUHK agent does
- Consider historical clearing prices as how the CUHK agent does
- Flight tickets
  - Bid twice per game instance
- Hotel reservations
  - Does not bid low price hotel reservations
  - Bid price is equal to ASK plus a fixed increment price step
- Entertainment tickets
  - Will not bid entertainment tickets[30]

Experimental Setting

We compared performances of S-agent and L-agent under varying population ratio. The whole experiment was composed of several tests. Initially, there was only one L-agent but there were seven S-agents. After each test of 10 games finished, we replaced one S-agent by another L-agent. In each test, an agent's performance was represented by its averaged scores over 10 games. If there were several agents of the same type in an experiment, the performance of that type of agents became mean of their averaged scores. The S.D. value refers to standard deviation of agents' scores of same type, not for their individual played games.

[30] Except the experiments for bidding strategies in continuous double auction

While comparing agents' performances, we always consider relative score rather than absolute score. First, an agent performance should be a comparative measurement. Second, there is randomness over agent's client preferences. Third, agent performance is affected by dynamic server loading and network variance.

Experimental Results

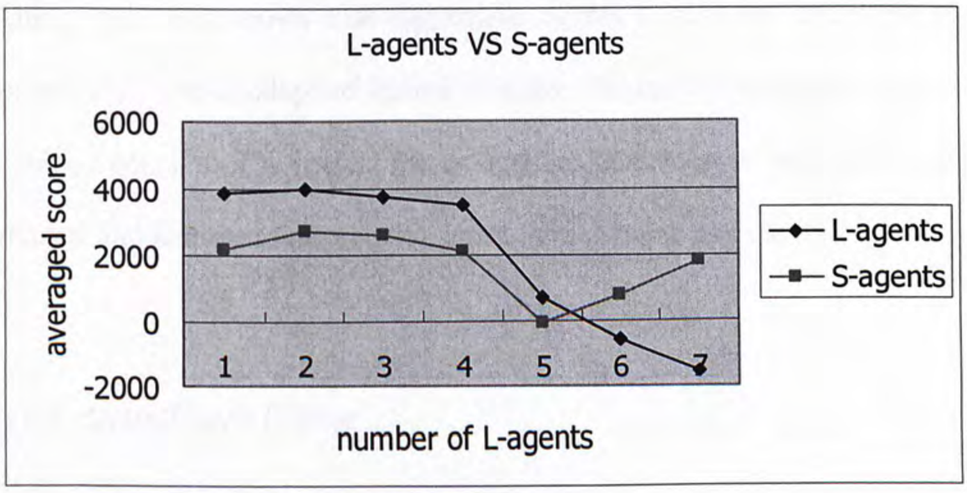


Figure 19: Comparisons between the performance of aggressive agents and adaptive agents

	No. of L-agents	1	2	3	4	5	6	7
	No. of S-agents	7	6	5	4	3	2	1
L-agent	Average scores	3837.50	3957.40	3719.00	3472.03	705.62	-518.07	-1506.56
	S.D.	N/A	168.29	129.00	188.39	322.21	580.28	555.91
S-agent	Average scores	2129.60	2676.72	2553.00	2098.13	-70.93	744.85	1815.50
	S.D.	680.39	359.70	375.38	365.73	1669.50	277.96	N/A

The result is consistent with our expectation. When there are many L-agent, S-agent outperforms L-agents. It is because S-agents is aware of high asking price ASK and hence avoids bidding those expensive goods. When there are many S-agents, the L-agents outperform S-agents. It is because no S-agent competes with aggressive

L-agents. Moreover, S-agents do not fight against each other result in low clearing prices. Hence, L-agents always success in acquiring goods with low clearing prices.

When there are more L-agents in a market, the average score of L-agents drop. It is because market advantages are shared among those L-agents. In addition, clearing prices are pushed to a higher level since competition become more aggressive.

The experiment shows that aggressive agents reduce the collective profit of the population; while adaptive agents stabilize the market economy (even they are all selfish agents[31]). It is perfect for an agent to behave as an aggressive agent in stable market and behave as an adaptive agent in non-stable market.

The Hawk-Dove Game

This model describes interactions between soft and tough behaviors in a population. Hawk is an aggressive bird and Dove is a peaceful bird. There are totally 3 cases for a bird meets another bird: 1) Hawk meets another Hawk, 2) Hawk meets Dove and 3) Dove meets another Dove.

Table 16: Payoff of Hawk and Dove in different cases

Payoff		Bird 1	
		Hawk	Dove
Bird 2	Hawk	-2, -2	2, 0
	Dove	0, 2	1, 1

[31]Selfish agent refers to agent which is aims at maximizing the own profit, but not the collective profit.

For case 1), two Hawks fight each other causing payoff of -2 each. For case 2), the Dove flees with payoff 0 and then the Hawk obtains payoff of 2. For case 3), two Dove share the payoff of 1 each. The model stated that there is equilibrium while the population ratio of Hawk is 1/3.

