DESIGNING BIDDING STRATEGIES FOR AUTONOMOUS TRADING AGENTS

By

Minghua HE

A thesis submitted for the degree of

Doctor of Philosophy

School of Electronics and Computer Science,

University of Southampton,

United Kingdom.

September 2004

ABSTRACT

Increasingly many systems are being conceptualised, designed and implemented as marketplaces in which autonomous software entities (agents) trade services. These services can be commodities in e-commerce applications or data and knowledge services in information economies. In such systems, dynamic pricing through some form of negotiation or auction protocol is becoming the norm for many goods and customers. Thus, negotiation capabilities for software agents are a central concern. Specifically, agents need to be able to prepare bids for and evaluate offers on behalf of the parties they represent with the aim of obtaining the maximum benefit for their users. They do this according to some negotiation strategies. However, in many cases, determining which strategy to employ is a complex decision making task because of the inherent uncertainty and dynamics of the situation. To this end, this thesis is concerned with developing bidding strategies for a range of auction contexts.

In this thesis, we focus on a number of agent mediated e-commerce settings. In particular, we design novel strategies for the continuous double auctions, for the international trading agent competition that involves multiple interrelated auctions, and for multiple overlapping English auctions. All these strategies have been empirically benchmarked against the main other models that have been proposed in the literature and, in all cases, our strategies have been shown to be superior in a wide range of circumstances. Moreover all our models exploit soft computing methods, in particular fuzzy logic and neuro-fuzzy techniques. Such methods are used to cope with the significant degrees of uncertainty that exist in on-line auctions and we show they are a practical solution method for this class of applications. In developing such strategies we believe this work represents an important step towards realising the full potential of bidding agents in e-commerce scenarios.

Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Professor Nick Jennings for his help, fruitful discussions, insightful advice and encouragement throughout my research and especially for the research project he chose for me. I particularly appreciated his efficient and prompt response to on-going papers and after a number of rounds of his comments the papers became more elegant and clear than I could have imagined. Working with him has been an absolute pleasure for me. Associate Professor Ho-fung Leung also gave me a lot of help when I was involved in the Ph.D program at The Chinese University of Hong Kong. It is there that I started my research in agent mediated e-commerce. I appreciate the encouragement and advice he gave me.

I greatly acknowledge the people who have helped me a lot during the course of my Ph.D study. Mike Luck gave me advice on writing my thesis; Adam Prügel-Bennett gave me help with the neural networks; Chris Preist, Andrew Byde, Javier Esplugas Cuadrado of Hewlett-Packard Lab, and Philippe De Wilde of Imperial College gave useful comments and suggestions on the multiple auction work. Thanks also to Cora B Excelente-Toledo and F. Lopez y Lopez for their help with the use of *Winedit* and *Sun Java Studio* and their friendship. Specially, I would like to sincerely thank my thesis examiners Mike Luck and Carles Sierra for their comments, which have helped to improve the quality of the thesis.

During the Trading Agent Competition (TAC), I received help and support from many people. In particular, I would like to thank the TAC team at the University of Michigan (TAC-01) and Swedish Institute of Computer Science (TAC-02), including Michael Wellman, Kevin O'Malley, William E. Walsh, Daniel Reeves Chris Kiekintveld, Joakim Ericsson and Niclas Finne for setting up the TAC server and the competition and for responding so promptly to our requests. A number of people in the IAM group also helped in this endeavour. Mike Luck went to the US to present our agent at TAC-01 at very short notice, Yan Zheng Wei helped with agent connecting problems and monitoring the seeding round games in TAC-02 and Alastair J. Riddoch, Jon Hallett and Tim Chown helped us with various network difficulties that we experienced in both competitions.

I also received help from people who answered my questions and gave me comments on my papers. They are Yoav Shoham at Stanford University, Robert H. Guttman of

Frictionless Commerce, Carles Sierra at the Spanish Artificial Intelligence Research Institute, Chris Preist at Hewlett-Packard Lab, and Dajun Zeng at the University of Arizona. Thanks are given to the anonymous reviewers of the conferences and journals where I submitted various aspects of this thesis. Their insightful comments have significantly improved the quality of the thesis.

Finally, but in no way least, I gratefully thank my husband Xudong Luo who has given me constant encouragement, support and love throughout every stage of my Ph.D. For example, he encouraged me to participate the TAC. He said, “You like playing computer games and now you can enjoy the game and you get paid.” He also did what he could to make me concentrate on my work completely, he cooked delicious food for me and brought the food to the office. Sometimes the competition was running round the clock, he always stayed together with me late into the night. These make me work even better. I also thank my son Michael Yourong Luo. The little one’s smiles are much more than cries and he brings me much more happiness than trouble. Some people thought I could not achieve a Ph.D at the same time bringing up a new life but I did since I have such a lovely son and a helpful husband. I would like to thank my parents and parents-in-law for their support, encouragement and love during my Ph.D.

Chapter 1

Introduction

Electronic commerce (e-commerce) is increasingly assuming a pivotal role in many organisations. It offers opportunities to significantly improve (make faster, cheaper, more personalised and/or more agile) the way that businesses interact with both their customers and their suppliers. However, in order to harness the full potential of this new mode of commerce it is important to increase both the degree and the sophistication of the automation. To achieve this, we believe that a new model of software is needed. This model is based upon the notion of *software agents*—software entities that act on behalf of their owner in an autonomous fashion in order to achieve their objectives [Jennings, 2001].

A key aspect of such agents is that they need to interact with one another in order to effect trades (i.e., to buy and sell goods or services). In this context, on-line auctions—institutions where goods are sold on the Internet by the process of making bids and allocating goods according to competition—are the most widely studied and employed means of interaction. Indeed, it is estimated that there are currently some $2,500$ on-line auctions (http://www.internetauctionlist.com) for almost all types of goods imaginable. Such auctions are so prevalent because they are an extremely efficient and effective method of allocating goods or services [Wurman, 2001]. Now these auctions come in many different forms, each with their own rules and ensuing properties. English auctions (in which the auctioneer starts with the reserve price and solicits successively higher public bids from the bidders until no one will increase the bid and the last bidder is the winner) are used to sell books, laptops, cars, and almost everything. First-price sealed bid and second-price sealed bid auctions (in which bidders submit sealed bids to the auctioneer and the bidder who submits the highest bid wins and pays their bid (in the former case) or the second highest bid (in the latter case)) are used for a variety of procurement situations. Dutch auctions (in which the auctioneer starts with a high price and decreases it until a bidder accepts the current price) are used for selling gold and jewellery. Continuous double auctions (in which both buyers and sellers submit bids at any moment during a trading period) are used to trade stocks, agricultural commodities and currencies.

Given this variety of protocols, it is perhaps not surprising that the bidding strategies of the participants cover a similarly broad spectrum of behaviours. However to be effective, bidding strategies need to be tailored to the type of the auction in which they are to be used. In short, there is no optimal strategy that can be used in all cases.

In such environments, agents can perform a variety of different roles: (i) monitoring auctions in order to keep the user informed of the latest progress of the various auctions, (ii) analysing the market situation and history record in order to build a profile of potential other bidders and estimate the trend of the price, (iii) determining what auctions to place bids in, and (iv) deciding when, how many and how much to bid in order to get the best deal. Now the more of these activities that can be automated, the more time that can be saved by the users. Moreover, in more complex settings we believe agents are likely to be more effective than human bidders. This is partly a matter of speed (agents can process information more quickly than humans), but also because agents can more easily and more systematically perform the complex decision making required to operate effectively in many auction settings. Preliminary evidence for this is contained in [Das et al., 2001] which shows that software agents outperformed their human counterparters in continuous double auctions.

However, such automation is complex. In particular, perhaps the key challenge in this area is to design effective and efficient strategies that agents can use to guide their bidding behaviour. Although challenging, such developments are necessary if trading agents are to realise their full potential. Moreover, we believe that the existence of effective strategies will mean that on-line auctions can be more readily deployed as the marketplace protocol. In the absence of such strategies, there is still some reluctance to adopt on-line auctions even though they are the most obvious solution in many cases. Given this background, the research reported in this thesis addresses exactly this challenge for a range of e-commerce auction scenarios.

1.1 Trading Agents for On-line Auctions

This section briefly introduces the basic terms and concepts that underpin this thesis. In particular we discuss agents, e-commerce, with a particular focus on on-line auctions, and trading agents. A more comprehensive review of auctions and their associated strategies is given in Sections 2.2.5 and 2.3.3, and a more detailed analysis of the potential roles of agents in e-commerce is provided in Sections 2.2 and 2.3.

1.1.1 Interacting Agents

To act flexibly on behalf of its owner in order to achieve particular objectives, an agent must exhibit the following properties [Wooldridge and Jennings, 1995]:

- it needs to be *autonomous*: capable of making decisions about what actions to take without constantly referring back to its user;

- it needs to be *reactive*: able to respond appropriately to the prevailing circumstances in dynamic and unpredictable environments; and
- it needs to be *proactive*: able to act in anticipation of future goals so that its owner's objectives are met.

Thus, for example, a buyer may instruct their agent to: (i) find a reasonably cheap notebook computer, with the latest technical specification, that can be delivered within a week and that has a two year guarrantee or (ii) to find a flight that takes her from London to San Francisco with a weekend stop over in New York. Similarly, a seller may instruct an agent to: (i) monitor the prices of all its known competitors and automatically adjust its offerings (either up or down) so that they remain attractive to their target audience or (ii) to offer a reasonable discount scheme and better credit facilities to highly valued customers.[1]

Whilst pursuing their objectives, the agents will invariably need to *interact* with other similarly autonomous agents. This interaction can vary from simple communication (*e.g.*, a buyer agent asks a seller agent how much a particular computer costs), to more elaborate forms of social interaction (*e.g.*, cooperation, coordination and negotiation). In the latter case, for example, an agent may be required to:

- participate in an on-line auction, *e.g.*, monitoring bids, making bids, and withdrawing from the auction;
- negotiate on behalf of its owner, *e.g.*, to ensure the desired good will be delivered in time or to make the price acceptable;
- cooperate with other agents, *e.g.*, two sellers of the same product may need to pool their resources in order to meet a large customer order or two sellers may bundle their distinct (complementary) offerings to make a single more desirable product.

In the context of this thesis, buyer and seller agents interact in some form of marketplace. Such marketplaces are controlled by an owner—an individual or organisation that sets the rules of the environment in which the trade occurs. In first generation systems, the market owner is usually synonymous with the seller. However, this need not always be the case; examples of other possibilities are the numerous third party auction sites that now exist (*e.g.*, eBay (http://www.ebay.com), Fastparts (http://www.fastparts.com) and Freemarkets (http://www.freemarkets.com) and situations where buyers put out requests for tender (*e.g.*, Labx (http://www.labx.com) and General Electric (http://www.ge.com)). Generally speaking, there will be multiple e-markets trading in different types of goods (*e.g.*, e-markets for purchasing holidays, e-markets for buying computer equipment, e-markets for finding plumbers). Moreover, there will, in many cases, be multiple e-markets for the

[1] Agents are not just used in the domain of e-commerce, although this is arguably their most popular domain. Rather, agent technology should be viewed as a general solution paradigm for developing complex systems [Jennings, 2000]. Overviews of the application of agents in other domains can be found in [Chaib-draa, 1995, Jennings et al., 2000a, Jennings and Wooldridge, 1998, Parunak, 1998, Luck et al., 2003]

same (or similar) goods (*i.e.*, there will not be a single market for dealing with holidays). Both within and between different vertical market segments, there will be a significant variety in the way that e-markets are organised. These will vary from simple, fixed-price catalogues, through various forms of on-line auction (the case we focus on in this thesis), to sites where buyers and sellers can negotiate directly with one another (discussed in Section 2.2.5).

1.1.2 E-Commerce and On-Line Auctions

According to the Electronic Commerce Association: "electronic commerce covers any form of business or administrative transaction or information exchange that is executed using any information and communications technology" [Till, 1998]. However we believe this definition is too broad and so we limit this thesis to cover commercial activities conducted on the Internet [Hake, 1999]. Therefore, other forms of remote transactions (*e.g.*, ordering an air ticket over the telephone or buying a computer by credit card) are not considered.

Within this context, e-commerce systems provide both commercial information (such as products' prices and available quantities) and facilitate various commercial actions (*e.g.*, buying, selling and negotiation). Moreover, the increasing use of information technology in this area has led to fundamental changes in the way these commercial activities are undertaken (*e.g.*, the rise of dynamic pricing, the ability to easily compare many goods and the ability to negotiate contracts much more frequently) [Shaw, 2000].

Central to much of this improvement is the increasing use of on-line auctions as the fundamental means of trade. Specifically, an auction is initiated by an auctioneer, and involves several bidders making bids according to the imposed protocol (which may permit one or multiple rounds). The outcome of the auction is then usually a deal between the auctioneer and the successful bidder. Such auctions are central because they bring a number of benefits to e-commerce. Firstly, auctions are a very efficient and effective method of allocating goods or services, in dynamic situations, to the entities that value them most highly [Wurman, 2001]. Secondly, in contrast to many human negotiations, auctions can be very fast since decisions and exchanges can occur rapidly. Thirdly, online auctions make the physical limitations of traditional auctions disappear (*e.g.*, time, space and presence). This means they can provide millions of globally dispersed customers with more varieties of goods than can be selected through a traditional auction [Bapna et al., 2001].

1.1.3 Trading Agents

An auction scenario consists of two clearly distinct components: a protocol and a strategy [Binmore, 1992]. The former defines the valid behaviour of the agents during the interaction. For example in an English auction, an agent needs to bid the current price plus the bid increment. The later is the method an agent employs to achieve its negotiation objectives *within the specified protocol*. For example, in an English auction a strategy that could be adopted is to bid a small amount more than the current highest bid and stop bidding when the user's valuation is reached. Generally speaking, the protocol is set at design time, by the marketplace owner, and is publicly known to all the participants. In contrast, the strategy is determined by each individual participant and is typically private (divulging it may leave them open to exploitation). Nevertheless, protocol and strategy are inextricadly linked because the effectiveness of a strategy is very much determined by the protocol. Thus a strategy that is effective for one protocol may perform very poorly for other protocols. Moreover for some protocols, the optimal bidding strategy is easy to determine and simple to compute. For example, the strategy proposed above for an English auction is in fact optimal in the case that all the agents bid based on their real valuation. However in most cases there is no such simple solution and developing the strategy is a significant research challenge.

Against this background, we first consider a particular form of auction that is common in e-commerce, in which there are multiple sellers and multiple buyers that want to trade simultaneously. Such auctions are called *double auctions* [Friedman and Rust, 1992] and they allow sellers to indicate the services they offer at various prices (called asks) and buyers to indicate the services they desire and the price they are willing to pay (called bids). The most common variety of double auction is the continuous double auction (CDA) which permits trade at any time in a trading period (cf. trades only being allowed at discrete time points) and which allows buyers and sellers to continuously update their bids and asks at any time throughout the trading period [Kagel and Vogt, 1991]. The CDA is a powerful market mechanism because of its speed and efficiency, and is the mechanism underlying the organization of open-outcry trading pits at major international derivatives markets [Cliff, 2001]. As well as being important for e-commerce, CDAs are widely used in the non-online world to trade stocks, agricultural commodities, metals, currencies and derivative instruments [Friedman, 1993]. To date, a number of bidding strategies have been developed for the CDA (see Section 3.3.3 for details), however they all have shortcomings that reduce their effectiveness. Given this, we will develop a new more effective strategy.

The second case we consider is that where an agent has multiple potential auctions in which it could bid in order to obtain the desired good or service. We believe this *multiple auction* context is likely to become ever more important as ever more on-line auctions are created. For example, on eBay alone there are typically over 13,000 auctions for digital cameras at any one time. This expansion in the number of auctions, coupled with greater reliance on trading in them, means agents will increasingly need to buy multiple goods from these multiple auctions. For example, an agent may need to buy a flight from one auction and book a corresponding hotel from another or buy different components from different auctions in order to construct a new computer system. Moreover the possibility of exploiting multiple auctions in this manner is something that is not really possible without software agents. Given such huge and complex search spaces, the agent has a number of interrelated tasks to perform: (i) monitor relevant auctions, (ii) compare and make trade-offs between the offerings, (iii) decide in which auction to bid, (iv) when to bid, (v) how many items to bid for and (vi) at what price to bid. However, to date comparatively little work has been done in this area (see Sections 3.5, 4.4 and 5.6) and so we also seek to develop new strategies for this case.

1.2 Research Aims

In designing bidding strategies for the auction scenarios we consider in this thesis, and in many others besides, there are a number of common issues that need to be dealt with. Moreover, we believe it is possible to identify a range of concepts and technologies that form a solid foundation for tackling such problems in a broad range of situations. We now consider each of these in turn.

First, an agent often needs to predict the closing price of an auction in order to determine what bids or asks to make [Wellman et al., 2004]. In the multiple auction context, such predictions are needed in order to determine which of the available auctions is likely to be the best one to bid in. This procedure is usually based on some form of utility analysis so that the auctions selected will maximise the likely return of the bidder (as discussed in [Byde et al., 2002]). However, in order to calculate such returns, the likely clearing price of each auction must be estimated first.

Second, an agent needs to be adaptive so that it can tailor its bidding strategy to reflect the environment in which it is situated. Being adaptive is particularly important in cases where the environment is subject to significant changes. These could happen, for example, when the agent is trading with the same (or similar) partners or opponents repeatedly. In such cases, the agent can adapt its behaviour according to the behaviour of other agents so that it can obtain a better outcome for its owner. However, when things change, perhaps because the agent needs to trade with new partners, the parameters which characterise the strategy need to be adapted again. This is hard to achieve by manually adjusting the parameters since this is a slow and error-prone process and so it is desirable if the agent can adapt itself on-line and automatically.

Third, the agent needs to be flexible in generating and responding to bids. Specifically, we mean that in many cases it is important for the agent to have a soft constraint in bidding or matching a bid. For example, in a CDA, if a buyer agent is going to bid 1000, but the lowest ask in the market is 1001 then the agent may benefit by relaxing its constraint (slightly) and bidding the 0.1% higher that is necessary to make the trade.

Fourth, different users will have different attitudes to risk and these can, in turn, have a significant impact on the performance of the agent. Individual attitudes to risk can be characterised according to how an agent approaches a fair gamble [Schotter, 1994]. A risk-seeking agent will prefer fair gambles to sure results, a risk-averse agent will take minimal risks with its actions and so rejects fair gambles, and a risk-neutral agent is indifferent if the sure result and the gamble have the same expected utilities. Even when using the same strategy, the adoption of different risk attitudes can make a significant difference to the outcomes obtained. Specifically, in auction contexts we believe such attitudes to risk should be influenced by the supply and demand within the market. The more the demand, the more the competition, and a risk-averse attitude is more effective. In contrast, the more the supply, the less the competition, and thus a risk-seeking attitude will bring more profit.

Fifth, an agent needs to be able to make trade-offs when participating in auctions with multi-attribute goods or services [Luo et al., 2003b, Luo et al., 2003a]. This is because there will often be some form of conflict between the attributes of the goods. For example, in a flight auction where goods are described in terms of their price and travel date, if a user wants to buy a cheap ticket, he needs to travel in the weekdays; however, the ticket is much more expensive at weekends. Given this, the agent needs some method to make trade-offs between the different attributes.

Given these aims, this thesis uses a range of fuzzy techniques to cope with the inherent uncertainty present in all of these activities. This uncertainty can come from a number of sources including the sellers, the bidders, the supply and demand quantity in the market or even the time of bidding. Such factors are usually highly ambiguous and fuzzy theory has proved itself to be effective in a range of applications with these characteristics [Fraichard and Garnier, 2001, Yao and Yao, 2001, Mohammadi et al., 2000]. In particular, we exploit fuzzy logic because of its intuitive nature and its embodiment in fuzzy rules means that it should be readily comprehensible to the agent's designers.

Against this background, this work is concerned with the design of bidding strategies for a number of particular auction contexts. To start, we choose a single auction protocol, CDA, since it is a complex auction type that has no dominant strategy (the best thing to do, irrespective of what the others do [Sandholm, 1999b]). This aspect of our work involves developing a strategy that both a buyer agent and a seller agent can use. Specifically, as a buyer, an agent needs to decide when to place a bid and at what price and as a seller, an agent needs to decide when to place an ask and at what price. To do these things effectively, an agent needs to (i) predict the likely transaction price at which trades will occur; (ii) adapt itself to suit the prevailing market context because the demand and supply in the market and other bidders' strategies are changing; (iii) vary its risk attitude in response to changes in the supply and demand so that it can deal with different situations; and (iv) relax its constraints on price in order not to miss out on deals by insignificant amounts.

Having developed a bidding strategy for a single auction, we turn to the more complex problem of developing an agent that can bid across multiple auctions that may be operating different protocols. This is a sufficiently challenging and important problem that an international competition was established in the area. In this Trading Agent Competition (TAC), software agents compete against one another in 28 simultaneous auctions in order to procure travel packages (flights, hotels and entertainment) for a number of customers. In this context, the failure to obtain some goods in one auction may lead to the failure of a whole travel package. Thus, there is a strong need to co-ordinate bidding across a range of interrelated auctions. In tackling this problem, we believe an effective trading agent needs to: (i) estimate the likely closing prices of the various auctions; (ii) adjust its bidding behaviour to suit environments in which the competition is more or less strong; (iii) vary its risk attitude to achieve effective outcomes; (iv) make trade-offs between buying flights early at low prices, but before the corresponding hotels can be guaranteed and buying flight late with guaranteed hotels but at high flight prices.

After dealing with the bidding issues in TAC, we wanted to generalise the multiple auction context to consider the most common type of auction in e-commerce (the standard English auction). A bidding strategy for this scenario needs to be highly adaptive because the market is very dynamic and uncertain due to the random number of auctions, agents and goods. Specifically, it is important to adapt the parameters involved in the bidding strategy so that the agent can adjust itself to suit the environment. Beside the feature of adaptivity, the following features need to be considered when designing a strategy for this context: (i) price prediction is important because the expected auction closing prices are needed to select the auctions to bid in; (ii) the risk attitudes of the agent need to be varied because the demand and supply are changing in the market; (iii) the agent needs to have soft constraints when selecting the auctions to bid in because this will increase the chances of obtaining the good in a good price; and (iv) trade-offs between the various attributes need to be made because it is very unlikely to maximise each attribute value when competing with other agents in the market.

1.3 Research Contributions

The work described in this thesis makes a number of important contributions to the state of the art in the area of bidding strategies that autonomous trading agents can use in a number of auction contexts. Specifically:

- We develop a novel fuzzy logic based bidding strategy that agents can use to participate in CDAs [He et al., 2003]. The effectiveness of the strategy is demonstrated by empirically benchmarking it against the main other strategies that have been proposed in the literature and this evaluation shows our strategy is superior in a wide range of situations.
- We develop novel fuzzy based bidding strategies that an agent can use to bid across multiple simultaneous auctions and purchase a number of interrelated goods [He and Jennings, 2003, He and Jennings, 2004]. In both cases, the agents can

vary their bidding behaviour according to their perception of the marketplace in which they are currently operating. The effectiveness of these strategies is demonstrated by our participation in the International Trading Agent Competition (in 2001 and 2002) in which our agent was the most successful participant over both competitions.

- We develop and implement, for the first time, a strategy that an agent can use to buy multiple independent goods from multiple English auctions [He et al., 2004]. This strategy uses fuzzy set theory to find the closest auctions to the optimal set to bid in and neuro-fuzzy techniques that can adapt the parameters in its fuzzy neural network through off-line and on-line learning. The effectiveness of the strategy is empirically demonstrated in a flight auction scenario and again we show our agent obtains a higher overall satisfaction degree than the related strategies available in the literature.

In addition to making advances in bidding strategies, this work is also one of the first to employ fuzzy theory (especially fuzzy logic and neuro-fuzzy techniques) in the area of agent-mediated e-commerce (see Section 2.2.5 for a discussion of other work in this area). In so doing, we further advance the claim that fuzzy techniques are a suitable tool to address the uncertainty that is inherent in many aspects of agent mediated e-commerce. In more detail:

- Fuzzy reasoning is successfully used in predicting the closing prices of the auctions. This can be observed from our work in predicting the closing prices of the hotel and flight auctions in the TAC and the various English auctions in the multi-auction scenario. In the former case, the auctions are interrelated, and the factors that are mainly related to the change of the hotel auction's closing prices are used in the fuzzy rules (Section 4.2.7). In the latter case, fuzzy reasoning rules are realised through a neuro-fuzzy network, however, the principle of the reasoning is the same. The main factors are expressed as fuzzy sets which correspond to neurons of the lowest layer of the network (Sections 5.3.1 and 5.3.2).
- Fuzzy techniques are used to enable our agents to adapt their bidding behaviour and strategy to better fit their prevailing circumstances. In our work, there is a progress from somewhat limited adaptation to highly flexible adaptation. In our work on CDAs, the adaptation is based on the frequency of transactions made by the agent and its aim is to change the risk attitude of the agent. Specifically, if the agent waits too long to conduct a deal, it will adapt its risk attitude to take less risk because it means its price constraint is far away from the market transaction price; on the other hand, if the agent transacts very frequently, it will adapt its risk attitude toward being more risky because it means the agent can make more profit by raising its threshold. This adaptation is somewhat limited because an agent can only vary its learning rate (Section 3.4). In the TAC, the agent has a number

of predefined parameters for each trading environment it encounters (competitive, semi-competitive and non-competitive). When the agent senses significant changes in the prices of the auction, it shifts its parameters involved in the fuzzy rules between these environments. However, the drawback of this adaptation is that the parameters are fixed in a specific environment. Having all of these in mind, the adaptation for the FNN agent is flexible. It uses a fuzzy neuro-fuzzy network to propagate the errors from the top level down to the lowest level, and all the parameters can change flexibly through learning of the real data.

- Fuzzy set theory is good for generating a flexible solution to bidding. It has been used in both TAC and the multiple auction context in order to find the closest solution to the optimal one. This solution covers what bid/ask to submit and which auction should be chosen to bid in for the multiple auction context. For example, the strategy used in the CDA of the TAC is applicable to any other type of CDA. In a CDA where an agent submits both asks and bids, most agents calculate an optimal bid price that can maximise its utility, and wait for the bid to be matched by other agents (*e.g.*, [Gjerstad and Dickhaut, 1998]). However, the strategy employed by our agent uses the similarity in fuzzy sets which accepts bids and asks if they are close enough to the optimal bid. In the multiple English auction scenario, the agent first evaluates the available auctions and then chooses the ones with the highest satisfaction degree. However, it also monitors all the other auctions which close earlier than this auction and it will bid in any auction that has a very close satisfaction degree to the optimal one.
- The risk attitude of the agent has an important influence on its performance and so it is usually adjusted by one or more parameters of the strategy. In particular, a key reason for this adjustment is often caused by the relationship of demand and supply and the kind of opponents in the market. To this end, in all of the agents we designed, risk attitudes are considered. In the CDA, the risk attitude parameters are the parameters in the fuzzy rules. At first, these values are pre-defined by the user, however subsequently they are adapted through the transaction rate of the agent. In the TAC, we vary the agent's behaviour from risk-seeking to risk-averse to suit the kind of environment. This is realised through the flight ticket numbers to buy at the beginning of the game and what price to bid in the hotel auction. In the multiple English auction case, this risk attitude parameter is the threshold by which the agent chooses the auctions that are close to the optimal set.

1.4 Published Papers

The following papers have been published based on this thesis:

- M. He, N. R. Jennings and A. Prügel-Bennett (2004) "A Neuro-Fuzzy Bidding Strategy for Buying Multiple Goods in Multiple English", submitted.

- M. He, N. R. Jennings and A. Prügel-Bennett (2004) "An adaptive bidding agent for multiple English auctions: A neuro-fuzzy approach", Proc. IEEE Conf. on Fuzzy Systems 1519-1524.
- M. He and N. R. Jennings (2003), "Designing a successful trading agent using fuzzy techniques", IEEE Trans on Fuzzy Systems 12 (3) 389-410.
- M. He and N. R. Jennings (2003), "SouthamptonTAC: An adaptive autonomous trading agent", ACM Trans on Internet Technology 3 (3) 218-235.
- M. He, H. Leung and N. R. Jennings (2003), "A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions", IEEE Trans on Knowledge and Data Engineering 15 (6) 1345-1363.
- M. He, N. R. Jennings and H. Leung (2003), "On agent-mediated electronic commerce", IEEE Trans on Knowledge and Data Engineering 15 (4) 985-1003.
- M. He and N. R. Jennings (2002) "SouthamptonTAC: Designing a successful trading agent", Proc 15th European Conf. on AI (ECAI-2002), 8-12.

1.5 Thesis Structure

The thesis is structured in the following manner:

Chapter 2 surveys and analyses the general state of the art of agent-mediated e-commerce. Specifically, our analysis concentrates on business-to-consumer (B2C) and business-to-business (B2B) contexts because we believe these are the most relevant for agent technology. In the former case, we discuss the roles of agents in terms of: need identification (what is the need), product brokering (what to buy), buyer coalition formation (find other buyers), merchant brokering (who to buy from) and negotiation (find the terms and conditions of transaction). In the latter case, the roles of agents are discussed through the *business-to-business transaction* model and particular attention is paid to partnership formation (find partners to collaborate), brokering (match buyers and suppliers) and negotiation. In both cases, however, auctions are identified as a central concern and so particular attention is focused on them and the roles that agents can play in them.

Chapter 3 concentrates on CDAs; developing new algorithms for buyer and seller agents. These algorithms employ heuristic fuzzy rules and fuzzy reasoning mechanisms in order to determine the best bid to make given the state of the marketplace. Moreover, we show how an agent can dynamically adjust its bidding behaviour to respond effectively to changes in the supply and demand in the marketplace. We then show, by empirical evaluations, how our agents outperform four of the most prominent algorithms previously developed for CDAs.

Chapter 4 describes the design, implementation and evaluation of SouthamptonTAC, one of the most successful participants in both the Second and the Third International Trading Agent Competitions. Our agent uses fuzzy techniques at the heart of its decision making: to make bidding decisions in the face of uncertainty, to make predictions about the likely outcomes of auctions, and to alter the agent's bidding strategy in response to the prevailing market conditions.

Chapter 5 presents the design, implementation and evaluation of a novel bidding algorithm that a software agent can use to obtain multiple goods from multiple overlapping English auctions. Specifically, an *Earliest Closest First* algorithm is proposed that uses neuro-fuzzy techniques to predict the expected closing prices of the auctions and to adapt the agent's bidding strategy to reflect the type of environment in which it is situated. This algorithm first identifies the set of auctions that are most likely to give the agent the best return and then according to its attitude to risk it bids in some other auctions that have approximately similar expected returns, but which finish earlier than those in the best return set. We show, through empirical evaluation against a number of methods proposed in the multiple auction literature, that our bidding strategy performs effectively and robustly in a wide range of scenarios.

Chapter 6 recaps the main contributions of this thesis and highlights the key open problems that need to be addressed if trading agents are to reach their full potential in agent-mediated electronic commerce.

Chapter 2

Agent Mediated Electronic Commerce

Given the aims and objectives of this research (as outlined in Section 1.2), this chapter places our work on bidding strategies for trading agents in on-line auctions into the wider context of agent-mediated e-commerce. In reviewing the state of the art in this area, this chapter builds upon a number of previous reviews of agent-mediated e-commerce. Particularly prominent amongst these are Guttman *et al.*'s review of agents in B2C e-commerce [Guttman et al., 1998] and Sierra and Dignum's roadmap of agent mediated e-commerce in Europe [Sierra and Dignum, 2001]. Other important sources include [Liu and Ye, 2001, Ye et al., 2001, Sandholm, 1999a, Beam and Segev, 1996, Brenner et al., 1998, Ma, 1999, Kalakota and Whinston, 1996, Liang and Huang, 2000, Murch and Johnson, 1999]. Neverthelsss, this chapter both extends and updates this previous material and also tries to present a more integrated and coherent view on the field. For example, in contrast to the aforementioned work, we categorise and systematically analyse applications of agent-based e-commerce in the B2C and B2B domains (using the consumer buying behaviour (CBB) model and the business-to-business transaction (BBT) model respectively). We extend the traditional CBB model so that it covers more B2C behaviours (such as buyer coalition formation), and we identify more uses for agents in the BBT model.

The remainder of this chapter is organised as follows. Section 2.1 introduces the roles of agents in e-commerce. Section 2.2 describes the basic roles and techniques of agents in B2C e-commerce. Section 2.3 performs a similar analysis for B2B e-commerce. Finally, Section 2.4 concludes this chapter.

2.1 Agents for E-Commerce

Agent-mediated electronic commerce involves software agents acting on behalf of some or all the parties in e-commerce transactions. The rationale for introducing such agents in e-commerce scenarios is to offer faster, cheaper, more convenient, and more agile ways for both customers and suppliers to trade. To realise this, a broad range of social, legal and technical issues need to be addressed. These issues relate to things such as security, trust, payment mechanisms, advertising, logistics and back office management

[Turban et al., 1999, Singh, 1999, Garfinkel et al., 2001, Zacharia et al., 2000]. In particular, trust has both a social and a technological facet and both of these need to be further addressed if users are to be happy to delegate increased autonomy to a software agent that acts on their behalf. From a social perspective, people will need to become happier to let a piece of software make decisions for them. This is something that is likely to take time and will only occur as agents show what they are capable of. From a technical perspective, agents need to clearly understand the limit of their responsibility and to act efficiently and safely within these bounds.

Even more fundamental than these issues, however, is the very nature of the various actors that are involved in e-commerce transactions. In most current (first generation) e-commerce applications, the buyers are generally humans who typically browse through a catalogue of well defined commodities (*e.g.*, flights, books, compact discs, computer components) and make (fixed price) purchases (often by means of a credit card transaction). However, this modus operandi is only scratching the surface of what is possible. By increasing the degree and the sophistication of the automation, on both the buyer's and the seller's side, commerce becomes much more dynamic, personalised and context sensitive. Moreover, these changes can be of benefit to both the buyers and the sellers. From the buyer's perspective, it is desirable to have software that could crawl all the available outlets to find the most suitable one for purchasing the chosen good (*e.g.*, the one that offers the cheapest price, the highest quality, or the fastest delivery time) and that could then go through the process of actually purchasing the good, paying for it and arranging delivery at an appropriate time. From a seller's perspective, it is desirable to have software that could vary its offering (in terms of price, quality, warranty, and so on) depending on: the customer it is dealing with (*e.g.*, offering discounts or special offers to particular target groups), what its competitors are doing (*e.g.*, continuously monitoring their prices and making sure its own price is competitive), and the current state of its business (*e.g.*, if it has plenty of a particular item in stock, it may be appropriate to reduce the price in order to try and increase demand).

Auctions are the most widely used kind of e-commerce today and agents have an active role in this area. In such online auctions, agents act on behalf of their users to monitor the auctions, analyse the market situations and decides when and how much to bid for the desired items. Agents can do this much faster, quicker and efficient than human bidders. For example, [Das et al., 2001] shows that agents outperform their human counterparters in a particular auction setting. Furthermore, a new possibility afforded by an agent-based approach is that a user can compete in multiple auctions simultaneously. Such a strategy has several advantages over participating in single auctions; for example, it can increase the chance of getting the good for customers; bring greater profit to customers by comparing multiple auctions and transacting at the cheapest price; and make the auction markets themselves more efficient by ensuring the transaction price is close to the equilibrium price [Preist et al., 2001]. Some good examples of auction websites are AuctionBot (http://www.auctionbot.com), Yahoo Auctions (http://auctions.shopping.yahoo.com) and eBay and more information on auctions can be found in [Milgrom, 1989, Wolfstetter, 1999, Milgrom and Robert, 1982, Vickrey, 1961, McAfee and McMillan, 1987].

To achieve this degree of automation, and move to second generation e-commerce[1] applications, we believe software agents have a key role to play. IDC (http://www.idc.com) estimates that the global market for software agents grew from $7.2 million in 1997 to $51.5 million in 1999 and that it will reach $873.2 million in 2004, with a compound annual growth rate of 76.2% between 1999 and 2004. They also assume that the dramatic growth in B2B e-commerce will accelerate the demand for agents. To this end, and in order to motivate the potential of agent mediated e-commerce, we consider the following medium term scenarios[2] as examples of what will be possible [Jennings et al., 2000b].

Scenario 1: Finding closest match to buyer's requirements. A buyer decides that they would like a holiday in one of the Greek islands, they would like to go next Friday, they would like to fly from London, and that the total cost should be less than £300. Their software agent is instructed to go and find out what is available and to report the options back to the user who will make the ultimate choice. In order to fulfil this objective, the buyer agent determines those e-markets that deal with leisure activities. From those, it tries to find out holidays that meet the specified requirements. However, it finds no appropriate fixed price offerings and after observing the outcome of several online auctions it decides that it will be very unlikely that it will be able to meet all of these requirements. It therefore decides to relax some of the user's constraints and tries to find holidays that are similar. The agent decides to relax the user's stated requirements in the following way: it looks for holidays to the Greek islands that leave any day next week, that leave from non-London airports in the UK next Friday and that cost up to £400. With these new requirements in place, the buyer agent returns to the relevant e-markets, collects the offerings that satisfy these new requirements and returns them to its user with an explanation of why it acted in this way.

[1] Second generation systems are here characterised as having a greater degree of automation on both the buyer's and the seller's side. Like many classifications, however, this distinction is not absolute and there are areas of uncertainty between the generations. Moreover, the same is also true of agent-mediated e-commerce in general. While there are some systems that are clearly agent-mediated and some that are not all agent-based, there is a degree of uncertainty in some cases. This is caused by the fact that in such systems agents are rarely the only technology that is used. Often an e-commerce system will be composed of a variety of technologies, only a fraction of which will be agent-based.

[2] We do not focus on current applications because they do not adequately highlight the full potential of agent-mediated e-commerce. Current applications tend to use agents in reasonably straightforward ways. Also organisations that have adopted agent-based techniques often do not disclose this fact for reasons of retaining competitive advantage. Focusing on medium-term scenarios overcomes both of these concerns without having to gaze too far into the future (which is notoriously unpredictable).

Scenario 2: Acting across multiple e-markets. A buyer decides that they would like to purchase a new laptop computer; they want a reasonably high specification, are prepared to pay for a good quality brand name, but it must be delivered within a week. Their software agent is instructed that they are prepared for the agent to find the most appropriate model, negotiate the best potential deal available, but that the user would like to make the final choice about purchase. In order to fulfil this objective, the buyer agent determines those e-markets that deal with selling computer equipment. From these, it selects those e-markets that offer products that meet the user's specification. In order to determine those machines that fit the specification, the buyer agent examines the sites of a number of computer manufacturers to determine the latest specification information and to determine an approximate price to pay. Armed with this information, the agent formulates a strategy for making a deal. The agent knows the maximum price it needs to pay (this will be the minimum of the cheapest fixed price offerings that are available in the catalogues). From this baseline, the agent tries to negotiate directly with several of the suppliers to see if they are willing to reduce the price (or bring forward the delivery time). In parallel to this, the agent tracks a number of online auctions to see if the same good can be purchased more cheaply (it will not actually bid in the auctions, since submitting a bid would constitute a commitment on behalf of its buyer). When it has completed its negotiations (or before if a very good deal appears in an auction), the buyer agent reports back a ranked list of purchasing options to its owner. The owner then makes their choice and instructs their agent to complete the deal (including arranging payment and setting the delivery time and place).

Scenario 3: Coalition formation. A bakery agent receives a request for tender from a supermarket agent who wishes to purchase 500 iced buns a day throughout the summer period. The bakery agent has sufficient capacity to make 300 buns per day. However, the bakery would like to set up links with the supermarket and so is keen to see if it can fulfil the order. Thus, rather than simply turning the order away, the bakery instructs its agent to search for a partner who will produce the remaining 200 buns for the rest of the summer period. In order to achieve this, the bakery agent contacts all the other sellers present in e-markets that offer iced buns. The bakery agent indicates it has a demand for 200 buns per day for the summer period and asks whether any of the other bakeries would like to join in a partnership with it to meet the supermarket's need. A number of potential collaborators come forward. The bakery agent then conducts a series of negotiations with these agents in order to set up the terms and conditions of the partnership. Eventually a deal is reached and the bakery agent reports details of the arrangements back to the bakery.

In terms of the nomenclature outlined in Section 1.1 scenarios 1 and 2 fall into the B2C domain. The former shows that agents can, on behalf of their owners, locate and retrieve information and make reasonable decisions (relaxing the constraints of the search)

• Find e-markets where desired good can be purchased. • Find relevant purchasing information from identified e-markets. • Rank identified goods and e-markets. • Select e-market(s) where will attempt to purchase good. • Determine precise conditions for purchasing the good in chosen e-market (this may range from obtaining the fixed price, through participating in an auction, to direct negotiation). • Decide to purchase the good. • Pay for the good. • Arrange delivery.	No automation: all activities are manual ↓ Increasing automation Complete automation: all activities performed by a software agent

Figure 2.1: Varying degree of agent automation.

based on the owner's profile. The latter scenario demonstrates how agents negotiate with multiple suppliers, monitor multiple auctions, and use intelligent strategies to find the best deal for the users. The agents in the third scenario represent companies/organisations in a B2B context. This example not only shows how agents can collaborate with one another to achieve a common goal, but also shows how an agent selects the best partners through negotiation.

Although agents can be used in a closed loop fashion (*i.e.*, without human intervention), in many cases users will simply not be willing to delegate complete autonomy to them. Moreover, the degree of automation that user's find acceptable is likely to vary between individuals and between tasks for the same individual. For example, some users will not want any automated support—they will directly enact all phases of the trade themselves. Others may be willing to use agents to collate information and present them with options from which they make the subsequent purchasing decision, while yet others will be happy to delegate all trading activities to their software agents. To reflect this situation, Figure 2.1 shows the range of the automation that a software agent may be given.

2.2 Agents in B2C E-Commerce

According to the nature of the transactions, the following types of e-commerce are distinguished [Turban et al., 1999]: business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer (C2C), consumer-to-business (C2B), nonbusiness e-commerce (use of the Internet by nonbusiness organisations such as academic institutions or government agencies to reduce expenses or improve services), and intra-business e-commerce. Currently, however, most applications are either B2C or B2B and, therefore, these are the two areas that we focus on here. In more detail, B2C mainly refers to online retailing transactions with individual customers, where shoppers can conduct transactions through a company's homepage. B2B refers to the transactions where both sellers and buyers are business corporations. Although most of the initial web-based e-commerce was in the B2C domain, B2C now constitutes a smaller portion of the overall landscape. For example, B2B transactions are expected to be in the range of $800 billion by 2003, which is five times as much as B2C [Sharma, 2002]. Moreover it is widely believed that B2B will be the predominant means of doing business within the next five years [Subramani and Walden, 2000, Shaw, 2000].

B2C e-commerce is becoming more widespread as more people come to recognise its convenience and its ability to offer a quick response to requests and as more products/services become available [Murch and Johnson, 1999]. As this adoption spreads, the impetus for employing software agents increases in order to enhance and improve the trading experience. In order to systematically analyse the tasks that agents can assist with, we employ the CBB model (based on [Guttman et al., 1998]) to capture consumer behaviour (see Figure 2.2). From the CBB model perspective, we believe agents can act as mediators in five of the stages: need identification, product brokering, buyer coalition formation, merchant brokering and negotiation.[3] Sometimes, the boundary between merchant brokering and negotiation is not always clear cut (because negotiation is sometimes also involved in brokering). For example, [Jung and Jo, 2000] introduce a brokering technique that uses a negotiation protocol to match seller and buyer agents; in the brokering service of [Bichler and Kaukal, 1999], a multi-attribute auction is proposed to find a suitable supplier for a buyer; and in [Easwaran and Pitt, 2000] the brokering service involves finding the optimal winner through a combinatorial auction. Against this background, each of the five above-mentioned agent mediated stages is explored in more detail in the remainder of this section.

2.2.1 Need Identification

In this stage, the customer recognises a need for some product or service. This need can be stimulated in many different ways (*e.g.*, by advertisement, through friends, and so on). However in the agent mediated e-commerce world it can also be stimulated by the user's

[3] The sixth stage (purchase and delivery) involves paying for the transaction and arranging delivery of the goods/services. Here the key problems are to ensure safe payment and delivery, problems that are common to e-commerce in general. The last stage involves product services (*e.g.*, repair and upgrade services) and evaluation (measuring the degree of satisfaction of the user about the goods and the buying procedure). Generally speaking, however, these two stages have little that is specific to agent mediated e-commerce and thus they are not discussed in detail here (interested readers can refer to [Garfinkel et al., 2001, Turban et al., 1999, McDermott, 2000, Standifird, 2001] for more information on these topics).

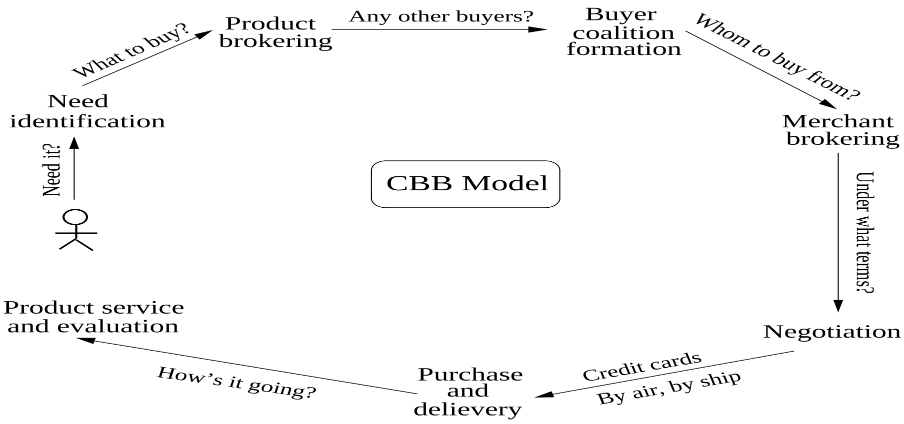


Figure 2.2: Consumer behaviour buying model.

agent. Such an agent is typically called a notification agent.[4] To do this, the notification agent needs to have a profile for the user. This profile can be obtained in many different ways: through observing the user's behaviour [Billsus and Pazzani, 1998], through direct elicitation techniques [Ribeiro, 1996] or through inductive logic programming techniques [Dastani et al., 2001]. Once the profile is installed in the agent, it can notify the user whenever an appropriate good/service becomes available (*i.e.*, the user's profile matches a good/service catalogue). For example, in Amazon Delivers (http://www.amazon.com), the latest reviews of exceptional new titles in categories that interest the user are sent automatically and Fastparts uses "AutoWatch" to allow users to list parts they need and notify them if those parts become available for sale.

2.2.2 Product Brokering

Having ascertained a need, the product brokering stage involves an agent determining what product to buy to satisfy this need. The main techniques used by the brokers in this stage are: feature-based filtering, collaborative filtering, and constraint-based filtering. Table 2.1 shows a number of exemplar e-commerce systems that exploit these techniques. Feature-based filtering involves selecting products based on feature keywords. For example, suppose a customer wants to buy a Sony notebook computer through Amazon. His agent selects the "Computers" category first, then indicates "Sony" in the *brands* field, and the notebook computers with these features are returned. Collaborative filtering [Shardanand and Maes, 1995] involves giving an agent personalised recommendations based on the similarities between different users' preference profiles. Here the product rating of shopper *A* is first compared with that of all the other shoppers in the system. Then, the "nearest neighbour" of *A* (*i.e.*, the shopper whose profile is closest to that of *A*) is identified. Since shoppers with similar tastes and preferences are likely to

[4]This is a reasonably simple type of agent. It acts autonomously to inform the user of relevant information, it responds to changes in the environment and occasionally it is proactive in that it may inform the user of information that is not exactly what had been asked for but is judged to be sufficiently interesting to warrant informing the user.

Table 2.1: Filtering techniques for product brokering in e-commerce systems.

	Feature-based	Collaborative	Constraint-based
Amazon	√		√
eBay	√		√
CDNOW	√	√	
Yahoo shopping	√		√
Net Perceptions		√	

Table 2.2: Comparisons of different product brokering techniques.