Consider an aggressive livingagents as a "Hawk" and an adaptive S-agent as a "Dove". While there is a large population of "Hawk", they fight with each other yielding "all-lost" result. Then, the small population of "Dove" has a higher payoff. While there is a large population of "Dove", the small population of "Hawk" has high payoff since it always obtain desired items without any competition.

5.2.3 Our agent model

In this experiment, we want to test the effectiveness of our agent model and its mechanisms under varying population of aggressive and adaptive agents. We expect to see that our agent model and mechanisms work well in all cases.

Experimental Setting

We let our CUHK agent play against varying population of L-agents and S-agents. Initially, there are one L-agents and six S-agents. Afterwards, we replace one S-agent by another L-agents per 10 games.

Experimental Results

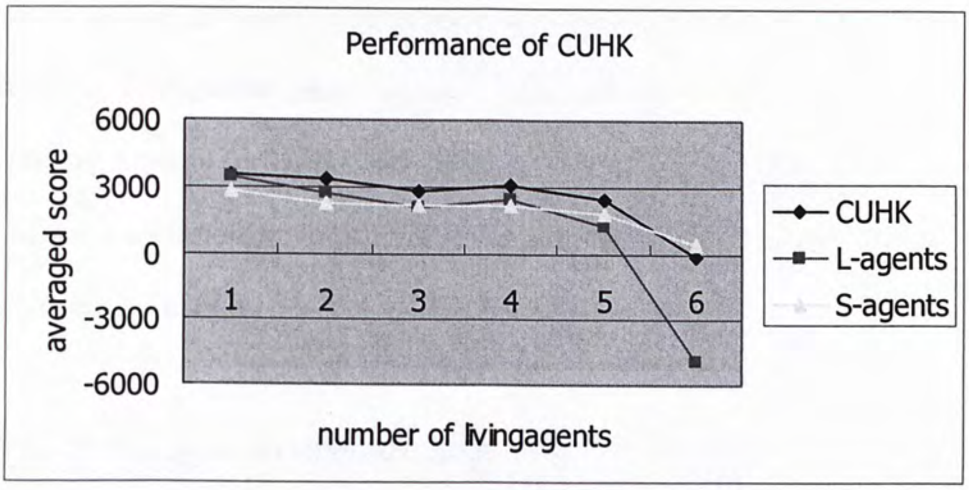


Figure 20: Performance of our CUHK agent under varying population of aggressive and adaptive agents

	No. of L-agents	1	2	3	4	5	6
	No. of S-agents	6	5	4	3	2	1
CUHK	Average scores	3568.78	3325.34	2806.39	3091.39	2395.06	-92.74
L-agent	Average scores	3436.54	2659.34	2101.64	2390.58	1204.91	-4934.19
	S.D.	N/A	230.37	234.13	147.39	500.48	902.08
S-agent	Average scores	2830.58	2205.08	2165.66	2195.43	1816.95	441.49
	S.D.	274.12	948.04	972.33	636.19	533.63	N/A

The CUHK agent performs well in all cases because our agent model contains both aggressive and adaptive characteristics.

While there are many adaptive S-agents, the market favorites to agent with aggressive attribute. Since our CUHK agent uses the utility obtainable from a travel package as a budget for bidding goods inside that travel package, it bids more aggressively than adaptive S-agents. As a result, the CUHK agent becomes an aggressive agent in a non-competitive market.

When there are many aggressive agents, a market becomes unstable and favorites to adaptive agents. Since AAS reviews allocation continuously, the CUHK agent can respond to dynamic price changes. Although the CUHK agent has an aggressive bidding strategy for hotel reservations, it can change the allocation when target hotel reservations become expensive. As a result, the CUHK becomes an adaptive agent in competitive market.

The CUHK agent significantly outperforms both L-agents and S-agents except two cases. First, the CUHK agent does not really outperform the only L-agents in a non-competitive market. It is because both of them have identical market advantage. Second, the only S-agent outperforms the CUHK agent in a competitive market. The CUHK agent can change allocation if and only if it can remove the commitments associated with its possibly winning items (see Chapter 4, Bid winning probability). Those commitments are removed only when other adaptive agents place bids on top of the bids associated with those possibly won items. When there is only one S-agent in a market, the number of adaptive agents is not sufficient to remove all commitments made. Hence, the CUHK agent cannot change its allocation effectively. Anyway, the CUHK agent still achieves good performance in those 2 cases.

5.2.4 Historical clearing price

We believe referencing historical clearing price for uncertain goods can effectively improve the agent performance. First, it acts as a measurement of competitors' aggressiveness. Although individual competitor's strategy is unknown, the collective behaviors can be somewhat observable from historical clearing prices. If all historical clearing prices are skyrocketing, it is likely the clearing price of current market will become high too. Second, historical clearing prices reflect the outcome of an agent strategy. Assume that the collective behaviors of competitors do not change dramatically. The same strategy would result in consistent outcomes.