	Feature-based	Collaborative	Constraint-based
When to use the technique	User's needs known	User's needs unknown	Some idea of user's needs
Requirements	Feature keywords for goods	Profiles of users	Conditions that goods satisfy
Interaction with user	Medium	Few	Medium
Results returned	Goods satisfying required features	Suggestions of goods to buy	Goods satisfying particular constraints
Goods suitable for	Most goods	Books, CDs, etc.	Most goods

buy similar products, the profile of the identified shopper is used to pass recommendations onto *A*'s agent. For example, in Net Perceptions (http://www.netperceptions.com), users are recommended the documents that their "knowledge neighbours" find valuable. In CDNOW (http://www.cdnow.com), users are notified about the CDs or movies that are popular with other users with similar preferences. Constraint-based filtering involves an agent specifying constraints (*e.g.*, the price range and date limit) to narrow down the products. In this way, customers' agents are guided through a large feature space of the product [Guttman et al., 1998]. For example, eBay guides a user agent to select the products by narrowing down the range of the possibilities based on the constraints the user gives (*e.g.*, price range, item location, and so on). In the end, a list of the desired products that satisfy the user's constraints is returned. Some e-commerce systems use more than one kind of filtering technique (since sometimes users do not know exactly the constraints of the goods they are looking for in advance). For example, eBay and Yahoo Shopping (http://shopping.yahoo.com) use both feature-based and constraint-based techniques. The differences among these techniques are summarised in Table 2.2.[5]

[5] Most of the dimensions in Table 2.2 are self explanatory. However, for "Interaction with user", few interactions are needed in collaborative filtering, since what the user agents need to do is just provide their user's profile and they can then get recommendations from the system. For feature-based and constraint-based systems, some keywords or constraints need to be input until the user can find the exact product they want. The last dimension in the table is "Goods suitable for". Collaborative filtering is more specialised than the other techniques because it works based on perceived quality and people's tastes rather than objective properties [Shardanand and Maes, 1995]. Thus, it is more suited to goods such as novels, CDs and DVDs because it is subjective judgements that act as the differentiator in these cases.

2.2.3 Buyer Coalition Formation

Having determined the product to buy, customers may move directly to the merchant brokering phase (see below) or they may interact with other similar buyers to try and form a coalition before moving to the merchant brokering phase. Here a coalition is viewed as a group of agents cooperating with each other in order to achieve a common task [Shehory and Kraus, 1998]. In these "buyer coalitions", each buyer is represented by their own agent and together these agents try and form a grouping in order to approach the merchant with a larger order (in order to obtain leverage by buying in bulk). In [Yamamoto and Sycara, 2001], for example, a buyer coalition formation scheme is proposed in which buyer agents specify multiple items in a category and their valuation of these items and the group leader agent is then responsible for dividing the group into coalitions and calculating the surplus division among the buyers. Similarly, [Tsvetovat and Sycara, 2000] views a buyer coalition model as being composed of five stages: negotiation, leader election, coalition formation, payment collection and execution stages. They test their algorithms in a collective book purchasing setting in the university and show how the supplier agent gives a volume discount according to the size of the coalitions. In both of the above systems, it is essential to have a trustworthy and reliable agent that will collect the buyer's information, divide the agents into coalitions, and negotiate with sellers (refer to [Yamamoto and Sycara, 2001, Tsvetovat and Sycara, 2000] for a full discussion of these issues).

2.2.4 Merchant Brokering

Having selected the desired product, and perhaps after having formed a buyer coalition, merchant brokering involves the agent finding an appropriate merchant to purchase the item from. Initial work in this area focused on finding the merchant that offered the good at the cheapest price. BargainFinder [Krulwich, 1996] was the first system of this kind to employ agents and it operated in the following way. If a customer wants to buy a music CD, BargainFinder will launch its agent to collect the prices from a pre-defined set of CD shops, and then it will select the CD with the lowest price for the customer. Another similar example is Priceline (http://www.priceline.com) which carries out the same set of tasks for airline tickets, hotel rooms and cars.

However, in many cases price is not the only determinant for the user. Other relevant issues, for example, might include delivery time, warranty and gift services. Also many merchants prefer their offerings not be judged on price alone. Thus there is a move to extend these agents to consider multiple attributes. Naturally the importance of the different attributes will vary between consumers and so there needs to be a way for this information to be easily conveyed to the agent. In the Frictionless Sourcing (http://www.frictionless.com) platform, "Vendor Scorecards" (multi-attribute comparisons) are used to measure the performance of suppliers. For example, when evaluating the performance of different laptop computer suppliers, the key factors considered include reliability, responsiveness (*e.g.*, reacting quickly), environmental friendliness (*e.g.*, minimal pollution of the environment), and business efficiency (*e.g.*, support for electronic purchasing over Internet). A total score is then calculated for each supplier based on the weighted score of these individual constituent components. These weights are obtained by the customer identifying themselves with a particular stereotype profile in which the weights are given.

2.2.5 Negotiation

Having selected a merchant (or set of merchants), the next step is to negotiate the terms and conditions under which the desired product will be delivered. To this end, we believe that one of the major changes that will be brought about by agent mediated e-commerce is that dynamic pricing and personalisation of offers will become the norm for many goods and customers. Thus, negotiation capabilities are essential for e-commerce systems [Beam and Segev, 1996]. In human negotiations, two or more parties bargain with one another to determine the price or other transaction terms [Fisher and Ury, 1981]. In an automated negotiation, software agents engage in broadly similar processes to achieve the same end [Jennings et al., 2001]. In contrast to many human negotiations, automated negotiation can be very fast since decisions and exchanges can occur rapidly. In some cases, negotiation is very complicated (*e.g.*, when it involves many interrelated goods) and can become too difficult for consumers to handle manually. In such cases, automated negotiation systems can help ordinary users perform like professional negotiators. Moreover, automated negotiation can also remove the human sensibilities that are often associated with negotiating.

In more detail, the agents prepare bids for and evaluate offers on behalf of the parties they represent with the aim of obtaining the maximum benefit for their users. They do so according to some negotiation strategy. As discussed in Section 1.1.3, such strategies are determined by the negotiation protocol that is in place. Given the wide variety of possibilities (as will be shown below), there is no universally best approach or technique for automated negotiations [Jennings et al., 2001], rather protocols and strategies need to be set according to the prevailing situation [Friedman, 1993]. Given this, our analysis of automated negotiation models as used in B2C e-commerce is divided into two categories:[6] auctions and bilateral negotiations.

Auctions

As discussed in Section 1.1.2, there are many types of auctions. However, most e-commerce scenarios concentrate on the basic four types of single sided auctions (English,

[6] There are other types of negotiation protocol such as multi-lateral negotiation (in which the negotiation involves bargaining between multiple non-cooperative parties [Adams et al., 1996]) and n-bilateral negotiations (in which the negotiation involves multiple bilateral bargaining encounters [Faratin, 2000]). However since these protocols are not as widely used in e-commerce we do not consider them here.

Table 2.3: Values of each dimension in Table 2.4.

Auction mode	one-sided (O) two-sided (T)	only bids or asks are permitted both bids and asks are permitted
Duration time	single-round (S) multi-round (M)	the auction only lasts one round the auction lasts multiple rounds
Unit of goods	one (O) many (M)	only one good is auctioned during the auction multiple goods are auctioned
Ratio of B-S	many to one (MO) one to many (OM) many to many (MM)	there are multiple buyers and only one seller there are multiple sellers and only one buyer there are multiple buyers and multiple sellers
Information revealed	yes (Y) no (N)	there is intermediate information revealed a bidder has no information about others
Closing price	first price (F) second price (S) different prices (D)	highest price among all the bidders second highest price among all the bidders trades take place any time at different prices
Closing rules	time (T) inactivity (I) budget (B)	when a time is reached when there are no more bids for a time period when a reserve price is reached

Table 2.4: Comparison of different types of auctions.

Auction	auction mode		duration time		unit goods		ratio of B-S			information revealed		closing price	closing rules
	O	T	S	M	O	M	MO	OM	MM	Y	N		
English	√			√		√	√	√		√		F	I
FPSB	√		√			√	√	√			√	F	T
Vickrey	√		√			√	√	√			√	S	T
Dutch	√			√		√	√	√		√		F	B
CDA		√		√		√			√	√		D	I

FPSB, Vickrey and Dutch) or the continuous double auctions (CDA). Thus, we compare these auction protocols according to the auction mode, duration time, unit of goods auctioned, ratio of buyer to seller, how much information is revealed during the auction, how the closing price is determined, and when the auction closes. See Table 2.4 for the detailed comparison and Table 2.3.[7] for the explanation of the dimensions in the table. Given these protocols, we now turn to the strategies the agents need to employ in order to be successful.

- *English auction*. The agent's dominant strategy (the best thing to do, irrespective of what the others do [Sandholm, 1999b]) is to bid a small amount more than the current highest bid and stop when the user's valuation is reached. For example, in Yahoo auctions, "automatic bidding" allows buyers to input their maximum bid

[7]Here we only consider the popular forms of the auctions on same type, non-divisible goods and the quantity is a single unit (see [Wurman, 2001] for a fuller description of the bidding rules). Also, many different types of information can be revealed in the course of the auction (*e.g.*, the identity of the bidders, the settlement price, the ask-bid spread and so on [Friedman, 1993]) here, however, we only consider whether any intermediate information is revealed (see [Wurman et al., 1998] for a detailed discussion of the impact of information revelation).

(*i.e.*, valuation of the item) and an agent will bid incrementally when it is necessary to win the auction.

- *First-price sealed-bid auction (FPSB)*. In general, there is no dominant bidding strategy in this auction. Here the price of the bid and the time to stop bidding are functions of the agent's own valuation of the item and its beliefs about the valuation of others' bidders. A good strategy is to bid less than the user's true valuation, but how much less depends on the user's attitude toward risk, the user's private valuation, as well as the prior beliefs about the valuations of other bidders. An analysis of such strategies can be found in [McAfee and McMillan, 1987].
- *Vickrey auction*. In a private value[8] Vickrey auction, the dominant strategy is to bid the user's true valuation [Sandholm, 1999b]. In this context, agents truthfully reveal their preferences which allows efficient decisions to be made.
- *Dutch auction*. The Dutch auction is strategically equivalent to the first-price sealed-bid auction. This is because in both games an agent's bid matters only if it is the highest, and no relevant information is revealed during the auction process [Sandholm, 1999b]. Klik-Klok (http://www.klik-klok.com) is an example of a Dutch auction website for gold and jewellery sale where auction prices decline until a buyer makes a bid. The analysis of strategies in Dutch auctions can be found in [Milgrom, 1989].
- *Continuous Double Auction (CDA) (to be discussed in more detail in Chapter 3)*. A variety of different CDA models have been constructed [Easley and Ledyard, 1993, Gjerstad and Dickhaut, 1998, Sadrieh, 1998] and these vary in terms of whether bids/asks are for multiple or single units, whether unaccepted offers are queued or replaced by better offers, and so on [Friedman, 1993]. Nevertheless all these protocols allow traders to make offers to buy or sell and to accept other traders' offers at any moment during a trading period [Friedman, 1993]. The messages exchanged generally consist of bids (offers to buy) and asks (offers to sell) for single units of the commodity, and acceptances of the current best bid or ask. Several bidding strategies have been proposed in the literature and these are discussed in more detail in Section 3.5.

In addition to the aforementioned auction type, and as argued in Section 1.1.3, we believe that the multiple auction context will become increasingly important in the domain of e-commerce. In this type of auction, the agent needs to monitor all the relevant auctions, decide which one to bid in and determine what to bid in order to get the goods at the best deal. This is an aspect of e-commerce that is only really made possible by the

[8] Private value means that the bidder only knows its own valuation of the item, but it does not know the value to other bidders [McAfee and McMillan, 1987]. For example, auctioning a cake (that the winner will eat) is a private value auction, whereas auctioning treasury bills (where the value of the bills comes from reselling) is a public value auction.

use of agents in a timely manner would be beyond most humans. See Section 4.4 for a more detailed analysis of the state of the art in this area.

Bilateral Negotiations

Bilateral negotiation involves two parties, a service/good supplier and a consumer, coming to a mutually acceptable agreement over the terms and conditions of a potential transaction [Sierra et al., 1997a]. In contrast to most of the auction work (which is a form of one to many, many to many, or many to one negotiation), bilateral negotiation is usually concerned with multi-attribute contracts (covering price, quality, delivery date, and so on). As with the auction work, there is no dominant negotiation model or strategy that is suitable for all occasions. Rather, it is a case of different models having different strategies that are suitable in different contexts. Given this situation, we classify extant work on bilateral negotiation into three groups.

- *Decision making by explicitly reasoning about the opponent's behaviour.* Agents using the strategies in this group explicitly reason about their opponent's objectives and behaviours and then decide what is the appropriate response to their likely behaviour. In this respect, non-cooperative game theory (which is particularly concerned with providing equilibrium strategies in which no agent wants to change its strategy whatever its opponents do) is an important approach for analysing strategic interactions among agents [Kreps, 1990, Tirole, 1998]. The recursive modelling method [Vidal and Durfee, 1996, Gmytrasiewicz and Durfee, 1995] is employed by an agent to reason about its opponent so that it can generate its own strategy in response. In [Zeng and Sycara, 1998], a Bayesian network is used to update the knowledge and belief that each agent has about the environment and other agents; and offers and counter-offers between agents are generated based on Bayesian probabilities. More discussion about these strategies can be found in Section 3.5.
- *Decision making by finding the current best solution*. Algorithms in this group focus on finding the offer/counter-offer that maximises the agent's profit given the agent's constraints, preferences, current negotiation situation, and the opponent's last offer. In Tête-à-Tête [Guttman and Maes, 1998], constraints on product features and constraints on merchant features are used to influence the decision of what and whom to buy from. [Luo et al., 2003c] developed a fuzzy constraint based framework for multi-issue negotiations in competitive trading environments and demonstrated it in a negotiation between a real estate agency and a customer. [Kowalczyk and Bui, 2000, Kowalczyk, 2000] also use fuzzy constraints to model multi-issue negotiation, but their approach performs negotiation on individual solutions one at a time. Matos and Sierra [Matos et al., 1998] employ fuzzy logic, case-based reasoning and evolutionary computing to deal with the bilateral negotiation

[Jennings et al., 2000a]. Faratin *et al.* develop a suite of algorithms for multi-issue negotiation that covers both concessionary behaviour [Faratin et al., 1998] and trade-offs that aim to find a win-win solution for both parties [Faratin et al., 2002].

- *Argumentation*. In this approach, agents exchange additional information over and above the basic terms and conditions of the contract [Jennings et al., 2001]. This information can be of a number of different forms, nevertheless, it is always some form of argument which explains/justifies the position of the agent making the argument. Thus, in addition to rejecting a proposal, an agent can offer a critique of the proposal, explaining why it is unacceptable (*e.g.*, the price is too high). The way in which argumentation fits into the general negotiation process was defined in [Sierra et al., 1997b] where a simple negotiation protocol for trading proposals was augmented with a series of illocutionary moves which allow for the passing of arguments.

2.3 Agents in B2B E-Commerce

Compared with B2C e-commerce, B2B deals with transactions among organisations (see Table 2.5 for a more detailed comparison). Generally speaking, relationships between organisations are more complex than those between businesses and consumers, since they involve the adoption of similar standards with respect to communications and collaboration, as well as joint information technology investment [Subramani and Walden, 2000]. In particular, one of the main aims of B2B e-commerce is to significantly improve the supply chain by facilitating more efficient and agile procurement [Dou and Chou, 2002]. Moreover the exchanges in B2B are increasingly tending to be private [Young, 2001]. Such exchanges enable companies to trade with their existing partners in a well defined environment without having to go through some of the early stages of the B2B lifecycle (see Figure 2.3).

Here we create a BBT model (see Figure 2.3) to explore the roles of agents because other models (*e.g.*, [Barnes-Vieyra and Claycomb, 2001, Turban et al., 1999]) cannot cover all the phases involved in the Internet based B2B e-commerce taking place today. Specifically, we believe agents are most useful in the partnership formation, brokering and negotiation stages because these stages all involve complex issues related to decision making, searching and matchmaking that agents are well suited to. Thus we will explore these roles in more detail in the rest of this section. Currently, agents are not used in the contract formation stage, but we believe they have the potential to be involved in this activity. Contract formation marks the termination of negotiation and involves the agreed terms being put into a legally binding contract. Traditionally contract formation involves two or more people, meeting face-to-face. However as e-commerce systems evolve this situation is starting to change. In the U.S., for example, Section 206

Table 2.5: Dimensions of B2C and B2B e-commerce.

Dimension	Business-to-customer e-commerce
Features	• Individual customer oriented • No collaboration between customer and company required • Brings convenience for buying in globally competitive markets • Quick response to the transaction • Convenient to use
Roles of agents	☐ Need identification • *e.g.*, Amazon (Delivers) and Fastparts (Auto Watch)
	☐ Product brokering • Feature-based, collaborative and constraint-based filtering • *e.g.*, Amazon, eBay, and Net Perceptions
	☐ Buyer coalition formation • *e.g.*, Collective book purchasing and GroupBuyAuction
	☐ Merchant brokering • Price comparison and multi-attribute comparison • *e.g.*, BargainFinder and Frictionless Commerce
	☐ Negotiation • Auctions and multi-attribute bilateral negotiation • *e.g.*, eBay, Yahoo Auctions and AuctionBot

Dimension	Business-to-business e-commerce
Features	• Organisation oriented • Close collaboration between organisations required • Facilitates both direct and indirect procurement and supply chain • Larger and global markets • Real-time and low cost transaction • Less inventory and dynamic pricing
Roles of agents	☐ Partnership formation • Virtual enterprises • Supply chain management
	☐ Brokering • Information retrieval & processing, negotiation, Profiling of users, notification, collaboration with other brokers •*e.g.*, OFFER, MULTIMEDIATOR, and Abrose
	☐ Negotiation • Auctions (sell-side, buy-side and combinatorial auction) • *e.g.*, Fastparts.com, GE.com, Ariba.com and labx.com • Contracting • *e.g.*, Contract net, marginal cost-based contract, OCSM levelled commitment contract and MAGNET

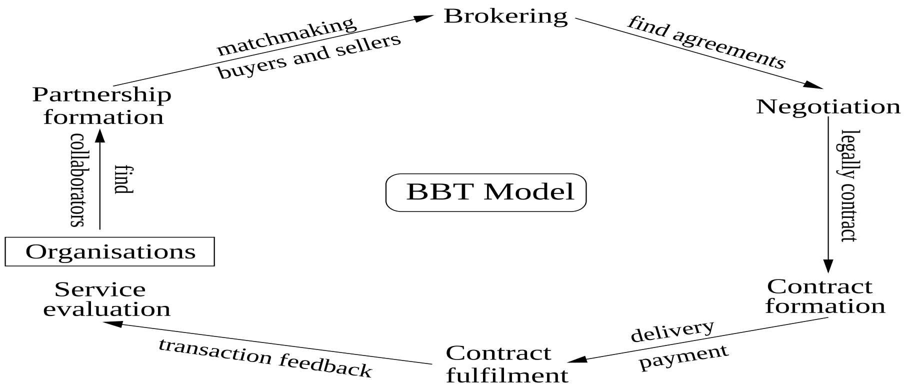


Figure 2.3: Business-to-business lifecycle model.

in the Uniform Commercial Code (U.C.C.) was proposed by reformers as a way of dealing with automated contract formation and clearly states that contracts can be formed by the interaction of electronic agents.[9] This proposal is motivated by the fact that it is a challenging task for the courts to determine where the communication system ends and when the legal agent begins.[10] The fifth stage, contract fulfilment, means the parties carry out the agreed transaction according to the terms specified in the contract. This stage usually includes: a detailed description of the good/service provided; the means of delivery (electronic or physical); how it will be paid for (*e.g.* partial payments up-front, with the balance paid on completion); which law governs the contract; how to resolve any disputes, how to deal with claims arising, how a contract can be monitored, and so on. We believe agents are not likely to be involved in this stage for some time, because it involves many complex legal issues and subjective judgements. The last stage, service evaluation, is the post-transaction stage, where traders evaluate their satisfaction with the transaction. Many e-commerce systems allow users to provide feedback on the transactions experienced. For example, eBay uses “Feedback Forum” to check the reputation or business practices of anyone at eBay. This feedback, representing the reputation of the trader, can then be made accessible to subsequent agents that wish to interact with the trader. Again because of its subjective nature, we do not believe there is a significant role for agents in this phase of the lifecycle.

2.3.1 Partnership Formation

The information technology available today makes it possible for a company to search for its partners worldwide [Kumar, 2001]. Given this fact, partnerships can be much more agile and fluid. Thus this step may include the forming of a new virtual organisation as well as finding the partners that provide products or services in a supply chain.

[9] The official draft of Article 2B of the U.C.C. is from 2002 and can be found at http://www.law.uh.edu/ucc2b.

[10] See http://www.jurisdiction.com/ecom3.htm for more details of this debate.

Virtual Enterprises

A virtual enterprise[11] (VE) is composed of a number of cooperating companies that share their resources and skills to support a particular product or project effort (for as long as it is viable to do so) [O'Leary et al., 1997]. The idea is that by collaborating the constituent companies can more effectively utilise their resources than if they acted in isolation [Goldman et al., 1995]. For example, an individual company may collaborate with several partner companies that provide related products so that each of them need only provide the services/products in which they specialise, but, when taken together, the VE can provide a broader range of offerings. Such VEs offer several potential advantages [Martinez et al., 2001]: maximising flexibility and adaptability to respond to environmental changes; developing a pool of competencies and resources by combining its members' resources; adjusting itself according to the market constraints; and managing the global supply chain optimally.

Given the fact that a VE is composed of a number of autonomous entities that need to interact with one another in flexible ways, agent technology is a natural underpinning model [O'Leary et al., 1997, Norman et al., 2004]. In more detail, the formation of a VE involves a selection process based on a number of variables such as organisational fit, technological capabilities, relationship development, quality, price and speed [Sarkis and Sundarraj, 2002]. Thus, a broker may assist in identifying the best partners from the set of potential collaborators [Meade et al., 1996] (see also Section 2.3.2). Having identified the partners, the agents need to negotiate with one another in order to set the terms and conditions of their partnership [Tuma, 1998] (and Section 2.3.3). Then, once the VE is established, the agents need to coordinate their actions so that they deliver their services in an effective manner. Here the VE might require a number of agents to manage its ongoing operations [Massotte, 1993]. For example, [Martinez et al., 2001] propose a multi-agent control system that consists of three kinds of controller agent: product agents (which manage the activity associated with each product), activity agents (which autonomously manage an entire manufacturing activity) and resource agents (which manage their own operative functions and propose service offers to activity agents). Together these agents use and control the other entities in the system in order to achieve the VE's overall aims. The MASSYVE (multi-agent agile manufacturing scheduling systems for virtual enterprises) project focuses on the use of multi-agent systems in agile scheduling in a VE environment [Rabelo et al., 1999]. The factors considered here range from distribution logistics scheduling in supply chains to negotiation in the VE using mobile agents. The AIMS (agile infrastructure for manufacturing systems) project enables companies to share resources and skills to facilitate the operations of VEs and agents function as a bridge between clients and servers [O'Leary et al., 1997]. Specifically, the agents act

[11] Coalition formation by buyer companies is similar to buyer coalition in the B2C domain (Section 2.2.3). Thus here we focus on virtual enterprises of supplier agents.

as: facilitators (routing requests to appropriate databases); aggregator agents (combining multiple orders); user programmable agents (automating routine tasks) and engineering databases agents (notifying users of design changes).

Supply Chain Management

A supply chain is formed by business units or facilities that purchase raw materials, convert them into intermediate goods and final products, and delivers these final products to customers [Tan et al., 2000]. A supply chain is used to coordinate the activities of the organisations involved in order to ensure that products pass through the chain in the shortest time and at the lowest cost [Lee and Billington, 1995]. Because of the business trend towards outsourcing services and resources, supply chain networks have become more complex [Kumar, 2001]. Given this, the software solutions being developed need to be more sophisticated than the current generation of workflow tools. In particular, the various components of the supply chain can be viewed as autonomous stakeholders and these various stakeholders need to interact in flexible ways. Thus an agent-based approach is well suited to this domain [Huhns et al., 2002, Jennings et al., 2000a]. In particular, agents can be used to execute the scheduling [Fox et al., 2000], negotiate about product prices [Sun et al., 1999] and share data between companies [Zeng, 2001].

To this end, a number of models for agent-based supply chain management have been reported. For example, Walsh and Wellman developed a market system, based on a task dependency network, for allocating tasks among agents that compete for scare resources [Walsh and Wellman, 1999]. [Sun et al., 1999] model and implement the order selection and negotiation process in a supply chain as a multi-agent system. Here a negotiating agent represents each company along the supply chain and agents generate a purchase plan, negotiate and generate counter proposals using constraint satisfaction techniques. Moreover, Zeng proposed the Leadtime-Cost Tradeoff Supply Chain Model [Zeng, 2001] in which each agent represents a business entity and the agents coordinate with each other to control activity in the supply chain. An agent-based approach for streamlining the business decision process is proposed in [Keskinocak et al., 2001]; here agents assist the decision maker by discovering matches between supply and demand.

2.3.2 Brokering

Brokering is the process that matches sellers who supply goods/services to the buyers who need them [Foss, 1998]. From the seller's side, it is how they can propagate their products and locate potential buyers. From the buyer's side, the problem is how to find the most appropriate seller to provide the good/services [Khosla and Kitjongthawonkul, 2001] (*e.g.*, lowest price or best quality). In contrast to merchant brokering in the B2C domain (Section 2.2.4), brokering in a B2B context typically involves repeated transactions and large volumes (in the B2C context brokering requests often tend to be one-off transactions, since individual customers tend not to buy the same product often).

Table 2.6: Brokers' functions and services.
The projects in the table are Abrose [Gleizes et al., 1999], OFFER [Bichler and Segev, 1999], MULTIMEDIATOR [Gallego et al., 1998] and Schmid *et al.* [Schmid and Lindemann, 1998].

Project/ Papers	Information retrieval and processing	Maintenance self-learning about user	Profiling of users	Nego-tiation	Collabo-ration	Notifi-cation
Abrose	√	√	√		√	
OFFER	√		√	√		√
MULTIMEDIATOR	√	√	√			
Schmid *et al.*	√			√		

As the Internet expands, it is becoming more expensive and more difficult to navigate in order to find the necessary information on companies and their offerings (this is especially true for small and medium size enterprises [Turban et al., 1999]). For example, it is estimated that about $5 trillion dollars is spent on the procurement of industrial parts each year [Tully, 2000]. Given the difficulty and value of this exercise, a common way of obtaining this information for companies in B2B e-commerce is through some form of information broker (they can also be called matchmakers [Ha and Park, 2001] or brokerage centres [Gamvroulas et al., 2000]) that acts as an intermediary between the buyers and sellers. Here a broker can be an agent or a multi-agent system. The functions offered by a broker may include the following [Foss, 1998]: information retrieval and processing; maintenance of a self-learning information repository about the user; profiling of users; monitoring for items of interest to the user; filtering of information; intelligent prediction of user requirements; commercial negotiation between customers and the providers; collaboration with other brokers; and protecting the user from intrusive access.

To summarise, Table 2.6 details the services provided by a number of agent-based broker systems. As shown in the table, most brokerage services today mainly focus on information search and matchmaking buyer's and seller's profiles, as well as comparing the products in the catalogues of different suppliers. We believe more advanced services (*e.g.*, collaboration with other brokers and protecting the user from intrusive access) will now start to emerge in order to provide more support to the buyers and sellers involved in the transactions.

2.3.3 Negotiation

After the appropriate providers and consumers have been brokered, the negotiation stage is where the traders aim to reach an agreement about what actions should be performed under what conditions. By establishing contracts on an as-needed, just-in-time basis, sellers can tailor their offerings both to their individual and the prevailing market situation at any given moment in time. Buyers can reduce their supply chain cost, benefit from dynamic pricing mechanisms, broaden their supplier database, and streamline the procurement process. Compared with negotiation in the B2C context, B2B negotiation is more complex. Typically, for example, it involves larger volumes, repeated transactions and more complex contracts. The negotiation methods discussed in the B2C context can also apply here, however, the two most popular means of conducting B2B negotiation are through auctions and contracting.

Auctions in B2B E-commerce

There are now many B2B marketplaces on the web that provide auction services and allow organisations to trade with one another on a global basis, for example, FreeMarkets (http://www.freemarkets.com) and Ariba (http://www.ariba.com). Indeed, industry analysts estimate that 25% of e-commerce now consists of exchanges through such mechanisms [Sashi and O'Leary, 2002]. These auctions offer many advances over traditional exchange methods (*e.g.*, fixed suppliers), such as a larger market, less inventory, reduced transaction costs, global expansion, and efficient pricing [Sashi and O'Leary, 2002]. We classify the commonly used auctions into three kinds: buy-side auctions (one buyer and multiple sellers); sell-side auctions (one seller and multiple buyers) and combinatorial auctions [Fujishima et al., 1999, Karp, 1972, Rassenti et al., 1982] (where bidders bid for a combination of related items). An agent can be either a buyer who submits bids or a seller who provides some products or services in these auctions. The sell-side auction is similar to the auctions discussed in the B2C context; the buy-side auction is the opposite of the sell-side auction (however, it can also be an English, Vickrey, FPSB, or Dutch auction). Combinatorial auctions only take place in B2B environments because of their inherent complexities.

Buy-side Auctions

Buy-side auctions, also called reverse auctions [Teich et al., 1999] or procurement auctions [Che, 1993], occur when buyers negotiate with multiple sellers in order to procure a particular good/service.[12] In this case, the negotiation usually involves multiple attributes, since buyers invariably have their particular requirements on the goods they need. Here the buyer sends out his requirements and the sellers who can meet them make bids. To make this process cost effective, some companies have built their own markets in which they can invite bids from potential sellers. Examples of this kind include General Electric (http://www.ge.com) and Boeing Inc. (http://www.boeing.com). The idea is that the cost spent in searching and comparing suppliers can be significantly reduced, because the companies repeatedly buy large volumes of many such products. In contrast, some companies conduct buy-side auctions through a third party website (*e.g.*, labx and Ariba).

[12] There is another form of buy side auction called a Request for Quote (RFQ) [Turban et al., 1999]. In a RFQ, the buyer requests quotes that can include the price, delivery dates and description of the goods or services being provided. The buyer uses this as a way to begin negotiation. However, since there is not an automatic criteria (*e.g.*, a scoring function to evaluate the bids) for selecting the winner, the strategy for the bidder is not obvious. Thus, we only discuss the reverse auction in this context because it has a clear selection criteria which means it is amenable to an agent-based solution.

More specifically, [Che, 1993] investigates government procurement using a two-dimensional auction (price and quality). A buyer solicits bids from multiple sellers. Each bidder submits a sealed bid specifying the price and quality and the bidder with the highest score wins. Based on the different ways in which the winner offers the goods/services, three auction schemes are proposed: first score (winner offers the price and quality it bids), second score (winner offers the goods/services matching the score of the second highest scored bidder), and second preferred offer (winner offers the goods or services at the same price and quality as the second highest scored bidder). In this model, the buyer evaluates the bids by a scoring function which converts a bid into a single number.

[Bichler et al., 1999] also defines a bidding procedure for multi-attribute auctions: a buyer first specifies a request for bids and defines his requirements and preferences for the goods/services in a scoring function. Then sellers submit their bids. After the auction closes, the wining bid is the one that has the highest score as computed by the buyer (*i.e.*, the seller who satisfies the buyer the most). This basic mechanism was applied to English, Vickrey, and FPSB auction protocols. Moreover, Bichler empirically analysed these multi-attribute auctions and found that the utility scores achieved by the buyer are significantly higher than those of the corresponding single-attribute auctions [Bichler, 2000]. In this setting, the scoring function is revealed to the bidders, thus the bidders know how to improve their bids in a way that makes them most attractive to the buyer, and least costly for them.

[Teich et al., 1999] developed multiple-attribute algorithms and heuristics for auctions. In the case where the “quantity” is not an attribute in the auction, the preference of the auctioneer is represented by the preference path that is the ordering of all the levels of each attribute. The preference path here acts as a scoring function that the bidders would follow. In the case where “quantity” is an attribute in the auction, a discriminative auction algorithm is proposed. The auctioneer can specify multiple reservation prices for different quantities and the bidders can accept the suggested bid or bid above the suggested bid price. The authors argue that the algorithm can make the market more efficient.

[Vulkan and Jennings, 2000] proposed a multi-attribute auction protocol for service allocation in the ADEPT [Jennings et al., 2000a] scenario. The ADEPT technology was used to develop a system for managing the British Telecom (BT) business process of providing a quotation for designing a network to provide particular network services to a customer. They show that the protocol is guaranteed to choose the service provider that makes the best offer from the buyer’s utility respective and that this offer is better than any offer that would have been forthcoming using any other negotiation protocol.

Sell-side Auctions

In this kind of auction, there is a seller who wants to sell goods/services and many buyers join the auction. The mechanism is usually one of the common single sided types described in Section 2.2.5 and the strategies described in that section also apply here. These sorts of auctions are often used by companies that hold excess inventory or that buy out-of-date inventory [Sashi and O'Leary, 2002]. Fastparts which sells electronic manufacturing products and Staples (http://www.staples.com) which sells business supplies and services are two of the most prominent examples of this genre.

Combinatorial Auctions

These are a special form of auction in which there are multiple kinds of goods to sell and bidders can bid on combinations of items. For example, a seller may want to sell several kinds of related goods (*e.g.*, licenses in spectrum auctions (http://wireless.fcc.gov/auctions)) and many bidders may have preferences over a combination of items (*e.g.*, bidding on license A and B for $ 300). After the seller receives all the bids, it will decide a non-conflicting allocation among these goods that maximises its revenue. These sorts of auction are involved in many situations in the real world. For example, in the Federal Communications Commission (FCC) spectrum auction (http://www.fcc.gov/wtb/auctions/), bidders placed bids on different combinations of spectrum licences. Between 1994 and January 2002, 38,829 licenses have been auctioned and 21,849 of them have been won through such combinatorial mechanisms. Other examples of combinatorial auctions are for airport time slots [Rassenti et al., 1982], railroad segments [Brewer, 1999], and delivery routes [Caplice, 1996].

These auctions are especially prevalent in a B2B context because companies often want to trade in a variety of interrelated assets. Moreover as different companies value the items or bundles of items differently, allowing them to bid on combinations provides greater flexibility in expressing their needs and enhances the economic efficiency of the market [Vries and Vohra, 2001]. From an agent perspective, the key challenge is that of winner determination (auctioneer selects a set of non-conflicting bids that maximise its revenue and this problem has been shown to be NP-complete [Fujishima et al., 1999]) and a number of algorithms have been developed to achieve this according to various criteria (*e.g.*, anytime algorithms [Fujishima et al., 1999], polynomial algorithms [Dang and Jennings, 2002] and optimal solutions [Sandholm, 2002]). In this context, the bidding agent has to express its preferences on every bundle it is interested in. However, transmitting these preferences to the auctioneer is a difficult task since the bundles in the bids are likely to be very large. To overcome this, some researchers have developed an "oracle" (a program that can compute the bid for each bidder) [Vries and Vohra, 2001] and others have developed a bidding language to encode the preferences of the bidders (*e.g.*, XOR-bids and OR-of-XORs [Sandholm, 2002]).

Contracting

Contracting covers the negotiation involved in reallocating work among agents; it involves one agent trying to contract out some of its tasks to another agent by promising some rewards [Kraus, 2001]. Contracts have been applied in fields such as electricity markets, bandwidth allocation, manufacturing planning and scheduling, and electronic trading of financial instruments [Sandholm, 1999a].

Smith's Contract Net Protocol [Smith, 1980] was the first multi-agent contracting protocol. In this protocol, a manager agent announces a task, receives and evaluates bids from potential contractors, and then awards the task to one of them and finally receives the result from this contractor. Sandholm extends this work to consider marginal cost-based contracts (an agent contracts in/out a task only if it can make a profit doing so) [Sandholm, 1993]. Sandholm's protocol was used as the basis of the TRACONET system which is an automated system for task reallocation among freight companies. Here each agent, representing a company, can take delivery tasks from or give out tasks to other agents. In the original contract only one task can be moved between agents at any one time. However this sometimes led to local optima. To overcome this, several new types of contract were added [Sandholm, 1993]: cluster contracts (exchanging multiple tasks), swap contracts (swapping a task for another task), and multi-agent contracts (more than two agents in the same contract). When taken together, contracts that combine all of the above can be shown to guarantee the optimal allocation through a finite number of contracts. [Andersson and Sandholm, 1998] also devised levelled commitment contracting in which an agent can decommit from contracts by means of paying a monetary penalty to the contracting partner as a way of releasing itself from the contract. [Collins et al., 1998] developed the MAGNET (Multi AGent NEgotiation Testbed) system which takes advantage of an independent market infrastructure and uses it as an intermediary to facilitate the interactions between agents. Compared with other negotiation approaches (*e.g.*, contract net), the fact that there is an explicit intermediary reduces counter speculation by enforcing negotiation rules and verifying the identity of the agents.

2.4 Summary

This chapter has surveyed and analysed the state of the art in agent mediated e-commerce, focusing particularly on the B2C and the B2B context. While agent mediated e-commerce is still very much in its infancy, a number of agent based deployments have already been made. In highlighting these endeavours, we have also tried to outline medium and longer term aspirations for this area. The key observation from this review is the importance of various forms of automated negotiation. Such negotiation enables trades to occur in more open and dynamic environments and enables trading to be significantly more agile than it is at present. Within this space, auctions are a key means of achieving its aim. However, a key impediment to the degrees of automation discussed in this chapter is the lack of effective and efficient bidding strategies for the agents that are to participate in these auctions. To this end, the remainder of this thesis concentrates on developing such strategies for several auction contexts that are important to e-commerce scenarios.

Given this analysis of the state of the art, we decided to concentrate on the strategy design highlighted in Section 2.2.5. As already discussed, this is a key aspect of agent-mediated e-commerce and one that has many different facets. Thus the remainder of this thesis focuses on the issue of designing effective and practical strategies for agents that participate in a variety of different auction settings. In more detail, this thesis first concentrates on developing a strategy for a particular auction setting. To demonstrate the power of the agent-based approach the CDA is chosen (Chapter 3). This is a reasonably common type of auction (see Section 2.2.5), but is sufficiently complex that it has no optimal strategy that can be pre-computed. Having successfully developed an agent for this scenario, we then focus on the more complex multiple auction setting where an agent tries to bid across multiple simultaneous auctions in order to procure a number of goods (Chapters 4 and 5).

Chapter 3

Bidding Strategies for Continuous Double Auctions

In the previous chapter, we surveyed the field of agent mediated e-commerce from the B2C and B2B perspectives, highlighted the importance of auctions and of developing effective bidding strategies for agents that operate in such contexts. Against this background, this chapter focuses on the strategy design for a particular auction protocol—the continuous double auction (see Section 2.2.5 for a detailed definition of a CDA). Our agent design addresses the following common issues discussed in Section 1.2: price prediction (the reference price defined in Section 3.2.1), adaptation (see Section 3.4), risk attitude (see Section 3.3.2) and flexible bidding (see Section 3.2.2).

In more detail, the bidding algorithms we develop are heuristic methods that exploit fuzzy logic techniques [Zadeh, 1965], especially fuzzy rules, to undertake their reasoning. The reason for this choice is that in CDAs there is no optimal bidding strategy [Friedman and Rust, 1992]. This is because an agent's decision making about bidding involves uncertainty, multiple factors and non-determinism that are affected by the attitudes towards risk of its opponents, the nature of the market supply (demand), and the preferences of the other bidders. Since no agent can have all this information in advance (it is, after all, a competitive environment) the best that can be achieved is a satisficing strategy [Simon, 1997]. Given this, we adopt a fuzzy logic based approach because fuzzy techniques have proven to be successful in a wide range of domains with these characteristics (see Section 1.2 for a detailed justification for using fuzzy techniques). The other alternatives we considered in this specific case are discussed in Section 3.5.

The specific contributions of this chapter are as follows. Firstly, we develop a novel fuzzy logic based bidding strategy—the FL strategy—for agents that participate in CDAs. Secondly, we present the design, implementation and evaluation of this strategy for buyer and seller agents. This strategy is shown, via empirical studies, to outperform the main strategies that have previously been proposed for CDAs. Thirdly, we enhance the basic strategy so that it can adapt its behaviour to the supply (demand) in the market (this revised strategy is called the adaptive FL -strategy). We then show how this revised strategy leads to a further improvement in the performance of both the individual agents (buyers and sellers) and of the overall marketplace.

The remainder of this chapter is organised as follows. Section 3.1 formalises a CDA and outlines the basics of our fuzzy reasoning mechanism. Section 3.2 presents the FL-strategy. In Section 3.3, the behaviours of our FL-agents are analysed in a range of experiments. Section 3.4 discusses the adaptive FL-agents and their evaluation. Section 3.5 discusses the related work. Finally, Section 3.6 summaries this chapter.

3.1 Preliminaries

This section outlines the basis of our FL-strategy - presenting a formal account of our CDA protocol and describing the fuzzy reasoning mechanism we employ.

3.1.1 Continuous Double Auctions

According to the parameterisation of CDAs given in [Friedman, 1993], we deal with the situation in which there are multi-unit goods in the market; there are two-way traders (buyers and serllers) and the numbers of buyers and sellers are 6 (3 buyers and 3 sellers); single indivisible units are to be traded (thus at any one time there is 1 outstanding bid and 1 outstanding ask); the preferences of the traders are the reservation prices of the goods; and traders have incomplete information of the market. The CDA terminates after a specified period of inactivity.

In more detail, there are agents that are willing to sell goods (s-agents) and agents that are willing to buy goods (b-agents). A given agent can be either a buyer or a seller in a given context. Specifically, an *ask* $\mathfrak{a}$ is the amount submitted by an s-agent willing to sell a unit of good. The lowest ask in the market is called the *outstanding ask*, denoted $\mathfrak{a}_o$. Similarly, a *bid* $\mathfrak{b}$ is the amount submitted by a b-agent willing to buy a unit of good. The highest bid in the market is called the *outstanding bid*, denoted $\mathfrak{b}_o$. A CDA can thus be described as a place where s-agents submit asks to decrease $\mathfrak{a}_o$, while b-agents submit bids to increase $\mathfrak{b}_o$, until $\mathfrak{b}_o$ is not less than $\mathfrak{a}_o$ [Gjerstad and Dickhaut, 1998]. At this moment, the s-agent that submits $\mathfrak{a}_o$ and the b-agent that submits $\mathfrak{b}_o$ can make a transaction, and the price of the transaction is called the transaction price. Formally, we have:

Definition 1 *The* **descriptor** *of a CDA is*

$$P_{CDA} = < g, \mathcal{B}, \mathcal{S}, V_b, C_s, \Delta_{price}, t_{round} >,$$

where:

1. *g is the good to be auctioned.*

2. $\mathcal{B} = \{b_1, \cdots, b_n\}$ *is the finite set of identifiers of b-agents, where n is the number of b-agents.*
3. $\mathcal{S} = \{s_1, \cdots, s_m\}$ *is the finite set of identifiers of s-agents, where m is the number of s-agents.*
4. $V_b = (\overrightarrow{V}_1, \cdots, \overrightarrow{V}_n)$*, where* $\overrightarrow{V}_i$ $(v_{i1}, v_{i2}, \cdots, v_{in_i})$ *is a vector of unit valuations of b-agent* b_i*. Here* n_i *is the number of units of g that* b_i *requires, and* v_{ij} *is the valuation value for the jth unit acquired.*
5. $C_s = (\overrightarrow{C}_1, \cdots, \overrightarrow{C}_m)$*, where* $\overrightarrow{C}_i$ $(c_{i1}, \cdots, c_{im_i})$ *is a vector of unit costs of s-agent* s_i*. Here* m_i *is the number of units that* s_i *wants to sell, and* c_{ij} *is the cost of the jth unit.*
6. Δ_{price} *is the minimum price step required in the auction. That is, a b-agent (s-agent) must increase (decrease) its bid (ask) at* $n \times \Delta_{price}$*, where n is a non-negative integer.*
7. t_{round} *is used for defining the condition for terminating the CDA; that is, if there are no new asks or bids during a time period* t_{round}*, the CDA terminates.*[1] □

Definition 2 *A* **round** *in a CDA is the time period between two successive deals or the period from the beginning of the CDA to the time when the first deal takes place. If a round is the rth* ($r \in \mathbb{N}^+$) *round of the CDA, then r is called the* **round number**. *A CDA usually consists of multiple rounds.* □

Definition 3 *For a CDA that has lasted r* ($r > 0$) *rounds, let* p_i ($1 \le i \le r$) *denote the price of the ith transaction. A* **history** H_l *in a CDA is the set of transaction prices during the last l rounds,*

$$H_l = \{p_{r-l+1}, \cdots, p_i, \cdots, p_r\},$$

where p_i ($r - l + 1 \le i \le r$) *is the transaction price of round i, and l* ($l \le r$) *is called the* **history length**.[2] □

The following is the formal definition of the valid behaviours of agents during a CDA.

Definition 4 *A* **CDA protocol** *with the descriptor* P_{CDA} *consists of the following steps:*

0. *r=0.*
1. *A new round of the CDA starts, r=r+1,* $\mathfrak{a}_o = \infty$ *and* $\mathfrak{b}_o$*=0.*

[1] Note that by this definition we exclude from this chapter CDAs that last infinite periods of time (such as stock markets). To model this, t_{round} can be set to infinity.

[2] Through experiments where both the history length (l) and the value (cost) of the goods that the agents trade varied, the performance of the agents with different history lengths was investigated. The results showed that the behaviour of FL-agents with a long history length ($l > 20$) was similar to or worse than that of an agent with a history length ranging from 3 to 20. This result shows that agents with short or intermediate history lengths can react more rapidly to changes in a CDA market. When the history length varied from 3 to 20, we found that 10 was a reasonable history length where almost all the agents achieve their highest profit. Thus this is the value selected for all the experiments in the rest of this chapter.

2. *Several situations might arise during a round:*
   (a) *When an s-agent submits an ask* $\mathfrak{a}$,
   (i) if $\mathfrak{a} \geq \mathfrak{a}_o$ *then* $\mathfrak{a}$ *is an invalid ask;*
   (ii) if $\mathfrak{b}_o < \mathfrak{a} < \mathfrak{a}_o$ *then* $\mathfrak{a}_o$ *is updated to* $\mathfrak{a}$*;*
   (iii) if $\mathfrak{a} \leq \mathfrak{b}_o$ *then this s-agent makes a deal at* $\mathfrak{b}_o$*; goto 1.*
   (b) *When a b-agent submits a bid of* $\mathfrak{b}$,
   (i) if $\mathfrak{b} \leq \mathfrak{b}_o$ *then* $\mathfrak{b}$ *is an invalid bid;*
   (ii) if $\mathfrak{b}_o < \mathfrak{b} < \mathfrak{a}_o$ *then* $\mathfrak{b}_o$ *is updated to* $\mathfrak{b}$*;*
   (iii) if $\mathfrak{b} \geq \mathfrak{a}_o$ *then this b-agent makes a deal at* $\mathfrak{a}_o$*; goto 1.*
3. *Step 2 repeats until no new bids (asks) are submitted during a time period* t_{round}.□

As can be seen, the outstanding ask and outstanding bid define the *bid-ask spread* $[\mathfrak{b}_o, \mathfrak{a}_o]$ [Gjerstad and Dickhaut, 1998] and only bids and asks that fall within this region are considered valid.