Experimental Setting

To investigate effects of historical clearing prices, an alternative version of CUHK is examined under varying population of L-agents and S-agents.

- CUHK(NHIST): The CUHK agent considers the current ASK as marginal costs; without considering historical clearing prices.

Experimental Result

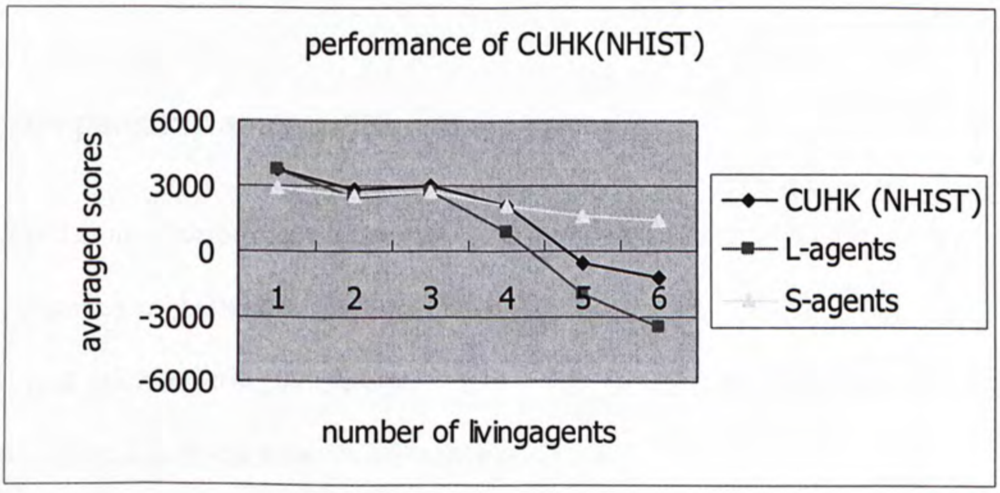


Figure 21: Performance of the CUHK agent without referencing historical clearing prices

	No. of L-agents	1	2	3	4	5	6
	No. of S-agents	6	5	4	3	2	1
CUHK(NHIST)	Average scores	3718.69	2787.80	3038.79	2096.18	-559.31	-1183.80
L-agent	Average scores	3761.36	2453.79	2745.84	842.18	-1945.80	-3450.04
	S.D.	211.86	327.87	197.52	825.83	242.45	866.12
S-agent	Average scores	2860.68	2536.35	2732.88	2090.70	1566.18	1398.80
	S.D.	N/A	621.50	265.09	93.19	47.94	N/A

It is shown that the performance of the CUHK is significantly decreased without referencing historical clearing prices. The effect is much clearer when there are more aggressive L-agents. In a competitive market, the clearing price can be skyrocketed and become far higher than the current ASK. Poor decisions are easily made without referencing any historical information. In a non-competitive market, prices are much predictable and hence the agent performance is not greatly affected.

Comparisons among different approaches

We use *median* of clearing prices in 10 most recently played games for representing outcomes of historical auctions. The idea is to filter the effects of abrupt clearing prices (skyrocketing or extremely low). We compare varying methods to show that using median is the most suitable approach.

- CUHK(Median): Median of clearing prices is used (original)
- CUHK(Mean): Mean of clearing prices is used
- CUHK(Sampling): One random selected clearing price is used.