3.1.2 Fuzzy Reasoning Mechanisms

The fuzzy reasoning inference mechanism employed in this chapter is based on the Sugeno controllers [Sugeno, 1985, Zimmermann, 1996]. Consider the following block of fuzzy IF-THEN rules:

$$\begin{array}{ll}
\mathcal{R}_1: & \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z_1 = c_1 \\
\textbf{also} & \\
\mathcal{R}_2: & \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z_2 = c_2 \\
\textbf{also} & \\
\vdots & \\
\textbf{also} & \\
\mathcal{R}_n: & \text{if } x \text{ is } A_n \text{ and } y \text{ is } B_n \text{ then } z_n = c_n \\
\textbf{fact:} & x \text{ is } x_0 \text{ and } y \text{ is } y_0 \\
\hline
\textbf{consequence:} & z_0
\end{array}$$

where $A_1, \cdots, A_n$ and $B_1, \cdots, B_n$ are fuzzy sets, $z_1, \cdots, z_n$ are variables and $c_1, \cdots, c_n$ are constants. The firing level α_i of the rules $\mathcal{R}_i$ is computed by the *Min* operator. That is,

$$\alpha_i = \min\{A_i(x_0), B_i(y_0)\}, \tag{3.1}$$

where $A_i(x)$ and $B_i(y)$ are the membership functions of the corresponding fuzzy sets A_i and B_i, respectively. If the output of the individual rule is denoted as z_i, then according to the Sugeno controller definition, the crisp control action of the rule base is obtained by:

$$z_0 = \frac{\sum_{i=1}^{n} \alpha_i z_i}{\sum_{i=1}^{n} \alpha_i}. \tag{3.2}$$

The extension principle [Zadeh, 1965] is one of the main means of fuzzifying a formula with crisply defined numbers. In particular, we extend (3.2) to the situation where

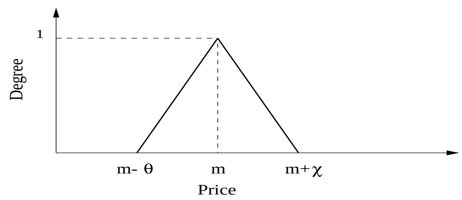


Figure 3.1: Triangular fuzzy number $\tilde{a} = (m, \theta, \chi)$.
m is the centre, θ is left spread and χ is right spread.

these real numbers z_i $(1 \leq i \leq n)$ are changed to triangular fuzzy numbers. We made this change because in developing our rules it is difficult to estimate the action using a single real value chosen from within a predefined range. Instead it is easier to estimate a parameter with fuzzy values, and this led us to use triangular fuzzy numbers (the user's preference is also presented by fuzzy numbers in Section 5.4.1) [Dubois and Prade, 1978]. Also, by the extension principle, arithmetic operations on trapezoidal fuzzy numbers have already been obtained [Bonissone and Decker, 1986, Luo et al., 1994] and fuzzy triangular numbers are special cases of fuzzy trapezoidal numbers [Bandemer and Gottwald, 1995]. Thus the arithmetic operations on fuzzy triangle numbers can be obtained from the arithmetic operations on fuzzy trapezoidal numbers. Given all this, in our inference mechanism, the output of each rule is a triangular fuzzy number defined with the following triple:

$$\tilde{a} = (m, \theta, \chi),$$

where m is called the centre, and θ and χ are called the left and right spreads, respectively [Yager and Filev, 1994] (Figure 3.1). For two triangular fuzzy numbers $\tilde{a}_1 = (m_1, \theta_1, \chi_1)$, and $\tilde{a}_2 = (m_2, \theta_2, \chi_2)$ $(\tilde{a}_1, \tilde{a}_2 > 0)$ and $k \in I\!R$, the following formulae hold [Bonissone and Decker, 1986, Luo et al., 1994]:

$$\begin{aligned}
&\tilde{a}_1 + \tilde{a}_2 = (m_1 + m_2, \theta_1 + \theta_2, \chi_1 + \chi_2), \\
&\tilde{a}_1 - \tilde{a}_2 = (m_1 - m_2, \theta_1 + \chi_2, \theta_2 + \chi_1), \\
&\tilde{a}_1 \times \tilde{a}_2 = (m_1 m_2, m_1\theta_2 + m_2\theta_1 - \theta_1\theta_2, m_1\chi_2 + m_2\chi_1 + \chi_1\chi_2), \\
&k \times \tilde{a}_1 = (km_1, k\theta_1, k\chi_1).
\end{aligned}$$

From the above formulae, (3.2) can be extended to the following in the situation where $\tilde{z}_i = (m_i, \theta_i, \chi_i)$ $(1 \leq i \leq n)$:

$$\begin{aligned}
\tilde{z}_0 &= \frac{\sum_{i=1}^{n}(\alpha_i \times \tilde{z}_i)}{\sum_{i=1}^{n}\alpha_i} \\
&= \frac{\sum_{i=1}^{n}(\alpha_i \times (m_i, \theta_i, \chi_i))}{\sum_{i=1}^{n}\alpha_i} \\
&= \left(\frac{\sum_{i=1}^{n}(\alpha_i \times m_i)}{\sum_{i=1}^{n}\alpha_i}, \frac{\sum_{i=1}^{n}(\alpha_i \times \theta_i)}{\sum_{i=1}^{n}\alpha_i}, \frac{\sum_{i=1}^{n}(\alpha_i \times \chi_i)}{\sum_{i=1}^{n}\alpha_i}\right). \qquad (3.3)
\end{aligned}$$

Thus, the reasoning mechanism becomes:

$$\begin{array}{ll}
\mathcal{R}_1: & \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } \tilde{z}_1 \text{ is } \tilde{c}_1 \\
\textbf{also} & \\
\mathcal{R}_2: & \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } \tilde{z}_2 \text{ is } \tilde{c}_2 \\
\textbf{also} & \\
\vdots & \\
\textbf{also} & \\
\mathcal{R}_n: & \text{if } x \text{ is } A_n \text{ and } y \text{ is } B_n \text{ then } \tilde{z}_n \text{ is } \tilde{c}_n \\
\textbf{fact:} & x \text{ is } x_0 \text{ and } y \text{ is } y_0 \\
\hline
\textbf{consequence:} & \tilde{z}_0
\end{array}$$

Having defined the protocol and the reasoning mechanism, we can now turn to the FL-strategy itself.

3.2 The FL-Strategy

Building on the foundations of the previous section, this section describes our FL-strategy and demonstrates how it works in an exemplar scenario.

3.2.1 Basic Notation and Concepts

In order to detail the FL-strategy, we first need to introduce a number of underpinning notations and concepts.

Definition 5 *A* **situation** s^* *during the course of a CDA is a* $6-tuple$,

$$s^* =< r, \mathcal{B}, \mathcal{S}, \mathfrak{a}_o, \mathfrak{b}_o, H_l >,$$

where r *is the current round number;* $\mathcal{B}$ *and* $\mathcal{S}$ *are the sets of b-agents and s-agents;* $\mathfrak{a}_o$ *and* $\mathfrak{b}_o$ *are the outstanding ask and the outstanding bid, respectively; and* H_l *is the history of the last* l *rounds.*[3] □

Definition 6 *Given a situation* s^**, the* **valid bids set (**D_b**)** *is the set of the valid bids that a b-agent could submit:*

$$D_b(v_{ij}) = \{\mathfrak{b} \mid \mathfrak{b}_o < \mathfrak{b} \leq \min(\mathfrak{a}_o, v_{ij})\}, \tag{3.4}$$

where $\mathfrak{b}$ *is the price at which a b-agent submits a bid; and* v_{ij} *is the valuation of the* j*th unit of the good by buyer* i*.* □

Definition 7 *Given a situation* s^**, the* **valid asks set (**D_s**)** *is the set of valid asks that an s-agent could submit:*

$$D_s(c_{ij}) = \{\mathfrak{a} \mid \max(\mathfrak{b}_o, c_{ij}) \leq \mathfrak{a} < \mathfrak{a}_o\}, \tag{3.5}$$

[3] Recall that l is the remembered history length of an agent, and thus l is not necessarily equal to $r-1$.

where $\mathfrak{a}$ is the price at which an s-agent submits an ask; and c_{ij} is the cost of the jth unit of the good for seller i. □

The prices of previous transactions are stored as history and may be referred to by the agents in the subsequent rounds. Generally speaking, CDA markets produce very efficient allocations and prices [Easley and Ledyard, 1993], and the transaction prices often converge to a competitive equilibrium price[4] while the CDA is in progress. Thus, the transaction prices in a CDA provide an important point for reference. To reflect this fact, we define the **reference price** P_R (this can be regarded as the prediction of the transaction price) in the situation s^* as the median of the ordered price history.[5] A reference price, as its name suggests, provides a reference point that an agent can use to guide its subsequent bidding behaviour. Formally, we have:

Definition 8 *Let r be the current round number ($r > 0$). Suppose the price history is a series of prices*

$$H_l = \{p_{r-l}, \cdots, p_i, \cdots, p_{r-1}\},$$

where p_i ($r-l \leq i \leq r-1$) is the price in round i. Let their ordered series be denoted as

$$p_{(1)} \leq \cdots \leq p_{(i)} \leq \cdots \leq p_{(l)}. \tag{3.6}$$

Then the reference price, P_R, is given by

$$P_R = p_{(\lfloor \frac{l+1}{2} \rfloor)}. \tag{3.7}$$

□

To summarise, when an agent submits its next ask (bid), it will consider the outstanding ask, the outstanding bid, the cost (valuation) of the current unit of good, and the reference price. The way in which these values are used is described in the next subsection.

3.2.2 Fuzzy Reasoning in the FL-strategy

The FL-strategy is based on a number of heuristic rules and the fuzzy reasoning mechanism outlined in Section 3.1.2. The relation of P_R, $\mathfrak{a}_o$, and $\mathfrak{b}_o$ during a round in a CDA falls into one of the cases below: (i) $P_R \leq \mathfrak{b}_o < \mathfrak{a}_o$, (ii) $\mathfrak{b}_o < \mathfrak{a}_o \leq P_R$, and (iii) $\mathfrak{b}_o \leq P_R \leq \mathfrak{a}_o$. In the first two cases, we use some heuristic rules (given below); the bidding issue in the

[4] The equilibrium price is determined by the intersection of the supply and demand curves of the market, and it is the point where the quantity supplied is equal to the quantity demanded [Perloff, 1998].

[5] Originally, both the mean and the median of the ordered price history were used, however experimental results showed that the median is more effective in providing a reference price. This is because the mean price can be overly influenced by a too high (low) price offered by an irrational agent. In contrast, the median of the ordered price history is less susceptible to such bias.

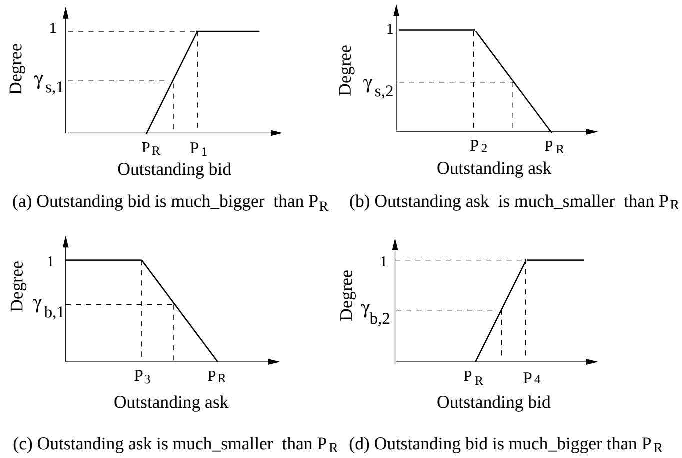


Figure 3.2: Fuzzy sets in heuristic rules.

third case, which is more complicated, is handled through the fuzzy reasoning mechanism on a rule base (described at the end of this subsection). Figure 3.2 describes all the fuzzy sets used in the heuristic rules. The heuristic rules applied in the first two cases for s-agents are:

- When $P_R \leq \mathfrak{b}_o < \mathfrak{a}_o$, the heuristic rule is:
  (SR_1) IF $\mathfrak{b}_o$ is *much_bigger* than P_R
  THEN accept $\mathfrak{b}_o$
  ELSE *ask* is $(\mathfrak{a}_o - \beta_{s,1}, \theta, \chi)$.
- When $\mathfrak{b}_o < \mathfrak{a}_o \leq P_R$, the heuristic rule is:
  (SR_2) IF $\mathfrak{a}_o$ is *much_smaller* than P_R
  THEN no new ask
  ELSE ask is $(\mathfrak{a}_o - \beta_{s,2}, \theta, \chi)$.

Intuitively, SR_1 states that when the outstanding bid $\mathfrak{b}_o$ is *much_bigger* than the reference price P_R, it is already very profitable for an s-agent to accept the current outstanding bid. The relation '$\mathfrak{b}_o$ is *much_bigger* than P_R' can be expressed as fuzzy set A_1. Let the threshold be $\gamma_{s,1}$, that is, if $A_1(\mathfrak{b}_o) \geq \gamma_{s,1}$, the s-agent will accept $\mathfrak{b}_o$. At this point, a transaction takes place between the s-agent and the b-agent which submits the outstanding bid. Otherwise, the s-agent will decrease the outstanding ask $\mathfrak{a}_o$ to a fuzzy number $(\mathfrak{a}_o - \beta_{s,1}, \theta, \chi)$ (see Section 3.1.2), where $\mathfrak{a}_o - \beta_{s,1}$ is the centre of the new ask, and θ and χ are the left and right spread. $\beta_{s,1}$ shows how much the agent would like to decrease its ask and this is decided by the agents' attitude to risk (to be discussed in Section 3.3.2). SR_2 is applied when $\mathfrak{a}_o$ is much smaller than P_R. At this moment, an s-agent is in an unfavourable position and it should be reluctant to decrease $\mathfrak{a}_o$. Thus the s-agent only decreases $\mathfrak{a}_o$ by a small step. The relationship '$\mathfrak{a}_o$ is *much_smaller* than P_R' is expressed as a fuzzy set A_2. Let $\gamma_{s,2}$ be the threshold, that is, if $A_2(\mathfrak{a}_o) \geq \gamma_{s,2}$, the agent believes the current ask is much smaller than P_R. In this case, the s-agent will not submit a new ask.

Similar heuristic rules also apply to b-agents:

- When $\mathfrak{b}_o < \mathfrak{a}_o \leq P_R$, the heuristic rule is:
  (BR_1) IF $\mathfrak{a}_o$ is *much_smaller* than P_R
  THEN accept $\mathfrak{a}_o$
  ELSE bid is $(\mathfrak{b}_o + \beta_{b,1}, \theta, \chi)$.
- When $P_R \leq \mathfrak{b}_o < \mathfrak{a}_o$, the heuristic rule is:
  (BR_2) IF $\mathfrak{b}_o$ is *much_bigger* than P_R
  THEN no new bid
  ELSE bid is $(\mathfrak{b}_o + \beta_{b,2}, \theta, \chi)$.

The relationship '$\mathfrak{a}_o$ is *much_smaller* than P_R' can be expressed as a fuzzy set A_3. Let $\gamma_{b,1}$ be the threshold, that is, if $A_3(\mathfrak{a}_o) \geq \gamma_{b,1}$, $\mathfrak{a}_o$ is regarded as being much smaller than P_R, and a b-agent will accept $\mathfrak{a}_o$; otherwise, a b-agent will increase $\mathfrak{b}_o$ to a fuzzy number $(\mathfrak{b}_o + \beta_{b,1}, \theta, \chi)$. The fuzzy set A_4 defines the relationship '$\mathfrak{b}_o$ is *much_bigger* than P_R'. Let $\gamma_{b,2}$ be the threshold for this rule, that is, if $A_4(\mathfrak{b}_o) \geq \gamma_{b,2}$, a b-agent will not submit a new bid because $\mathfrak{b}_o$ is already high enough and no profit can be made according to its preference; otherwise, it will increase $\mathfrak{b}_o$ to a fuzzy number $(\mathfrak{b}_o + \beta_{b,2}, \theta, \chi)$. In the above, P_1, P_2, P_3, and P_4 are the parameters of the fuzzy sets (see Figure 3.2) and they are decided by human intuition and experience according to the range of the cost and valuation of the goods. The fuzzy number produced by these heuristic rules is dealt with in the same way as the fuzzy number produced by the reasoning mechanism (which we will discuss below).

Now for the third case ($\mathfrak{b}_o \leq P_R \leq \mathfrak{a}_o$), the fuzzy reasoning on a rule base is required. First, the rule bases for the s-agents and b-agents are presented in Tables 3.1 and 3.2, respectively. Again the fuzzy numbers are all triangular fuzzy numbers as described in Section 3.1.2; the distance between $\mathfrak{a}_o$ (or $\mathfrak{b}_o$) and P_R is expressed using the fuzzy linguistic terms: far_from, *medium_to*, and *close_to*, which are defined in Figure 3.3, and *or* corresponds to operator *Max*. $\lambda_{s,1}, \cdots, \lambda_{s,4}$ and $\lambda_{b,1}, \cdots, \lambda_{b,4}$ are parameters decided by the risk attitude of the agent (see Section 3.3.2). Based on these rule bases, we can perform inference through the fuzzy reasoning mechanism presented in Section 3.1.2. The overall output of our fuzzy reasoning is a fuzzy number, *i.e.*, a set of asks (bids) with membership degrees. For example, $\tilde{z}$ may equal $(2.0, 0.02, 0.04)$, where 2.0 is the centre, 0.02 is the left spread, and 0.04 is the right spread and its membership degree might be given by:

$$\tilde{z}(x) = \begin{cases} \text{50x-99} & \text{if } 1.98 \leq x \leq 2.0, \\ \text{-25x+51} & \text{if } 2.0 < x \leq 2.04. \end{cases}$$

Now the decision sets DS_s (acceptable asks for s-agents) and DS_b (acceptable bids for b-agents) can be determined. Suppose $z_s = (m_s, \theta_s, \chi_s)$ is the output fuzzy number of the fuzzy reasoning or the heuristic rules for an s-agent, $z_b = (m_b, \theta_b, \chi_b)$ is the output fuzzy number of the fuzzy reasoning or the heuristic rules for a b-agent, and the parameter π_s, for the s-agent, and π_b, for the b-agent, are the thresholds to decide to which degree the

Table 3.1: Fuzzy rule base for s-agents.

IF	($\mathfrak{b}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{a}_o$ is *far_from* P_R)
THEN	ask is $(\mathfrak{a}_o - \lambda_{s,1}, \theta, \chi)$.
IF	($\mathfrak{b}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{a}_o$ is *medium_to* P_R)
THEN	ask is $(\mathfrak{a}_o - \lambda_{s,2}, \theta, \chi)$.
IF	($\mathfrak{b}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{a}_o$ is *close_to* P_R)
THEN	ask is $(\mathfrak{a}_o - \lambda_{s,3}, \theta, \chi)$.
IF	$\mathfrak{b}_o$ is *close_to* P_R
THEN	ask is $(P_R + \lambda_{s,4}, \theta, \chi)$.

Table 3.2: Fuzzy rule base for b-agents.

IF	($\mathfrak{a}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{b}_o$ is *far_from* P_R)
THEN	bid is $(\mathfrak{b}_o + \lambda_{b,1}, \theta, \chi)$.
IF	($\mathfrak{a}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{b}_o$ is *medium_to* P_R)
THEN	bid is $(\mathfrak{b}_o + \lambda_{b,2}, \theta, \chi)$.
IF	($\mathfrak{a}_o$ is *far_from* or *medium_to* P_R) and
	($\mathfrak{b}_o$ is *close_to* P_R)
THEN	bid is $(\mathfrak{b}_o + \lambda_{b,3}, \theta, \chi)$.
IF	$\mathfrak{a}_o$ is *close_to* P_R
THEN	bid is $(P_R - \lambda_{b,4}, \theta, \chi)$.

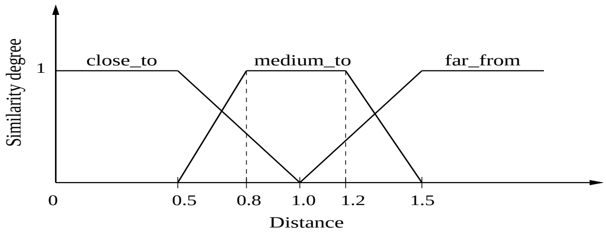


Figure 3.3: Fuzzy sets used in fuzzy reasoning.

ask (bid) could be accepted. Again, π_s and π_b can be decided by the risk attitudes of the agents. The asks that the s-agent could submit are in the decision set:

$$DS_s = \{\mathfrak{a} \mid \mathfrak{a} \in D_s \cap \{\mathfrak{a} \mid z_s(\mathfrak{a}) \geq \pi_s\}\}, \tag{3.8}$$

where D_s is the valid asks set (see Definition 7). Similarly, the bids that the b-agent could submit are in the decision set:

$$DS_b = \{\mathfrak{b} \mid \mathfrak{b} \in D_b \cap \{\mathfrak{b} \mid z_b(\mathfrak{b}) \geq \pi_b\}\}, \tag{3.9}$$

where D_b is the valid bids set (see Definition 6).

Finally, the agent can decide whether to accept an ask (bid) or submit an ask (bid), or submit nothing. For an FL-agent, if the decision set, DS_s (DS_b), is empty, it shows that there is no acceptable asks (bids) at which this agent can make any profit, thus it will not submit an ask or a bid. Otherwise, the ask (bid) to be submitted is decided by the following formulae:

- for FL s-agents:

$$ask = \begin{cases} \mathfrak{b}_o & \text{if } \mathfrak{b}_o \in DS_s, \\ \arg \underset{\mathfrak{a} \in DS_s}{max} \{z_s(\mathfrak{a})\} & \text{otherwise;} \end{cases} \tag{3.10}$$

- for FL b-agents:

$$bid = \begin{cases} \mathfrak{a}_o & \text{if } \mathfrak{a}_o \in DS_b, \\ \arg \underset{\mathfrak{b} \in DS_b}{max} \{z_b(\mathfrak{b})\} & \text{otherwise.} \end{cases} \tag{3.11}$$

For an FL s-agent (b-agent), if the outstanding bid (ask) falls into DS_s (DS_b), it is a sign that $\mathfrak{b}_o$ ($\mathfrak{a}_o$) is acceptable (Here the decision set is a fuzzy number and it realises the flexible bidding we discussed in Section 1.2). The FL s-agent (b-agent) will submit $\mathfrak{b}_o$ ($\mathfrak{a}_o$) in order to make a transaction at $\mathfrak{b}_o$ ($\mathfrak{a}_o$). Otherwise, it will select the ask (bid) which corresponds to the maximum similarity degree among those asks (bids) constrained by DS_s (DS_b).

This completes the description of our FL-strategy for both buyer and seller agents in a CDA. We now illustrate its use in an exemplar scenario.

3.2.3 The FL-strategy in Operation

Assume there are three valuation vectors for b-agents b_1, b_2, and b_3:

$$\overrightarrow{V}_1 = \{3.3, 2.7, 2.4\}, \overrightarrow{V}_2 = \{2.8, 2.5, 2.2\}, \overrightarrow{V}_3 = \{2.7, 2.4, 2.1\},$$

and three cost vectors for s-agents s_1, s_2, and s_3:

$$\overrightarrow{C}_1 = \{1.6, 2.2, 2.4\}, \overrightarrow{C}_2 = \{1.75, 2.0, 2.3\}, \overrightarrow{C}_3 = \{1.6, 1.9, 2.1\}.$$

Furthermore, suppose the CDA market is as follows (see Definition 1):

$$P_{CDA} =< g, \{b_1, b_2, b_3\}, \{s_1, s_2, s_3\}, (\overrightarrow{V}_1, \overrightarrow{V}_2, \overrightarrow{V}_3), (\overrightarrow{C}_1, \overrightarrow{C}_2, \overrightarrow{C}_3), 0.01, 30 > .$$

In this market, there are 3 b-agents, each with valuations for three units, and 3 s-agents each with costs for three units. Consider the following situation (see Definition 5):

$$s^* =< 6, \{b_1, b_2, b_3\}, \{s_1, s_2, s_3\}, \mathfrak{a}_o, \mathfrak{b}_o, H_l > .$$

The fuzzy sets employed in the FL-strategy are shown in Figs. 3.2 and 3.3. Based on the ranges of asks and bids, a difference below 0.01 in the ask (bid) value is here assumed to be indifferent to the users. Thus, we choose 0.01 as the price step, *i.e.*, $\Delta_{price} = 0.01$. Also, for simplicity, the thresholds for all the fuzzy sets used in the rules are set to 0.5, *i.e.*, $\gamma_{s,1} = \gamma_{s,2} = \gamma_{b,1} = \gamma_{b,2} = 0.5$. For all the fuzzy numbers involved, suppose their left spread $\theta = 0.02$ and their right spread $\chi = 0.02$, which ensures a reasonable degree of flexibility in this context.

Example 1 *This example shows how to use the heuristic rules to submit a bid. For ease of explanation, let the history length* $l = 3$ *(however it does not affect the rationale of the strategy). Let* $H_l = \{2.3, 2.2, 2.1\}$, $\mathfrak{b}_o = 2.4$, $\mathfrak{a}_o = 2.5$, $P_1 = 2.5$, *and an s-agent be about to submit its next ask. First from* s^*, $r = 6$. *Then, by Definition 8, we have*

$$P_R = p_{(\lfloor \frac{l+1}{2} \rfloor)} = p_{(\lfloor \frac{3+1}{2} \rfloor)} = p_{(2)} = 2.2.$$

From Figure 3.2(a), we can find that $A_1(\mathfrak{b}_o) = A_1(2.4) = 0.667 > \gamma_{s,1} = 0.5$. *That is,* $\mathfrak{b}_o = 2.4$ *is considered to be much bigger than* $P_R = 2.2$. *Therefore according to rule* SR_1, *the s-agent will accept* $\mathfrak{b}_o$, *i.e., ask* $= \mathfrak{b}_o = 2.4$.

Example 2 *This example shows how to use the fuzzy reasoning mechanism to submit a bid. For ease of explanation, we set the history length to be* 5. *Let* $H_l = \{2.0, 2.4, 2.3, 2.2, 2.1\}$

($l=5$), $\mathfrak{a}_o=2.85$, $\mathfrak{b}_o=1.2$, and the FL b-agent b_1 (with valuation $\overrightarrow{V}_1$) be about to submit a new bid for its second unit of good, that is, the valuation of the second unit of good is $v_{12}=2.7$. By Definition 8,

$$P_R=p_{(\lfloor\frac{l+1}{2}\rfloor)}=p_{(\lfloor\frac{5+1}{2}\rfloor)}=p_{(3)}=2.2.$$

Since $\mathfrak{b}_o\le P_R\le\mathfrak{a}_o$, the fuzzy reasoning on the rule base is employed. Let $\lambda_{b,1}=0.05$, $\lambda_{b,2}=0.04$, $\lambda_{b,3}=0.01$, and $\lambda_{b,4}=0.02$. From Figure 3.3, we can find $close\_to(\mathfrak{a}_o)=0.7$, $medium\_to(\mathfrak{a}_o)=0.5$, $far\_from(\mathfrak{a}_o)=0$, $close\_to(\mathfrak{b}_o)=0$, $medium\_to(\mathfrak{b}_o)=1$, and $far\_from(\mathfrak{b}_o)=0$. By formula (1), the four rules' firing levels in Table 3.2 are:

$$\begin{aligned}
\alpha_1 &= \min\{\max\{far_from(\mathfrak{a}_o), medium_to(\mathfrak{a}_o)\}, far_from(\mathfrak{b}_o)\}\\
&= \min\{\max\{0,0.5\},0\}=0,\\
\alpha_2 &= \min\{\max\{far_from(\mathfrak{a}_o), medium_to(\mathfrak{a}_o)\}, medium_to(\mathfrak{b}_o)\}\\
&= \min\{\max\{0,0.5\},1\}=0.5,\\
\alpha_3 &= \min\{\max\{far_from(\mathfrak{a}_o), medium_to(\mathfrak{a}_o)\}, close_to(\mathfrak{b}_o)\}\\
&= \min\{\max\{0,0.5\},0\}=0,\\
\alpha_4 &= close_to(\mathfrak{a}_o)=0.7.
\end{aligned}$$

Thus, according to Table 3.2, the four rules' outputs are:

$$\begin{aligned}
\tilde{z}_1 &= (\mathfrak{b}_o+\lambda_{b,1},\theta,\chi)=(1.20+0.05,0.02,0.02)=(1.25,0.02,0.02),\\
\tilde{z}_2 &= (\mathfrak{b}_o+\lambda_{b,2},\theta,\chi)=(1.20+0.04,0.02,0.02)=(1.24,0.02,0.02),\\
\tilde{z}_3 &= (\mathfrak{b}_o+\lambda_{b,3},\theta,\chi)=(1.20+0.01,0.02,0.02)=(1.21,0.02,0.02),\\
\tilde{z}_4 &= (P_R-\lambda_{b,4},\theta,\chi)=(2.2-0.02,0.02,0.02)=(2.18,0.02,0.02).
\end{aligned}$$

Finally, by formula (3.3), the overall output fuzzy number is calculated as follows:

$$\begin{aligned}
\tilde{z}_0 &= \frac{\alpha_1\times\tilde{z}_1+\alpha_2\times\tilde{z}_2+\alpha_3\times\tilde{z}_3+\alpha_4\times\tilde{z}_4}{\alpha_1+\alpha_2+\alpha_3+\alpha_4}\\
&= \frac{0\times\tilde{z}_1+0.5\times\tilde{z}_2+0\times\tilde{z}_3+0.7\times\tilde{z}_4}{0+0.5+0+0.7}\\
&= \frac{0.5\times\tilde{z}_2+0.7\times\tilde{z}_4}{1.2}\\
&= \frac{0.5\times(1.24,0.02,0.02)}{1.2}+\frac{0.7\times(2.18,0.02,0.02)}{1.2}\\
&= \frac{(0.5\times1.24,0.5\times0.02,0.5\times0.02)}{1.2}+\frac{(0.7\times2.18,0.7\times0.02,0.7\times0.02)}{1.2}\\
&= \frac{(0.62,0.01,0.01)+(1.526,0.014,0.014)}{1.2}\\
&= (1.79,0.02,0.02).
\end{aligned}$$

Then, by Definition 6, the valid bids set is:

$$D_b = \{\mathfrak{b} \mid \mathfrak{b}_o < \mathfrak{b} \leq \min(\mathfrak{a}_o, v_{ij})\} = \{\mathfrak{b} \mid 1.2 < \mathfrak{b} \leq \min(2.85, 2.7)\} = \{\mathfrak{b} \mid 1.2 < \mathfrak{b} \leq 2.7\}.$$

And then, by formula (3.9), the bids that this b-agent can accept are in the decision set:

$$DS_b = \{\mathfrak{b} \mid \mathfrak{b} \in D_b \cap \{\mathfrak{b} \mid z_b(\mathfrak{b}) \geq \pi_b\}\} = \{1.78, 1.79, 1.80\},$$

where $\pi_b = 0.5$*. Finally, by formula (3.11), we have*

$$\arg \max_{\mathfrak{b} \in DS_b} \{z_b(1.78), z_b(1.79), z_b(1.80)\} = 1.79.$$

Thus, the bid that the b-agent will submit is $bid = 1.79$*.*

3.3 Evaluation of FL-agents

This section investigates, in an empirical fashion, the influence of the key parameters of the FL-strategy, the selection of these parameters, and the comparison of the FL-strategy with a number of other prominent strategies that have been proposed in the literature.

3.3.1 The Experimental Setting

This subsection describes the settings for the experiments conducted in the rest of this chapter. First, the time period that an agent can allow to elapse before sending a message about asks or bids is specified as an exponentially distributed random variable. This is chosen because: (i) each agent's timing decision is independent of domain characteristics such as costs or valuations, and (ii) exponential distribution is often a good approximation of the actual distribution [Ross, 1989]. Second, to measure how well an agent performs in a CDA, we evaluate its profit (the monetary gain for the agent). For an s-agent, the gain on its ith unit sold is the difference between the price, p_i, received from a b-agent for that unit, and the cost, c_i, at which the unit is produced, *i.e.*, $p_i - c_i$. If the s-agent sells m units at prices $p_1, \cdots, p_m$, then its profit is $\sum_{1 \leq i \leq m}(p_i - c_i)$. Similarly, for a b-agent, if this agent trades n units of goods, its profit is $\sum_{1 \leq i \leq n}(v_i - p_i)$, where v_i is the valuation value for the ith unit and p_i is the price of buying the ith unit of good. For the rest of this chapter, an agent's profit is calculated as the sum of the profit in $1,000$ simulations.[6]

Based on the above settings, each experiment is composed of many sessions and then each session consists of $1,000$ runs.[7] In each run of the session, an s-agent is endowed

[6] This number is chosen because it is sufficient to produce statistically significant results. By a *t-test*, the *p value* of 0.007 is reported from the sample of 900 runs and that of $1,000$ runs. Thus the profit variance for the two samples are virtually the same and the results are therefore statistically significant at the 99.3% level of confidence.

[7] From the beginning of the CDA to its termination is called a run. $1,000$ runs with the same s-agents and b-agents make up a session.

with a number of units of goods whose costs are independently drawn randomly from a uniform distribution with support [1.00, 3.00]. A b-agent is endowed with a number of units of goods whose valuations are independently drawn from a uniform distribution with support [2.00, 4.00]. These values were chosen because the cost values are generally smaller than the valuation values [Cason and Friedman, 1991]. Thus, this is consistent with reality. The supply of the market is calculated by the total number of units of goods that all the s-agents want to sell; and the demand is calculated by the total number of units of goods that all the b-agents desire to buy. For example, if there are 5 s-agents and 5 b-agents in the market, each s-agent is endowed with 5 units of goods, and each b-agent is endowed with 6 units of goods to buy. The supply is 25 and the demand is 30.

3.3.2 Agents with Attitudes Towards Risk

This subsection first defines the different attitudes towards risk that an agent can adopt and then analyses the influences of these attitudes through experiments based on the settings described above. As discussed in Section 1.2, an agent's attitude towards risk is an important factor to consider when designing a trading agent. Due to the complexity and uncertainty of the CDA bidding problem, it is not possible to analytically determine the optimal configuration of parameter values for a given context [Friedman, 1993]. The best that can be achieved is to know the likely range of parameters such that the agent will perform effectively. To this end, the concept of attitude towards risk is introduced. Individual attitudes to risk can be characterised according to how an agent approaches a fair gamble [Schotter, 1994]; they can be: risk-neutral, risk-averse or risk-seeking. Take the utility functions of an s-agent as an example. In Figure 3.4, the price of the outstanding bid ($\mathfrak{b}_o$) appears on the horizontal axis and the utility generated by accepting the current $\mathfrak{b}_o$ is shown on the vertical axis. For the same value of p, agents with different risk attitudes have different utilities; that is, $U^{(A)}(p) \geq U^{(N)}(p) \geq U^{(S)}(p)$. The agent with the utility function $U^{(A)}$ represents the risk-averse agent which takes minimal risks with its actions. Suppose the cost of the current unit of the good is c, as a result, it is unwilling to sacrifice a sure profit of $(\mathfrak{b}_o - c)$ although there may be a greater chance of gaining more profit. In short, risk-averse agents reject fair gambles. In contrast, there are agents that actually prefer fair gambles to sure results. These agents are called risk-seeking and are represented by the utility function $U^{(S)}$. The agents with the attitude between these two extremes are called risk-neutral agents and their utility function is always represented as a straight line. This kind of agent will be indifferent if the sure result and the gamble have the same expected utilities.

Thus, in the FL-strategy, given the same fuzzy sets, different parameters will correspond to different agent attitudes.

Definition 9 *Given the same situation, suppose the utilities of an ask $\mathfrak{a}$ that two s-agents s and s' submit are $U_s(\mathfrak{a})$ and $U_s'(\mathfrak{a})$, respectively. For all $\mathfrak{a} \in D_s$, agent s is said to be*

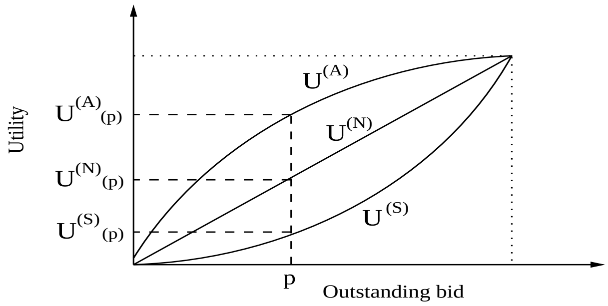


Figure 3.4: Utility functions of agents with different attitudes.

more averse towards risk than agent s' *if* $U_s(\mathfrak{a}) \geq U_s'(\mathfrak{a})$. *This we denote as* $s \succeq_a^s s'$. $\square$

The following propositions are a straightforward result of Definition 9.

Proposition 1 *Given the same situation, suppose the utilities of submitting an ask* $\mathfrak{a}$ *for three agents* $s^{(A)}$, $s^{(N)}$ *and* $s^{(S)}$ *are* $U_s^{(A)}(\mathfrak{a})$, $U_s^{(N)}(\mathfrak{a})$ *and* $U_s^{(S)}(\mathfrak{a})$, *respectively. For all* $\mathfrak{a} \in D_s$, $s^{(A)} \succeq_a^s s^{(N)} \succeq_a^s s^{(S)}$ *if and only if* $U_s^{(A)}(\mathfrak{a}) \geq U_s^{(N)}(\mathfrak{a}) \geq U_s^{(S)}(\mathfrak{a})$. $\square$

Proposition 2 *For three s-agents* $s^{(A)}$, $s^{(N)}$ *and* $s^{(S)}$, *represented by* $(\beta_{s,1}^{(A)}, \beta_{s,2}^{(A)}, \gamma_{s,1}^{(A)}, \gamma_{s,2}^{(A)}, \lambda_{s,1}^{(A)}, \cdots, \lambda_{s,4}^{(A)})$, $(\beta_{s,1}^{(N)}, \beta_{s,2}^{(N)}, \gamma_{s,1}^{(N)}, \gamma_{s,2}^{(N)}, \lambda_{s,1}^{(N)}, \cdots, \lambda_{s,4}^{(N)})$ *and* $(\beta_{s,1}^{(S)}, \beta_{s,2}^{(S)}, \gamma_{s,1}^{(S)}, \gamma_{s,2}^{(S)}, \lambda_{s,1}^{(S)}, \cdots, \lambda_{s,4}^{(S)})$, *respectively. If all the following conditions hold:*

$$
\begin{aligned}
(i)\, & \beta_{s,i}^{(A)} > \beta_{s,i}^{(N)} > \beta_{s,i}^{(S)}\ (for\ \ i = 1 \text{ and } 2)\\
(ii)\, & \gamma_{s,1}^{(A)} < \gamma_{s,1}^{(N)} < \gamma_{s,1}^{(S)},\\
(iii)\, & \gamma_{s,2}^{(A)} > \gamma_{s,2}^{(N)} > \gamma_{s,2}^{(S)},\\
(iv)\, & \lambda_{s,j}^{(A)} > \lambda_{s,j}^{(N)} > \lambda_{s,j}^{(S)}\ (for\ \ j = 1,2 \text{ and } 3)\ \text{ and }\ \lambda_{s,4}^{(A)} < \lambda_{s,4}^{(N)} < \lambda_{s,4}^{(S)},
\end{aligned}
$$

then, $s^{(A)} \succeq_a^s s^{(N)} \succeq_a^s s^{(S)}$. $\square$

Proof. For an s-agent, for all $\mathfrak{a} \in D_s$, $U_s(\mathfrak{a})$ is a non-decreasing function. That is, the bigger the ask, the more utility the agent obtains. Let the ask submitted by each s-agent be $\mathfrak{a}^{(A)}$, $\mathfrak{a}^{(N)}$, and $\mathfrak{a}^{(S)}$. From (i) to (iv), we can always get $\mathfrak{a}^{(A)} < \mathfrak{a}^{(N)} < \mathfrak{a}^{(S)}$. That is, $s^{(A)}$ always submits a lower ask compared with $s^{(N)}$ and $s^{(S)}$, and that $s^{(N)}$ is always lower than $s^{(S)}$. Thus $U_s^{(A)}(\mathfrak{a}) \geq U_s^{(N)}(\mathfrak{a}) \geq U_s^{(S)}(\mathfrak{a})$. Based on Proposition 1, we have: $s^{(A)} \succeq_a^s s^{(N)} \succeq_a^s s^{(S)}$.

Similarly, $\succeq_a^b$ can be defined as follows.

Definition 10 *Given the same situation, suppose the utilities of a bid* $\mathfrak{b}$ *for two b-agents* b *and* b' *submit are* $U_b(\mathfrak{b})$ *and* $U_b'(\mathfrak{b})$, *respectively. For all* $\mathfrak{b} \in D_b$, *agent* b *is said to be more averse towards risk than agent* b' *if* $U_b(\mathfrak{b}) \geq U_b'(\mathfrak{b})$. *This we denote as* $b \succeq_a^b b'$. $\square$

The following propositions are a straightforward result of Definition 10.

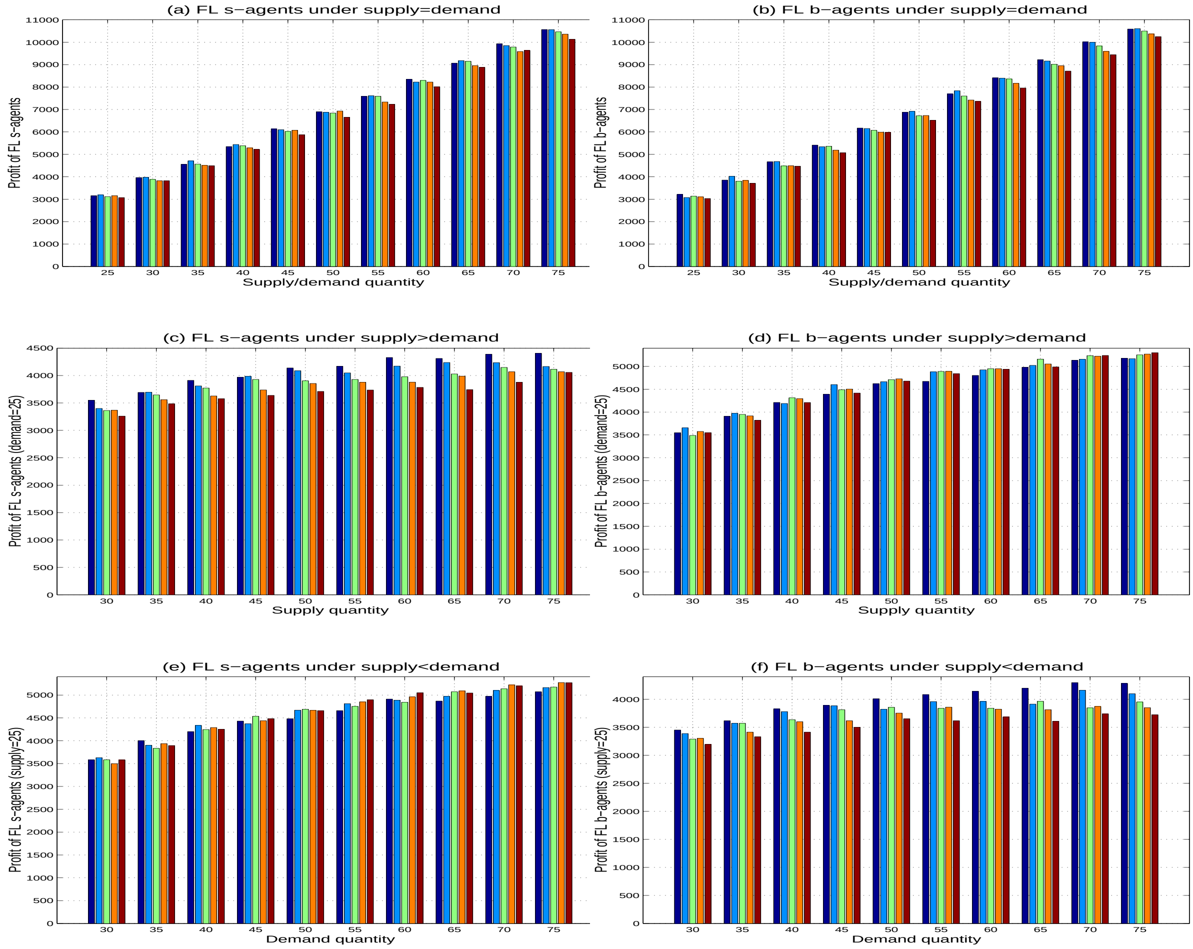


Figure 3.5: Performance of FL-agents with different risk attitudes.
In each group of bars, the five bars represent, from left to right, the agents with different attitudes: averse, weakly averse, neutral, weakly risky and risky. The horizontal axis shows the supply (demand) quantity of the session. The vertical axis represents the profit of the various FL-agents in the session.

Proposition 3 *Given the same situation, suppose the utilities of submitting a bid $\mathfrak{b}$ for the b-agents $b^{(A)}$, $b^{(N)}$, and $b^{(S)}$ are $U_b^{(A)}(\mathfrak{b})$, $U_b^{(N)}(\mathfrak{b})$, and $U_b^{(S)}(\mathfrak{b})$ respectively. For all $\mathfrak{b} \in D_b$, $b^{(A)} \succeq_a^b b^{(N)} \succeq_a^b b^{(S)}$ if and only if $U_b^{(A)}(\mathfrak{b}) \geq U_b^{(N)}(\mathfrak{b}) \geq U_b^{(S)}(\mathfrak{b})$.* $\square$

Proposition 4 *For three agents $b^{(A)}$, $b^{(N)}$, and $b^{(S)}$ represented by $(\beta_{b,1}^{(A)}, \beta_{b,2}^{(A)}, \gamma_{b,1}^{(A)}, \gamma_{b,2}^{(A)}, \lambda_{b,1}^{(A)}, \cdots, \lambda_{b,4}^{(A)})$, $(\beta_{b,1}^{(N)}, \beta_{b,2}^{(N)}, \gamma_{b,1}^{(N)}, \gamma_{b,2}^{(N)}, \lambda_{b,1}^{(N)}, \cdots, \lambda_{b,4}^{(N)})$ and $(\beta_{b,1}^{(S)}, \beta_{b,2}^{(S)}, \gamma_{b,1}^{(S)}, \gamma_{b,2}^{(S)}, \lambda_{b,1}^{(S)}, \cdots, \lambda_{b,4}^{(S)})$, respectively. If all the following conditions hold:*

$$(i)\ \beta_{b,i}^{(A)} > \beta_{b,i}^{(N)} > \beta_{b,i}^{(S)} \quad (for\ \ i = 1\ and\ 2)$$
$$(ii)\ \gamma_{b,1}^{(A)} < \gamma_{b,1}^{(N)} < \gamma_{b,1}^{(S)}$$
$$(iii)\ \gamma_{b,2}^{(A)} > \gamma_{b,2}^{(N)} > \gamma_{b,2}^{(S)}$$
$$(iv)\ \lambda_{b,j}^{(A)} > \lambda_{b,j}^{(N)} > \lambda_{b,j}^{(S)} \quad (for\ \ j = 1, 2\ and\ 3)\ \ and\ \ \lambda_{b,4}^{(A)} < \lambda_{b,4}^{(N)} < \lambda_{b,4}^{(S)}$$

then, $b^{(A)} \succeq_a^b b^{(N)} \succeq_a^b b^{(S)}$. $\square$

Proof. For a b-agent, for all $\mathfrak{b} \in D_b$, $U_b(\mathfrak{b})$ is a non-increasing function. That is, the smaller the bid, the more utility the agent obtains. Let the bid submitted by each b-agent be $\mathfrak{b}^{(A)}$, $\mathfrak{b}^{(N)}$, and $\mathfrak{b}^{(S)}$. From (i) to (iv), we can always get $\mathfrak{b}^{(A)} > \mathfrak{b}^{(N)} > \mathfrak{b}^{(S)}$. That is, $b^{(S)}$ always submits a lower bid compared with $b^{(N)}$ and $b^{(A)}$, and that $b^{(N)}$ is always lower than $b^{(A)}$. Thus $U_b^{(A)}(\mathfrak{b}) \leq U_b^{(N)}(\mathfrak{b}) \leq U_b^{(S)}(\mathfrak{a})$. Based on Proposition 3, we have: $b^{(A)} \succeq_a^b b^{(N)} \succeq_a^b b^{(S)}$.