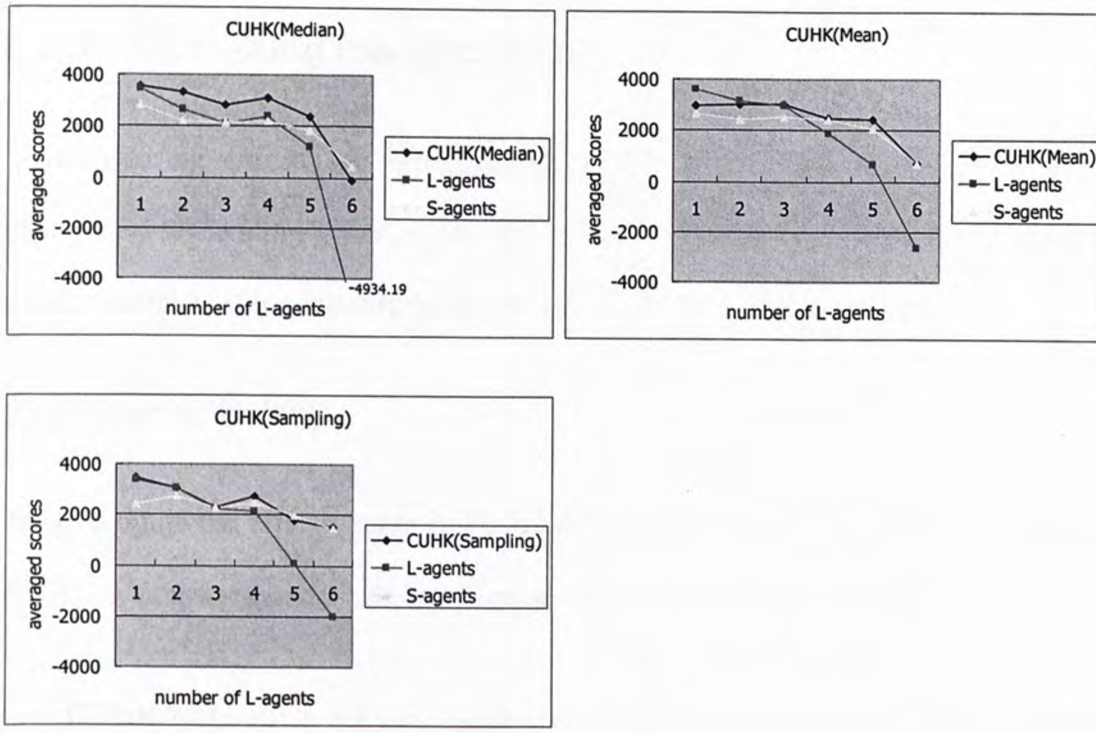


Figure 22: Comparisons between different approaches of referencing historical clearing prices

The median approach is more stable in outperforming competitors except when there are many aggressive agents. The reason of exception is based on the extremes clearing prices in an unstable market. The clearing prices in an unstable market sometimes can are skyrocketed as 1000, but sometimes can also be extremely low as zero.

Consider an example in TAC. Assume the clearing prices of TT3 in 10 most recently played games are 0, 0, 0, 0, 0, 0, 1000, 1000, 1000 and 1000. It is not surprising that next auction possibly closes at 0 or 1000, based on recent occurrence statistic. The median approach predicts marginal cost as zero, which is far from another extreme 1000. If the next auction closes at clearing price 1000, the penalty for inaccurate prediction becomes high.

5.2.5 Increasing marginal cost

Since clearing prices are unknown in multi-unit ascending auction, there is uncertainty for bidding goods in this type of auction. We believe an agent can improve the performance if it considers marginal cost of uncertain goods is increasing.

Experimental Setting

To investigate the effectiveness of increasing marginal cost, an alternative version of CUHK is examined under varying population of L-agents and S-agents.

- CUHK(NI): The CUHK agent considers marginal cost of goods is not increasing in multi-unit ascending auction. (auctions for hotel reservations in TAC)

Experimental Result

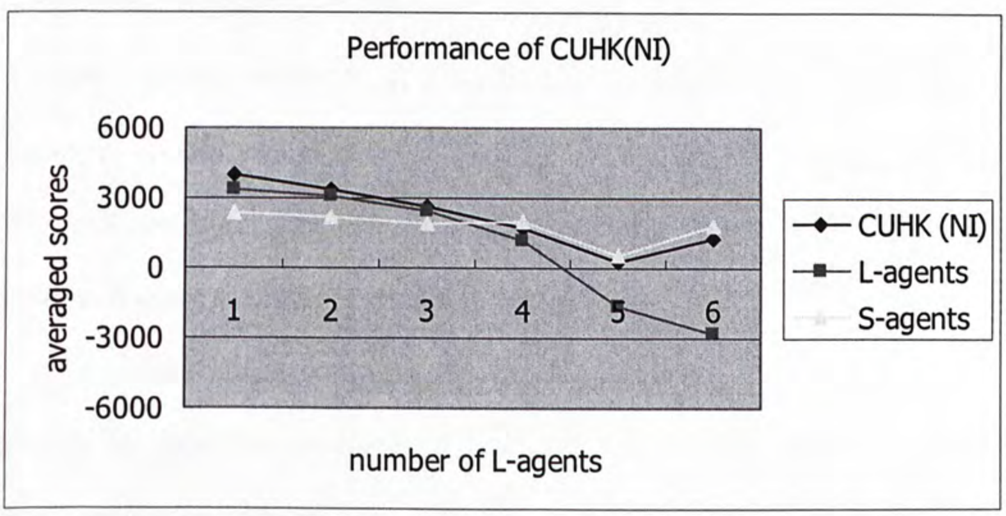


Figure 23: Performance of the CUHK agent without adopting increasing marginal costs

	No. of L-agents	1	2	3	4	5	6
	No. of S-agents	6	5	4	3	2	1
CUHK(NI)	Average scores	4017.41	3357.78	2630.11	1709.17	280.58	1278.59
L-agent	Average scores	3362.88	3112.06	2460.56	1169.56	-1625.72	-2830.11
	S.D.	N/A	224.23	196.43	534.00	288.60	486.60
S-agent	Average scores	2406.68	2222.06	1919.20	1971.31	575.84	1802.97
	S.D.	274.12	455.85	591.94	114.21	577.26	N/A

The diagram shows that the performance of our CUHK agent has slightly decreased. The difference is clearer when there are more aggressive L-agents. Compared to historical clearing prices consideration, the effectiveness of increasing marginal cost is less significant.