Now given the fact that different parameters correspond to different attitudes towards risk, the key question is how to choose the appropriate risk attitudes of agents given a particular environment? The rest of this subsection is devoted to answering this question. In particular, the influence of the relation of supply and demand quantity (a key environmental factor) is considered.

Conjecture: The relation of supply and demand quantity influences the performance of agents with different attitudes. If the supply (demand) quantity is greater than the demand (supply) quantity, an s-agent (b-agent) with an averse attitude towards risk can make more profit. $\square$

To test our conjecture, a series of six experiments were conducted. Beside the three aforementioned kinds of agents (risk-averse, risk-seeking, and risk-neutral) two extra kinds of agents are considered: agents between the neutral and averse attitude (weakly averse), and agents between the neutral and risk attitude (weakly risky). These are added in order to make the trend of influence of the risk attitudes to the market supply (demand) more clear. In each session, only one agent uses the FL-strategy, and from session to session, the attitude of the FL-agent varies from risk-averse to risk-seeking. All the other agents utilise one of our benchmark strategies.[8]

Figure 3.5 shows the profit of FL-agents in different sessions. Figure 3.5(a)&(b) show the profit of agents when supply is equal to demand. In these cases, the left-hand bars in each group are always taller than the other bars in the same group. These bars represent the profits of the averse agents. A risk-averse agent is easily satisfied, so it can make transactions quickly and with a high volume. As a result, its profit is high. Thus, in this environmental setting, an averse agent can make more profit, whether it is an s-agent or a b-agent. In Figure 3.5(c), supply is greater than demand, thus an s-agent is in an unfavourable position. The chances of selling a good are small, because there are very few b-agents. So an averse agent makes more profit. This explanation also holds for the b-agents in Figure 3.5(f). Figure 3.5(d) shows the behaviour of agents when supply is less than demand. Here the attitude with which an s-agent performs best varies with the

[8]In fact they use the ZI-strategy which will be described in Section 3.3.3 (this is the earliest and simplest of our benchmark strategies). Thus, if there is one FL s-agent in the market, there are 4 s-agents and 5 b-agents using the ZI-strategy; if there is one FL b-agent, there are 5 s-agents and 4 b-agents using the ZI-strategy.

change of the difference between supply and demand quantity. At the beginning, when the difference is small, the weakly averse agents always make more profit. However with the increase in supply, the trend is that an s-agent with a risk-seeking attitude can make more profit. This is caused by the fact that the market is not very competitive when the difference between supply and demand is small. This also explains the bars for b-agents in the situation when supply is less than demand (Figure 3.5(e)).

In summary, this experiment clearly shows how to choose the suitable attitude for an agent in different situations *if we have knowledge about the real-time supply and demand*. As indicated by our conjecture, an s-agent should be averse when supply is greater than or equal to demand; and the attitude should change from risk-averse to risk-seeking with the increase of demand when the supply is less than demand. For a b-agent, it should be averse when supply is less than or equal to demand; and the attitude should change from risk-averse to risk-seeking with the increase of supply when the supply is greater than demand.

3.3.3 Benchmarking the FL Strategy

Having determined the best parameter configuration with respect to risk in a given environment, this section compares our strategy (with the risk attitude tailored to the environment) with a number of others that have been proposed in the literature. These other strategies represent the most widely cited strategies for agents participating in CDAs. In more detail, the benchmark strategies are:

- **Zero Intelligence (ZI) strategy** [Gode and Sunder, 1993b]. A ZI b-agent submits a bid drawn randomly between outstanding bid ($\mathfrak{b}_o$) and the valuation of its current unit. Similarly, a ZI s-agent submits an ask drawn randomly between the cost of its current unit and outstanding ask ($\mathfrak{a}_o$). This strategy is an extension of the "budget-constraint zero intelligence" trader in the economics literature[9] [Gode and Sunder, 1993b]. In [Gode and Sunder, 1993b], the lower limit that a b-agent submits as a bid is 0 and the upper limit that an s-agent submits as an ask is 1. We believe our extension is appropriate because we are dealing with single unit trades and because the ask (bid) bounding conditions increase the possibility of matching an outstanding bid (ask).
- **Fixed Mark-up (FM) strategy.** An FM b-agent (s-agent) submits the outstanding bid (ask) plus (minus) some predefined mark-up (this is a specialisation of [Preist and van Tol, 1998]). This is a simple strategy because the agent does not need to model other agents and it tries to reduce the ask-bid spread until its cost or valuation is met.

[9] Actually, the ZI-strategy is equivalent to the ZI-C strategy proposed by Gode and Sunder [Gode and Sunder, 1993a].

- **Chris Preist (CP) strategy.** The CP-strategy consists of a small number of heuristics and a learning rule [Preist, 1999]. The heuristics first determine the target profit margin based on the current outstanding bid (ask) and an independent random variable distributed in the range $[0, 0.2]$. Then, given the target, a CP agent does not jump straight to that value, but moves towards the target at a learning rate which determines the speed of the adjustment.
- **Gjerstad-Dickhaut (GD) strategy.** The GD-strategy [Gjerstad and Dickhaut, 1998] is a more sophisticated strategy. A GD agent records all the asks (bids) made in the history H occurring in the last several transactions. From the history, an agent can compute the probability of a bid or ask being accepted. For example, for a buyer,

$$\hat{q}(\mathfrak{b}) = \frac{TBL(\mathfrak{b}) + AL(\mathfrak{b})}{TBL(\mathfrak{b}) + AL(\mathfrak{b}) + RBG(\mathfrak{b})}, \tag{3.12}$$

  where $\hat{q}(\mathfrak{b})$ is the probability of $\mathfrak{b}$ being accepted, $TBL(\mathfrak{b})$ is the number of accepted bids not greater than $\mathfrak{b}$ in H, $AL(\mathfrak{b})$ is the number of asks not greater than $\mathfrak{b}$ in H, and $RBG(\mathfrak{b})$ is the number of rejected bids not less than $\mathfrak{b}$ in H. Then, cubic spline interpolation is used to compute the probability of a given bid being accepted given the history. A GD b-agent submits a bid, $\mathfrak{b}$, which maximises $\pi_b(v - \mathfrak{b})$, where π_b is the belief function of a bid that is accepted, and v is the valuation of the good. Similarly, a GD s-agent submits an ask $\mathfrak{a}$ which maximises $\pi_s(\mathfrak{a} - c)$, where π_s is the belief function of an ask that is accepted, and c is the cost of the good to sell.

To evaluate the behaviour of each agent, we compare their profits in three situations: (i) supply equals demand (Figure 3.6(a)&(b)); (ii) supply is less than demand (Figure 3.6(c)&(d)); and (iii) supply is greater than demand (Figure 3.6(e)&(f)). In each subfigure of Figure 3.6, the horizontal axis shows the supply (demand) quantity and the vertical axis represents the profit of agents using various strategies. There are five curves in each sub-figure and each one represents the profit of one kind of strategy. Given the same supply (demand), the bigger the profit, the better the strategy.

From Figure 3.6, we can see that the FL-agents often obtain higher profits than all the other corresponding agents. The exception is that FL-agents perform slightly worse than the GD-strategy when (i) for the b-agent, supply$=$ 25 and demand=40, 45, and 50, respectively; and (ii) for the s-agent, supply$=$ 25 and demand=30 and 35, respectively. The reason for this inferiority is that our agent cannot adjust its risk attitude during the course of the CDA and the value that is used is based on the experimental results in Section 3.3.2 which is for the general case. However, in most situations, FL-agents can outperform agents using other strategies. We attribute this success to two factors. Firstly, when making a decision in a given situation, an FL-agent considers the outstanding ask, outstanding bid, and reference price (inferred from history) in deciding its next ask (bid).

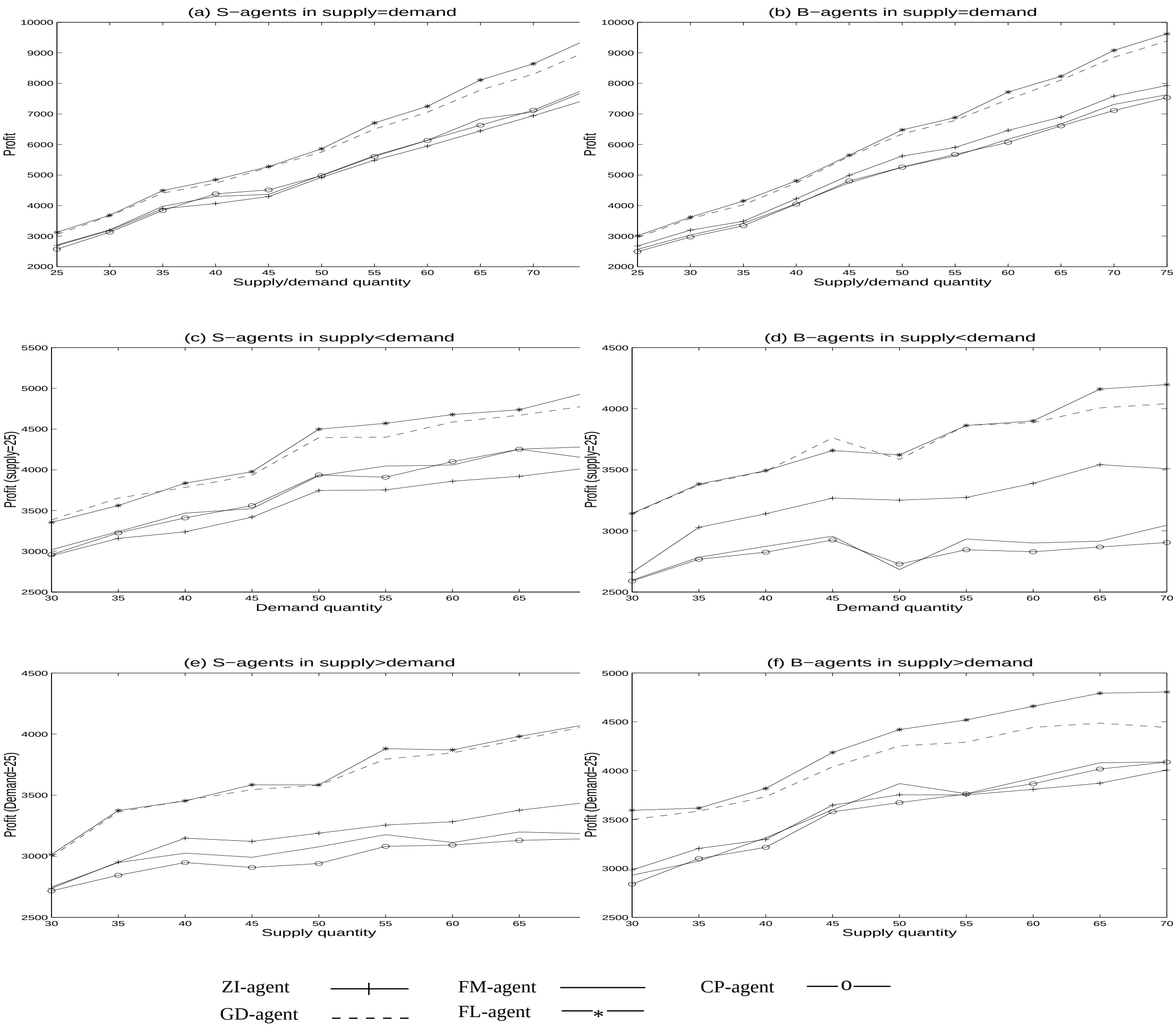


Figure 3.6: Performance of agents with different strategies.
The horizontal axis represents the demand (or supply) configuration of the corresponding market and the vertical axis represents the total profit of the corresponding agent using a specific strategy in one session. There are 5 s-agents and 5 b-agents in each experiment. In (a)&(b), supply is equal to demand, the unit of good that each agent is endowed with to buy (sell) increases from session to session in the range $[5, 15]$, the total supply (demand) is shown in the horizontal axis. In (c)&(d), supply is less than demand, the unit of good that each s-agent is endowed with is fixed to 5, and the unit of good for b-agents increases from session to session in the range $[6, 14]$. In (e)&(f), supply is greater than demand, the unit of good that each b-agent is endowed with is fixed to 5, and the unit of good for s-agents increases from session to session in the range $[6, 14]$.

We believe reference price is a very important factor in bidding, and our strategy is the only one to exploit this information. Secondly, the FL-strategy can dynamically vary the rate of increase (decrease) in bid (ask) according to the prevailing context. Sometimes, for example, an FL-agent can jump from a very low price to a transaction price. This is markedly different from the other strategies which only increase (decrease) their bids gradually.

The performance of the other strategies is statistically worse than our FL-agents. GD-agents behave worse than the FL-strategy although they do maintain a history. However, these agents ignore the outstanding ask (bid), which we believe is one of the most important factors in deciding an agent's next bid (ask). The other three strategies ignore the transaction history. ZI-agents have no knowledge about the auction, they submit their bids (asks) randomly. However, ZI-agents can sometimes deal at a very low bid or high ask. FM-agents and CP-agents can only increase (decrease) their bids (asks) in a fixed step or small varied steps without caring about the outstanding bid (ask). Thus, they miss out on some deals which they should have made.

3.4 Adaptive FL-agents

In the above experiments, the risk attitude of the FL-agent is selected manually based on design time knowledge of the relation between supply and demand. However, in many environments this information is a priori unknown. Also, in an open CDA, the number of agents can be changing continuously as new agents enter the market and the existing ones drop out. Further, the parameters that are suitable in one CDA market may not behave well in others because success is inextricably linked to the strategies of the competitor agents. For all these reasons, we believe it is desirable for an FL-agent to have the ability to automatically adapt itself to its market context (this realises the adaptivity feature we discussed in Section 1.2). Thus this section reports on a number of extensions in this direction that we made to the basic FL-strategy.

3.4.1 Learning Principle for FL-agents

As discussed in Section 3.2.1, each FL-agent has a reference price (P_R) to decide whether it sells (buys) a good at a profitable price. Given this price, an agent can submit an ask (bid) based on its risk attitude (parameters). However, different attitudes can lead to different asks (bids) (Section 3.3.2). Furthermore, even the same asks (bids) have different effects in different environments. Thus an agent needs another measure to decide whether it submits too high a bid or too low an ask. To this end, an agent can observe how frequently it can make transactions. If an s-agent (b-agent) waits too long to conduct a deal, it shows that it should be more averse in the next round if it is to make more transactions. On the other hand, if an s-agent (b-agent) can transact very frequently, it is a sign that its bids (asks) are too high (low). Thus, during the next round of the CDA, the agent should change its attitude in the direction of risk-seeking (hoping it can still make a transaction while increasing its profit). We call this kind of hill-climbing behaviour the *adaptive FL-agent* (denoted A-FL-agent).

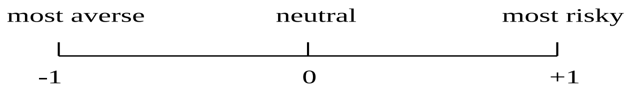


Figure 3.7: Risk Attitude of an agent.

Suppose agent i's attitude is expressed by $A^{(i)}_{attitude}$ which corresponds to a value in $[-1, +1]$ as shown in Figure 3.7. Each value of $A^{(i)}_{attitude}$ corresponds to a group of parameters which define its attitudes towards risk that satisfy Proposition 2 or Proposition 4. Formally, the learning principles for A-FL-agents can be expressed as the rules in Table 3.3 (where δ is the minimum step and $\mathfrak{r}$ ($\mathfrak{r} > 0$) is the learning rate). The terms "waits_long" and "transacts_frequently" are expressed as two fuzzy sets shown in Figure 3.8.

Table 3.3: Learning rules for A-FL-agents.
$A^{(i)}_{attitude}$ denotes the attitude of agent i, $\mathfrak{r}$ is the learning rate, and δ is the minimum step.

IF	agent i waits_long to transact
THEN	$A^{(i)}_{attitude} = A^{(i)}_{attitude} - \mathfrak{r}\delta$
IF	agent i transacts_frequently
THEN	$A^{(i)}_{attitude} = A^{(i)}_{attitude} + \mathfrak{r}\delta$

In this context, the learning rate $\mathfrak{r}$ determines the speed with which the adjustment takes place. Some agents may adapt themselves slowly but steadily, while others may change their attitudes quickly. Thus, we compare three different representative adjustment methods: (i) an agent increases (decreases) at the constant rate minimum step δ, that is, $\mathfrak{r} = 1$; (ii) $\mathfrak{r} = m\delta$, where $m > 1$; that is, the agent increases (decreases) at a bigger

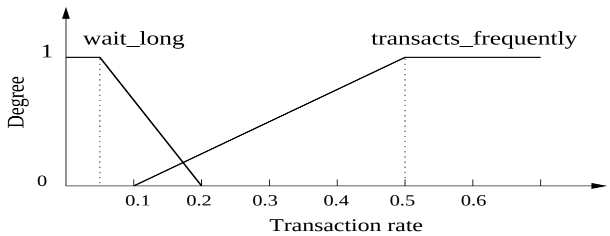


Figure 3.8: Two fuzzy sets for transaction rate.
This rate is calculated by the number of transactions made by an agent divided by the total transaction numbers in the market after the latest change of the agent's attitude towards risk.

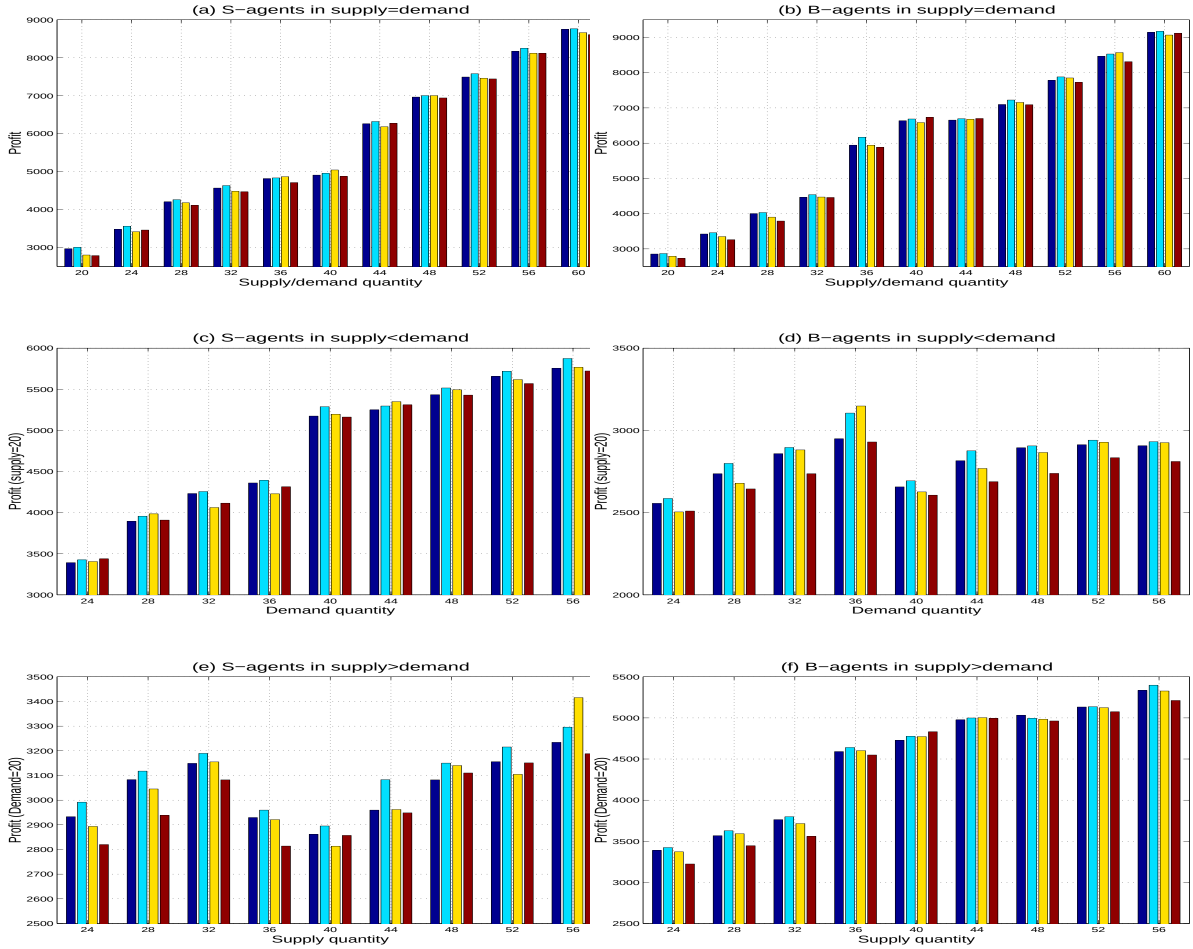


Figure 3.9: Performance of A-FL-agents with different adaptive speeds. Initially, all the adaptive agents are risk-neutral. In each sub-figure, the bars in each group represent the agents (from left to right): FL-agents (without adaptivity), A-FL_1 agents ($\mathfrak{r} = 1$), A-FL_2 agents ($\mathfrak{r} = 5$), and A-FL_3 agents ($\mathfrak{r}$ is drawn randomly from $[1, 10]$).

step than δ; (iii) $\mathfrak{r}$ is an independent random number uniformly distributed in the range $[1, \tau]$, where τ is the maximum adjustment number.

In these experiments we assume there is no abrupt increase or decrease in the supply and demand quantity. That is, over any period the CDA market in each session is relatively stable (*i.e.*, there is a fixed supply and demand quantity). Further, to compare the performance of different learning rates, we compare the three adaptive FL-agents and the FL-agents with the parameters shown by the selection principle in Section 3.3.2. The experiments are conducted in different situations (see Figure 3.9).

Generally, the agents whose learning rate is 1 (the agent which increases (decreases) the attitude value at a small and constant rate) perform best. This is because this kind of agent can fine tune its parameters which avoids over-response to supply (demand) changes in the market. Thus, we choose this kind of learning rate to adjust the attitude

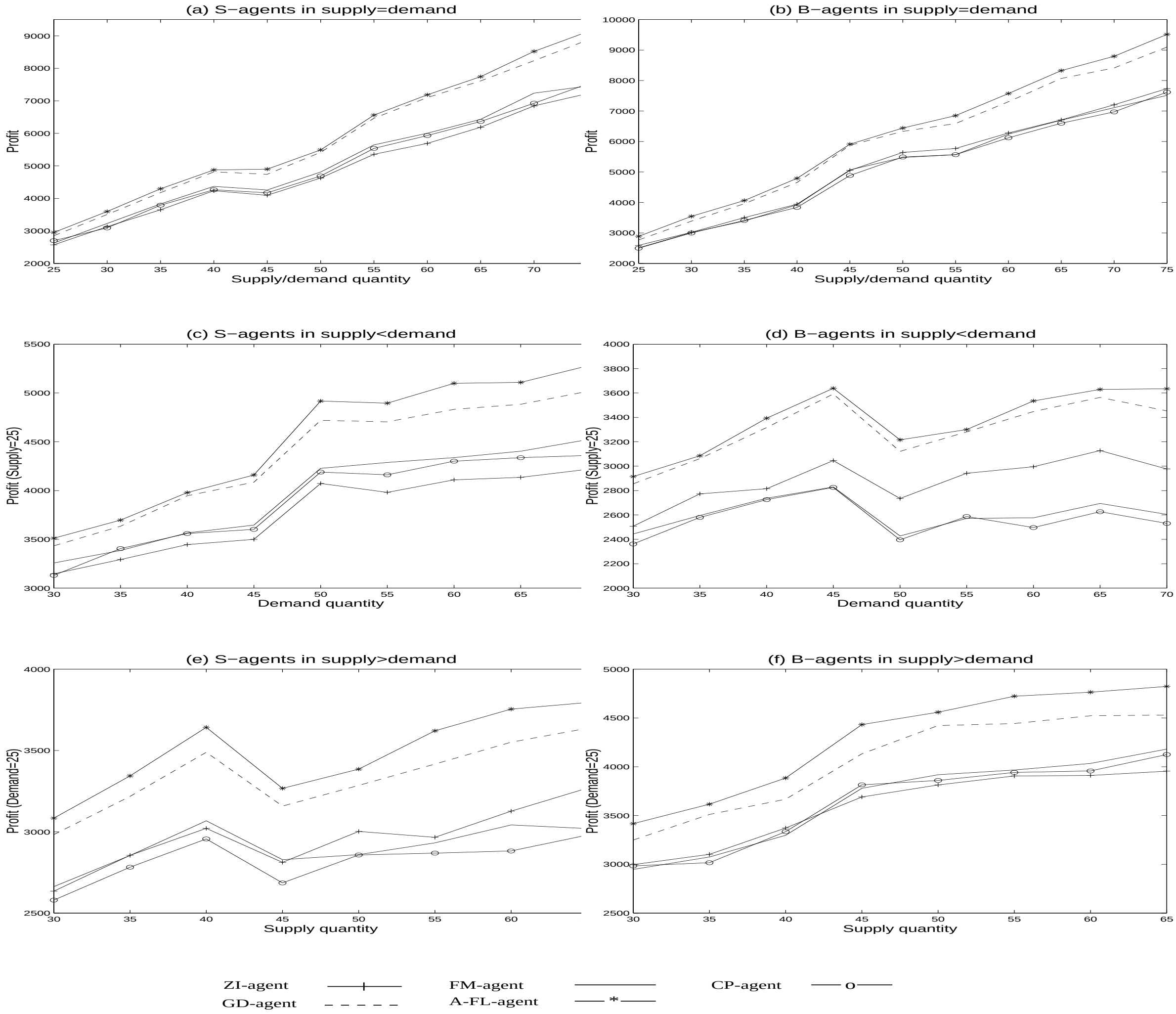


Figure 3.10: Competition of A-FL-agents with other strategy agents.

for the adaptive FL-agents. Also, the adaptive FL-agents with a small learning rate do better than other FL-agents. From Figure 3.9, it can be seen that the A-FL_1 agents obtain a higher profit than the FL-agents. This means that even without the knowledge of supply and demand, the adaptive agent can effectively tailor its strategy to its prevailing circumstances. While this result is promising, there is the caveat assumption that there is no abrupt change in supply or demand. In such circumstances, a learning rate that takes small steps may not be able to respond quickly enough to be effective. However, the best means of dealing with abrupt change is left for future work.