In a non-competitive stable market, clearing prices are more predictable and hence they are not far away from our prediction. As a result, penalty of inaccurate prediction is low even if an agent allocation heavily depends on a few kinds of goods. On the other hand, the penalty of inaccurate prediction becomes high in an unstable market. Suppose clearing prices are skyrocketed and become far higher than our prediction. An agent should pay all won goods items at skyrocketed clearing prices. Increasing marginal cost is successful in reducing the penalty of inaccurate prediction because it guides an agent to bid fewer and distributed goods.

During the agent development, we placed both CUHK and CUHK(NI) in the official TAC servers to compete with other participants. From game 3780 – 3823 in server "tac.sics.se", the CUHK and CUHK(NI) agents obtained an averaged score 2776.46 and 2457.06 respectively. The CUHK score is 300 ahead of the CUHK(NI). The result was consistent with our result under a controlled environment.

5.2.6 Bid winning probability

In multi-unit ascending auction, an agent should acquire its won items at clearing price. It is known as commitment. We believe an agent can perform better by considering bid winning probability (see Chapter 4, Bid winning probability). At a particular time, some commitments are already fixed and removable. If an agent "surely" wins those items, they should be considered as sunk costs.

Experimental Setting

To investigate the effectiveness of bid winning probability consideration, an alternative version of CUHK is examined under varying population of L-agents and S-agents.

- CUHK(CR): The CUHK agent assumes all commitments of its possibly winning items are removable.

Experimental Result

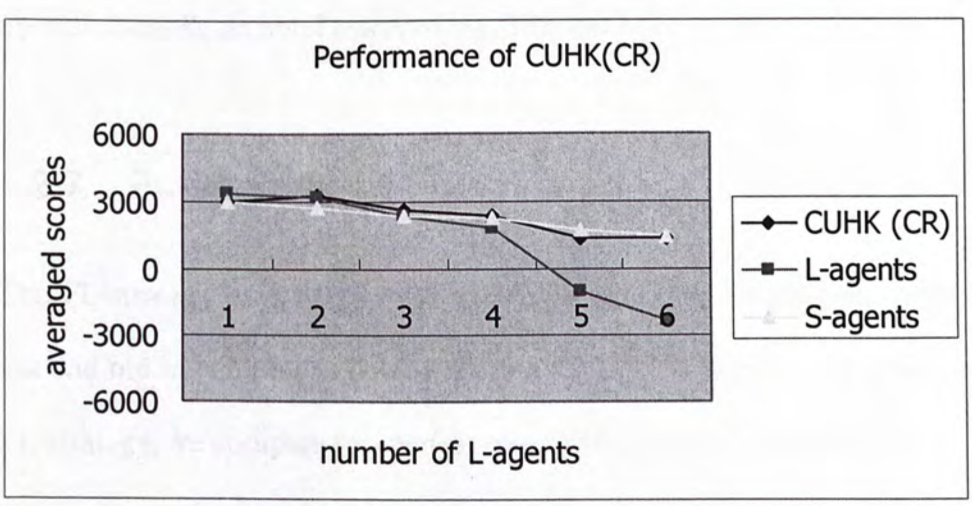


Figure 24: Performance of the CUHK agent without considering bid winning probability

	No. of L-agents	1	2	3	4	5	6
	No. of S-agents	6	5	4	3	2	1
CUHK(CR)	Average scores	2969.22	3210.99	2591.88	2320.78	1289.35	1289.68
L-agent	Average scores	3339.66	3196.00	2365.36	1743.86	-1019.93	-2298.99
	S.D.	N/A	71.38	157.01	71.36	235.08	902.08
S-agent	Average scores	2961.06	2686.34	2369.59	2277.71	1708.70	1398.44
	S.D.	115.88	360.58	310.21	381.27	169.28	N/A

Our result shows that the performance of CUHK is decreased without considering bid winning probability. It is because the CUHK(CR) agent is too optimistic such that it bid duplicate goods.

Consider an example. Suppose a client requires a hotel reservation at day 3 for his/her travel package. Initially, the CUHK(CR) agent allocates a hotel reservation SS3 to that client because SS3 is predicted to be cheaper than TT3. After several cycles, the predicted marginal cost of SS3 becomes higher than that of TT3. Hence, the CUHK(CR) agent switches to bid hotel reservation TT3 for that client. Unfortunately, the commitment associated with SS3 hotel reservation is not removable. Thus, The CUHK(CR) needs to pay for that SS3 hotel reservation. The CUHK(CR) agent ends up with 2 duplicate hotel reservations, SS3 and TT3.

5.2.7 FL-strategy

The FL-strategy uses fuzzy rules and fuzzy reasoning mechanism to decide the best ask and bid in continuous double auction (CDA). To examine the effectiveness of the FL-strategy, we compare this performance with another “A-strategy”.

A-strategy

The A-strategy is inspired by the bidding strategy of ATTac used for CDA. The identical strategy was used in both TAC '00 and TAC '01 (Stone et al., 2001; Stone et al., 2002). We believe A-strategy is a good reference for comparison. First, ATTac achieved excellent results in both TAC '00 and TAC '01. Second, there were other TAC entries using likewise bidding strategies (He & Jennings, 2002). In A-strategy, the initial ask is high and the initial bid is low[32]. Afterwards, the A-strategy steadily decreases the ask (increases the bid) when bundles are almost completed.

The A-strategy sells entertainment tickets only if it is more profitable than allocates them to clients. Assume ticket E is allocated to a client with fun bonus V(E). Then, A-strategy sells the ticket at min(200, V(E) + σ) where σ decreases linearly from 80 to 20 in TAC. In other words, the minimum profit is 20 for selling an allocated ticket. For an unallocated ticket E, the A-strategy sells the ticket at Δ decrease linearly from 100 to 50 in TAC. The reservation price of 50 prevents to benefit competitors too much. The agent acquires an additional ticket only if the utility obtainable is higher than the cost. For an acquiring ticket E, assume that the fun bonus obtainable of E is V(E). The A-strategy acquires the ticket at max(0, V(E) – σ) where σ increases linearly from 20 to 80. In other words, the minimum profit is 20 for acquiring an additional ticket.

[32] Remember that ask refers to selling price offered for an item of goods, while bid refers to buying price offered for an item of goods (see Definition)

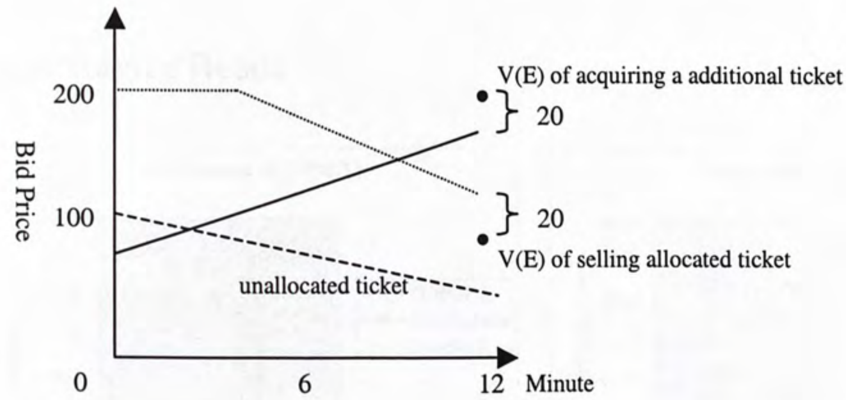


Figure 25: The A-strategy, which is aspired by ATTac's bidding strategy for CDA

The A-strategy is based on an idea that the uncertainty of completing bundles is decreasing. Thus, the initial ask and bid should be as optimistic as possible. An agent can accept ask or bid with less profit when bundles are almost completed.

Experimental Setting

We compare the performance of alternative versions of CUHK agents using FL-strategy and A-strategy under varying population of L-agents and S-agents.

- CUHK(FL): The CUHK agent using FL-strategy for CDA
- CUHK(A): The CUHK agent using A-strategy for CDA

In the experiment, the S-agent is extended to trade entertainment tickets at current prices. In other words, S-agent buys tickets at ASK+1 and sells tickets at BID-1. The livingagents always trades entertainment tickets at fixed prices. It buys tickets at 60 and sells tickets at 120.

Experimental Result

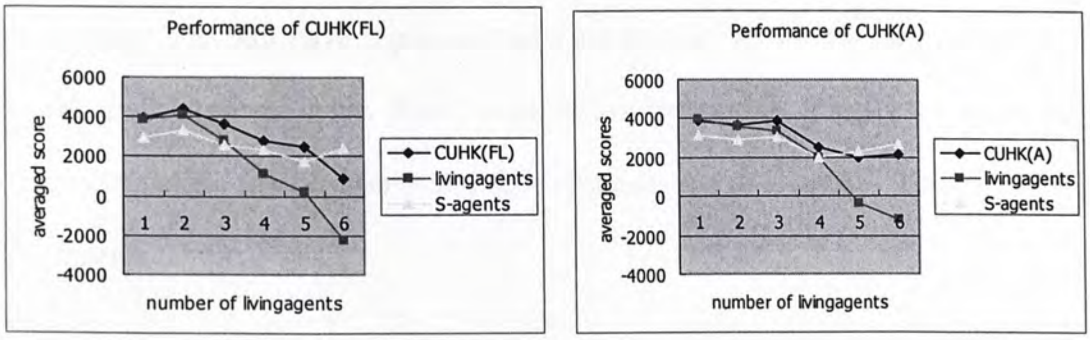


Figure 26: Comparisons between the performance of FL-strategy and A-strategy

	No. of L-agents	1	2	3	4	5	6
	No. of S-agents	6	5	4	3	2	1
CUHK(FL)	Average scores	3935.72	4375.47	3598.15	2793.37	2453.89	863.59
L-agent	Average scores	3844.74	4063.79	2770.72	1087.62	230.15	-2219.12
	S.D.	N/A	23.41	282.09	376.22	804.69	487.81
S-agent	Average scores	2949.27	3337.50	2579.02	2358.49	1741.41	2385.51
	S.D.	432.65	102.84	286.42	445.79	278.06	N/A
CUHK(A)	Average scores	3856.33	3657.93	3896.77	2522.06	1987.71	2118.03
L-agent	Average scores	3961.51	3564.90	3354.76	1999.30	-345.66	-1159.31
	S.D.	N/A	127.97	217.97	257.80	301.57	748.73
S-agent	Average scores	3148.95	2881.08	3077.76	2022.18	2319.01	2648.14
	S.D.	168.99	411.98	142.22	160.14	113.50	N/A

From the diagram, it is shown that FL-strategy is more stable in outperforming the competitors. The major reason is the adaptive behavior of FL-strategy. The FL-strategy adjusts this risk attributes based on recent transactions. Effectively the agent has more transactions than L-agents because it provides lower asks for buyer and higher bids for seller. On the other hand, the agent's transactions are more profitable than that of S-agents. An agent with FL-strategy is willing to take more risk for higher profit when the market demand is high.

There is an exceptional case when all competitors are L-agents. Unlike A-strategy, the FL-strategy does not have a pre-set minimum profit. The FL-strategy continuously adjusts risk attributes when there is no recent transaction. Finally an agent buys expensive tickets from L-agents and sells cheap tickets to L-agents. Thus, it benefits to L-agents.

Chapter 6 – Conclusion and Future work

We achieved several contributions in this thesis. First, we proposed a generic agent model using a divide-and-conquer approach. The performance of the model is examined under both TAC '02 tournament and our controlled testing environment. The experimental result showed that our model performed well in all competitive and non-competitive markets. Although the model adopted several previously published works, the innovation is to indicate appropriate methods and combine them in an effective way.

Second, we proposed a set of mechanisms to tackle divided sub-problems. The mechanisms include referencing historical clearing prices, increasing marginal costs for uncertain goods, considering bid winning probabilities and adopting fuzzy logic bidding in continuous double auctions. The effectiveness of each mechanism was examined in the TAC benchmark problem under a controlled environment. The mechanisms described in this thesis may be tailor-made for auction protocols in TAC. However, those mechanisms demonstrated the key underlying principles we adopted in bidding, cost prediction and allocation. Additional mechanisms for other auction protocols can be formulated using principles like budget bidding, marginal cost, sunk cost, opportunity cost and so on.

Third, this thesis studied interactions of aggressive and adaptive agents. The study of agents' interactions was motivated from tournament result in TAC '01: an open-loop, aggressively bidding "livingagents" obtained the highest scores among all agents in final round. Since most opponents changed their target goods to avoid bidding war, "livingagents" won all its desired goods without strong competition. We conducted a series of experiments to investigate how such interactions may determine their performance. The experimental results showed that a market exhibits properties similar to the Hawk-Dove Game, in which the minority has an advantage.

In conclusion, this thesis investigated how an agent can bid and bundle goods for multiple clients effectively from online auctions. The problem is interesting and complicated: clients' preferences are over bundles but goods inside a bundle are traded in various auction protocols. Moreover, there is no best strategy for all possible cases since agents with various strategies are interacting. The traditional auction theory is not applicable for the problem because it focused on single item and single auction protocol. To verify feasibility of our proposed agent model, a functioning agent called "CUHK" was developed and participated in TAC '02 tournament. The agent was successful in winning high positions in all tournament rounds. The tournament result is a good indicator of our agent model's effectiveness. The considerable efforts put in designing, development and fine-tuning are our key success factors. The experiments in this thesis are conducted in relative small-scale market of eight trading agents. The market size of eight agents is the standard of TAC '02 tournament. The software package downloadable in TAC '02 page is specified for the tournament standard as well. Although we may develop our own auction server with larger market scale, it is impractical for us to fully implement other agents from all world-leading groups. Hence, analysis and comparisons draw from TAC '02 tournament results are all in this market size. We propose experiments in larger-scale markets (e.g. stock market) are more suitable for future work.

The study in this thesis can be further extended in several directions. One valuable future work is to investigate how our agent model and mechanisms can be applied into other similar application domains. In TAC '03, the game scenario will change to a supply chain market problem. Participating agents will need to compete for client orders and acquire components from suppliers simultaneously. The game scenario is a general to represent many other supply chain markets: agents are required to concurrently involve in multiple markets of interacting goods, store up inventory with unreliable supply goods and estimate customers' demands. The supply chain market contains challenges similar to those we studied in this thesis: multiple market mechanisms, interacting goods, various opponents' strategies, and more. Hence, we believe our agent model is also applicable to supply chain market scenarios. Stock market is yet another interesting application domain similar to the market scenario we studied in this thesis. The stock market mechanisms are alike but more complicated to that of Continuous Double Auction. Investors can hold various kinds of stocks at the same time to balance risk and profit. Some investors are risk seeking while others are risk averse. Thus, the stock market application domain also involves problems of interacting goods, simultaneous markets and opponents' strategies. The TAC has features common to many real-life problems. We believed our generic agent model and mechanisms are applicable for a wide range of similar real-life problems.

Another future direction is to investigate the possibility of extending our agent model for sequential auctions and combinatorial auctions. In this thesis, we assumed goods are traded separately in simultaneous auctions. "*Sequential auctions*" is an auction protocol where auctions are running in rounds. In other words, a new auction will not start before the current auction close. "*Combinatorial auctions*" is another auction protocol where goods are traded in bundles instead of single item. We believe both auction protocols are worthy to be studied because they are common for trading bundles of goods.

Reference

Anthony, P. et al. 2001, Autonomous Agent for Participating in Multiple Online Auctions. In *Proc IJCAI Workshop on E-Business and the Intelligent Web*, Seattle, WA, 54-64, 2001.

Banpa, R. et al. 2001, Insights Analyses of Online Auction. *Communications of the ACM Vol.44, No. 11*, November 2001.

Fritschi, C. and Dorer, K. 2002. Agent-oriented software engineering for successful TAC participation. *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, 2002.

Greenwald, A. 2002. The 2002 Trading Agent Competition: An Overview of Agent Designs. *AI magazine*, 2002.

Greenward, A. and Kephart, J. 1999, Shopbots and Pricebots. *Proceedings of IJCAI '99 (International Joint Conferences on Artificial Intelligence)*, July 31- August 6, 1999, Stockholm, Sweden.

Greenwald, A. and Boyan, J. 2001a, Bid Determination in Simultaneous Auctions: A Case Study. In *Proceedings of the Third ACM Conference on Electronic Commerce*. Pages 210-212, 2001.

Greenwald, A. and Boyan, J. 2001b, Bid Determination in Simultaneous Auctions: An Agent Architecture. In *Proceedings of the Third ACM Conference on Electronic Commerce*, Pages 115-124, 2001.

He, M. and Leung, H. F. 2001, "*An agent bidding strategy based on fuzzy logic in a continuous double auction*", Proceedings of the Fifth International Conference on Autonomous Agents, Pages 61-62, 2001.

He, M. and Jennings, N. R. 2002, SouthamptonTAC: Designing a successful trading agent. *Fifteenth European Conference on Artificial Intelligence*, Lyon, 2002.

Ito, T. et al. 2000. Sycara. Biddingbot: a multiagent support system for cooperative bidding in multiple auctions. *In Proceedings of the Fourth International Conference on MultiAgent Systems*, Pages 399-400, 2002.

Klemperer, P. 1999. Auction Theory: A Guide to the Literature. *Journal of Economic Surveys*, Vol .13(3), 1999.

Lanzi, P. L. et al., 2002. A statistical analysis of the trading agent competition 2001. *SIGecom Exchanges*, 3(2): 1-8, 2002.

Stone, P. et al., 2001, ATTac-2000: An adaptive Autonomous Bidding Agent. *Journal of Artificial Intelligence Research,* 15:189-206, 2001.

Stone, P. et al, 2002. ATTac-2001: A Learning, Autonomous Bidding Agent. In *Workshop on Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, Bologna, Italy, 2002. Under review.

Stone, P. and Greenwald, A. 2000. The first international trading agent competition: Autonomous bidding agents. *Journal of Electronic Commerce Research*, 2000.

Surmacz, J. 2001, 'Analysts: Online auctions bid higher in e-commerce' [Online], *CNN.com.* 12 Oct. Available:
http://www.cnn.com/2001/TECH/industry/10/12/bidding.war.idg/index.html
[2002, Nov 16]

Vickrey, W. 1961, Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16, Pages 8-37, 1961.

Wellman, M. P. et al., 2001. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2), Pages 43-51, 2001.

Wellman, M. P., Greenwald, A., Stone, P., Wurman, P. R., 2002. The 2001 Trading Agent Competition. In *Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pages 935-941, Edmonton, 2002.