3.4.2 Comparison with the Other Strategies

In this experiment, we compare the adaptive FL-agents with the four benchmark strategies of the previous section (ZI, FM, CP and GD).[10] Figure 3.10 clearly shows that the adaptive strategy is effective. A-FL-agents behave better, sometimes much better, than all

[10]The set-up of this experiment is the same as that of Section 3.3.3.

the other strategies. This is because an A-FL-agent can tailor its bidding behaviour to the prevailing market context.

Besides the profit of the agents, we also investigated the transaction price distribution of each agent. We did this because this metric is a good indication of how consistently an agent performs in a CDA. Table 3.4 shows the transaction price distributions in $1,000$ runs of two agents[11] when supply is equal to demand (Table 3.4 (a)), supply is less than demand (Table 3.4 (b)), and supply is greater than demand (Table 3.4 (c)). In Table 3.4, P_0 is the average equilibrium price of $1,000$ runs, obtained from the supply and demand curves. Accordingly, Q_0 is the average quantities at the equilibrium prices. $\overline{P}$ is the average price for each agent and the Change Rate (CR) shows the percentage of $(\overline{P} - P_0)$ to the equilibrium price P_0. This is a key measure of how well the agent behaves and is calculated in the following way:

$$CR\left(\overline{P}, P_0\right) = \frac{\overline{P} - P_0}{P_0} \times 100\%.$$

Generally, for a b-agent, the lower the average price, the better the strategy; for an s-agent, the higher the average price, the better the strategy. In Table 3.4, the average prices of A-FL b-agents are always the lowest among all the b-agents. This means that A-FL b-agents always pay low prices to acquire goods. Also, the average prices of A-FL s-agents are always the highest among all the s-agents. This means that A-FL s-agents always sell goods at high prices. An agent's Change Rate also gives an indication of how high or low the average transaction price is compared with the equilibrium price of the CDA. For an s-agent, the higher the *CR*, the higher the price at which it sells its goods; for a b-agent, the lower the *CR*, the lower the price at which it buys the goods. Our A-FL s-agents always get the maximum *CR* value and A-FL b-agents always obtain the minimum *CR* value. This means that our adaptive FL-agents buy goods at the lowest average price and sell goods at the highest average price among all the agents using various strategies. Thus, the adaptive FL-agents outperform all the other strategies.

3.4.3 Collective Behaviour of A-FL-agents

Since the A-FL strategy is effective in making good profits in a CDA, we expect many A-FL-agents may appear in a given CDA market. Thus, we need to test the efficiency of a CDA market that is populated with multiple A-FL-agents. In particular, we would like to investigate how the performance of an A-FL-agent changes as the percentage of A-FL-agents in the population increases, and to what extent the efficiency of the CDA market is affected by this change in population.

[11] Here we show the transaction price distribution of two agents. In total, in the experiment shown in Figure 3.10, there are 28 configurations of different market situations. Two agents (one s-agent and one b-agent) from 6 sessions are shown in Table 3.4. To distinguish these two agents, we use $strategy\_s/b\_1$ and $strategy\_s/b\_2$ respectively. For example, $A-FL\_b\_1$ means b-agent 1 utilises A-FL strategy.

Table 3.4: Transaction price distributions of each agent.

$P_0 = 2.50$, $Q_0 = 37$			$P_0 = 2.50$, $Q_0 = 55$		
Agent	$\overline{P}$	CR	Agent	$\overline{P}$	CR
ZI_s_1	2.46	-1.54%	*ZI_s_2*	2.39	-4.39%
FM_s_1	2.49	-0.55%	*FM_s_2*	2.43	-2.65%
CP_s_1	2.47	-1.22%	*CP_s_2*	2.42	-3.03%
GD_s_1	2.58	3.33%	*GD_s_2*	2.53	1.31%
A-FL_s_1	**2.59**	**3.47%**	**A-FL_s_2**	**2.55**	**2.09%**
ZI_b_1	2.58	3.36%	*ZI_b_2*	2.54	1.69%
FM_b_1	2.59	3.66%	*FM_b_2*	2.55	2.08%
CP_b_1	2.59	3.77%	*CP_b_2*	2.56	2.37%
GD_b_1	2.46	-1.46%	*GD_b_2*	2.42	-3.11%
A-FL_b_1	**2.44**	**-2.38%**	**A-FL_b_2**	**2.39**	**-4.53%**

(a) Transaction prices of each agent when supply=demand.

$P_0 = 2.64$, $Q_0 = 25$			$P_0 = 2.92$, $Q_0 = 29$		
Agent	$\overline{P}$	CR	Agent	$\overline{P}$	CR
ZI_s_1	2.60	-1.63%	*ZI_s_2*	2.81	-3.91%
FM_s_1	2.62	-0.60%	*FM_s_2*	2.85	-2.24%
CP_s_1	2.62	-0.94%	*CP_s_2*	2.84	-2.75%
GD_s_1	2.70	2.42%	*GD_s_2*	2.94	0.85%
A-FL_s_1	**2.72**	**2.99%**	**A-FL_s_2**	**3.00**	**2.77%**
ZI_b_1	2.69	1.88%	*ZI_b_2*	2.92	0%
FM_b_1	2.72	2.89%	*FM_b_2*	2.98	1.92%
CP_b_1	2.73	3.22%	*CP_b_2*	3.00	2.64%
GD_b_1	2.60	-1.56%	*GD_b_2*	2.82	-3.51%
A-FL_b_1	**2.58**	**-2.19%**	**A-FL_b_2**	**2.81**	**-3.77%**

(b) Transaction prices of each agent when supply<demand.

$P_0 = 2.36$, $Q_0 = 25$			$P_0 = 2.12$, $Q_0 = 29$		
Agent	$\overline{P}$	CR	Agent	$\overline{P}$	CR
ZI_s_1	2.35	-0.57%	*ZI_s_2*	2.15	1.28%
FM_s_1	2.36	0%	*FM_s_2*	2.13	0.67%
CP_s_1	2.33	-1.12%	*CP_s_2*	2.12	-0.11%
GD_s_1	2.43	2.93%	*GD_s_2*	2.22	4.79%
A-FL_s_1	**2.46**	**4.06%**	**A-FL_s_2**	**2.24**	**5.79%**
ZI_b_1	2.44	3.43%	*ZI_b_2*	2.26	6.56%
FM_b_1	2.44	3.43%	*FM_b_2*	2.23	5.37%
CP_b_1	2.45	3.63%	*CP_b_2*	2.25	6.02%
GD_b_1	2.36	-0.14%	*GD_b_2*	2.15	1.49%
A-FL_b_1	**2.32**	**-1.50%**	**A-FL_b_2**	**2.12**	**-0.23%**

(c) Transaction prices of each agent when supply>demand.

We test the collective behaviour of A-FL-agents in situations where the quantity of demand is (i) greater than, (ii) equal to, and (iii) less than the quantity of supply, respectively. For each situation, the experiment is composed of multiple sessions. Figure 3.11 shows the results of the profits of A-FL-agents in different sessions under different situations.[12] The horizontal axes represent the session numbers, and the vertical axes represent the sum of the profits in $1,000$ runs of each agent. The curves show each agent's profit in different sessions.

In different sessions, we only change the strategy of one agent. Actually, we increase the number of agents that employ the A-FL strategy by one in each session. Take Figure 3.11(a) as an example. In this situation, the demand is greater than supply, thus the competition among A-FL b-agents is highlighted. There are 10 b-agents and 8 s-agents and each agent has three units of goods to buy (sell). In session 1, only b-agent b_1 uses the A-FL strategy; in session 2, we change the strategy of b-agent b_2 to the A-FL strategy, while fixing all the other parameters; in session 3, the strategy that b_3 uses is changed; and finally in session 10, all the b-agents use the A-FL strategy.

As shown in Figure 3.11(a), (b) and (c), the profit of A-FL b-agent b_1 decreases initially and then increases steadily. The trend of the change of the profit of agent b_2 and agent b_3 is similar. This phenomenon can be explained as follows. For agent b_1, with an increasing number of A-FL-agents, more A-FL-agents compete with b_1, and thus the profit of b_1 decreases. When the number of A-FL-agents is sufficiently large,[13] A-FL-agents make the transaction prices of the market low. The same explanation holds for A-FL s-agents. Thus, the reference prices of each agent decrease from session to session and all the A-FL-agents make greater profits.

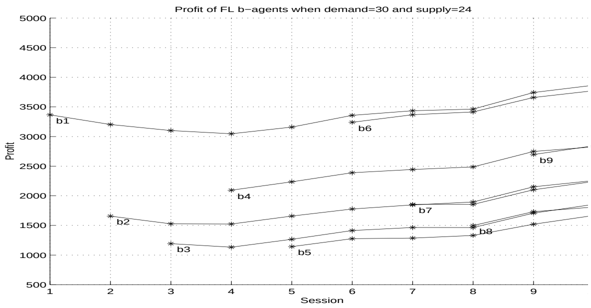
Since A-FL-agents make more profits in the long term, another question arises. How will the CDA market as a whole be affected with an increasing population of A-FL-agents? We evaluate the profit obtained by all agents in the market divided by the surplus when agents trade their goods at the equilibrium price to determine the efficiency of the market.[14] Tables 3.5 to 3.7 summarise this efficiency data of the collective profit of all the agents and the market efficiency from experiments of different sessions. As can be seen, for CDA markets with varying numbers of A-FL b-agents,[15] both the total profit and efficiency of the market increase initially and then decrease little by little. For a CDA market with varying numbers of A-FL s-agents, the trend is not so obvious. The market becomes less efficient due to the increase in strategic reasoning of the A-FL-agents. However, we can see that from the session when no agent uses the A-FL strategy to the session when all the b-agents and s-agents use the A-FL strategy, the efficiency of

[12] In order to investigate the performance of A-FL-agents, the profit of non-A-FL agents is not shown in Figure 3.11. These agents use various randomly selected strategies from our set of benchmarks.

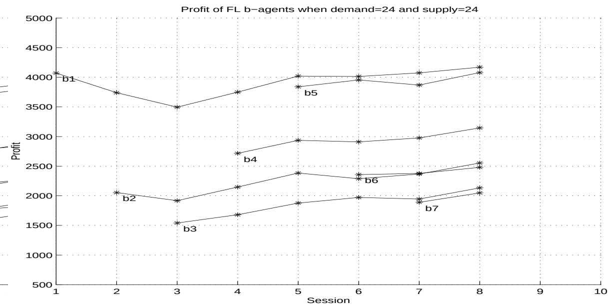
[13] In the experiment, this number is 50% of the total number of b-agents or s-agents.

[14] The efficiency of the market is obtained by the following formula (based on the intuitions described in [Gjerstad and Dickhaut, 1998]): $Efficiency = \frac{\sum_i p_a^{(i)}}{\sum_i p_e^{(i)}}$, where $p_a^{(i)}$ is the actual profit of the agent i in a session ($1,000$ runs); $p_e^{(i)}$ is the profit of the agent if the agent i trades its goods according to the equilibrium price given their costs and valuations of the goods in a session.

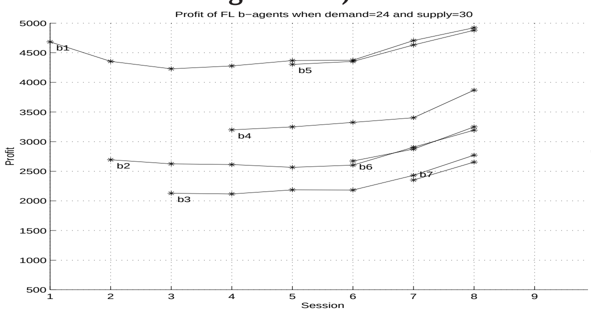
[15] Note that the total number of agents in each set-up is fixed; that is, in Table 3.5, there are 8 b-agents and 8 s-agents; in Table 3.6, there are 8 b-agents and 10 s-agents; in Table 3.7, there are 10 b-agents and 8 s-agents.



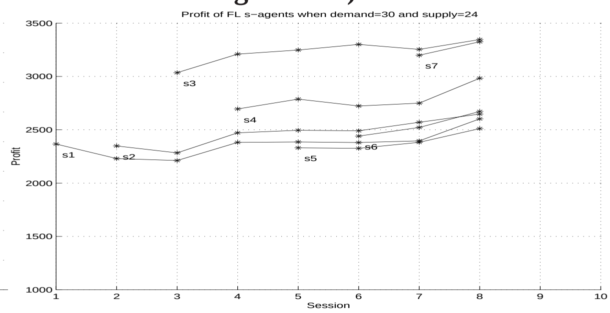
(a) Profits for A-FL b-agents when demand>supply. (number of b-agents is 10 and number of s-agents is 8)



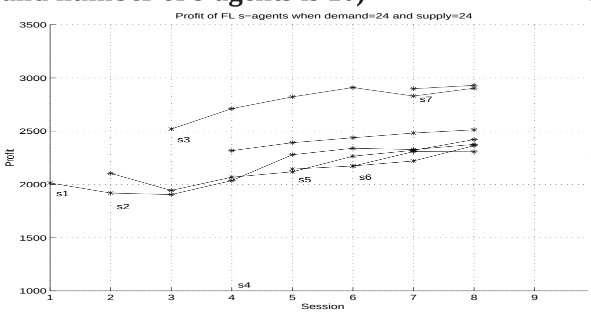
(b) Profits for A-FL b-agents when demand=supply. (number of b-agents is 8 and number of s-agents is 8)



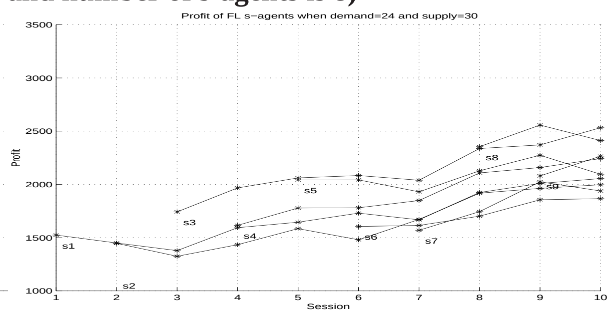
(c) Profits for A-FL b-agents when demand<supply. (number of b-agents is 8 and number of s-agents is 10)



(d) Profits for A-FL s-agents when demand>supply. (number of b-agents is 10 and number of s-agents is 8)



(e) Profits for A-FL s-agents when demand=supply. (number of b-agents is 8 and number of s-agents is 8)



(f) Profits for A-FL s-agents when demand<supply. (number of b-agents is 8 and number of s-agents is 10)

Figure 3.11: Profits for A-FL-agents in different sessions.

the market does not decrease significantly. That is, even in the worst case the efficiency is still reasonably high. The market with all A-FL-agents is also investigated. In this market, all the agents use the A-FL strategy, and the efficiency is 85.45% for a market with 5 s-agents and 5 b-agents. This figure is still reasonable with respect to experiments shown in Tables 3.5 to 3.7. Thus we can conclude that widespread adoption of the A-FL strategy does not lead to a significant deterioration in the effectiveness of the overall market.

Table 3.5: Efficiency statistics when demand=supply.

Number of A-FL b-agent	Profit	Effici -ency	Number of A-FL s-agent	Profit	Effici -ency
0	29,463	96.92%	0	29,463	96.92%
1	29,267	96.27%	1	29,280	96.32%
2	29,560	97.24%	2	29,155	95.91%
3	29,575	97.29%	3	29,283	96.33%
4	29,446	96.86%	4	29,136	95.84%
5	29,103	95.73%	5	28,781	94.67%
6	29,086	95.68%	6	28,332	93.20%
7	29,028	95.49%	7	27,510	90.49%
8	28,449	93.58%	8	26,045	85.67%

Table 3.6: Efficiency statistics when demand<supply.

Number of A-FL b-agent	Profit	Effici -ency	Number of A-FL s-agent	Profit	Effici -ency
0	31,972	94.04%	0	31,972	94.04%
1	32,592	95.86%	1	31,776	93.46%
2	32,971	96.97%	2	31,751	93.39%
3	33,071	97.11%	3	32,140	94.53%
4	32,994	97.04%	4	32,020	94.18%
5	32,909	96.79%	5	32,201	94.71%
6	32,899	96.76%	6	32,081	94.36%
7	32,941	96.89%	7	31,971	94.03%
8	32,860	96.65%	8	30,843	90.71%
			9	29,454	86.63%
			10	28,267	83.14%

3.5 Related Work

There are a number of strands of work that are related to what we have described in this chapter. Firstly, the work on bidding strategies for various forms of auctions. Secondly, the work on using fuzzy techniques to manage an agent's interactions (see Section 2.2.5 for the discussion of bilateral negotiations). Finally, alternatives to fuzzy reasoning for coping with the uncertainties in bidding.

Table 3.7: Efficiency statistics when demand>supply.

Number of A-FL b-agent	Profit	Efficiency	Number of A-FL s-agent	Profit	Efficiency
0	31,444	86.86%	0	31,444	86.86%
1	31,504	87.03%	1	31,345	86.59%
2	31,480	86.96%	2	31,334	86.56%
3	31,448	86.87%	3	31,285	86.42%
4	31,448	86.87%	4	31,195	86.17%
5	31,316	86.51%	5	31,045	85.76%
6	31,029	85.72%	6	30,808	85.10%
7	30,998	85.63%	7	30,487	84.22%
8	30,961	85.53%	8	29,041	80.22%
9	30,598	84.52%			
10	30,332	83.79%			

Non-cooperative game theory is an important tool for analysing strategic interactions between agents [Kreps, 1990]. However, one of its weaknesses is that the theory is only suitable for highly stylised, simple settings [Jennings et al., 2001], thus a clear game-theoretic solution to the CDA problem is not possible. The Recursive Modelling Method [Vidal and Durfee, 1996] has been proposed as an approach for an agent to reason about other agents and generate an appropriate strategy for negotiation. However, in most practical cases, the agent can only build finite nesting models due to the limitation of acquiring knowledge. Thus with this approach, not all the information in the recursive model may be relevant to the agent and it is possible that little or no information may be available for the agent to use. Park, Durfee and Birmingham [Park et al., 1999] propose the adaptive agent bidding strategy (called the p-strategy) based on stochastic modelling for a CDA. The idea of the p-strategy is to model the auction process using a Markov Chain (MC). However, in many cases, it is hard to obtain the probability values required for the MC model, such as the transition probabilities and the probabilities of success and failure for particular trading actions. Moreover, the computation involved in this approach is large. Badea [Badea, 2000] applied Inductive Logic Programming (ILP) to induce trading rules for a CDA. He first identified buy (sell) opportunities from historical market data. Then, these buy (sell) opportunities are input as examples to an ILP learner to produce understandable rules. However, this learning strategy relies heavily on historical data which is often not available in the contexts we consider.

Finally, we consider the alternatives to fuzzy reasoning for handling uncertainty in agent interactions (see [Luo et al., 2001] for a comprehensive survey about handling uncertainty in agent systems). As stated, we chose fuzzy logic based methods because they have proven to be a practicable solution in solving decision making problems under uncertainty (e.g., [Fraichard and Garnier, 2001, Yao and Yao, 2001, Tan and Tang, 2001, Mohammadi et al., 2000]). Fuzzy rules are the most visible manifestation of this approach and have been successfully used in industrial applications, manufacturing, process control, automotive control, and financial trading [Yen, 1999]. There are, however, alternative techniques for handling uncertainties. For example, the possibility based approach [Giménez-Funes et al., 1998, Matos and Sierra, 1998] has been used to perform multi-agent reasoning under uncertainty for bilateral negotiation. In this work, uncertainties due to the lack of knowledge about other agents' behaviours are modelled by possibility distributions. Based on information from a case base of previous negotiation behaviours, the possibility distributions are generated by choosing the most similar situation to the current context and the most similar price from the case base. Since this approach relies on a case base, it is unclear what would happen if no highly similar situations were available. Moreover, even if a similar case exists, it is possible that the strategy used successfully in that situation does not work in the current environment due to the variety of competitors. The Bayesian learning method [Zeng and Sycara, 1998] has also been used to explicitly model multi-issue negotiation in a sequential decision making model. In this work, a Bayesian network is used to update the knowledge and belief each agent has about the environment and other agents, and offers and counter-offers between agents during bilateral negotiations are generated based on Bayesian probabilities. However, this method is inappropriate in our context because assigning prior probabilities of a bid (ask) being accepted is difficult given the dynamism and uncertainty of the CDA context.

3.6 Summary

This chapter developed new algorithms that guide an agent's buying and selling behaviour in a CDA. The FL-strategy uses heuristic fuzzy rules and a fuzzy reasoning mechanism to decide what bids or asks to place. We then extended this strategy so that the agent could adapt its bidding behaviour to its prevailing market context. In both cases we benchmarked the performance of our algorithm against the most prominent alternatives available in the literature. This evaluation showed the superior performance of our method. This result is especially promising since the benchmark strategies have been shown to outperform human bidders in experimental settings [Kephart, 2002]. Speaking more generally, we also believe that the development of efficient and practicable algorithms for bidding behaviour increase the opportunities of using CDAs as the auction protocol for on-line marketplaces. We, therefore, view our contribution as an important step in this direction.

In more detail, the experiments in Section 3.3.2 show how to select the appropriate risk attitude for an agent in different situations. The result is consistent with our conjecture: if supply (demand) quantity is greater than demand (supply) quantity, an s-agent (b-agent) with an averse attitude towards risk can make more profit. This is also consistent with our discussions of the risk attitude in Section 1.2. Based on this selection principle, the experiment in Section 3.3.3 shows that the FL-strategy outperforms some of the most commonly used bidding strategies in a range of situations. Since agents often have no prior knowledge of the relation between supply and demand, it is not always possible to tell in advance what kind of attitude an agent should have. Thus adaptive FL-agents are introduced (recall that we discussed adaptation in Section 3.3.3) which can tailor their strategy to the supply (demand) of the market. Through the experiments in Section 3.4.1, we find that the learning rate which is adjusted in small steps behaves best in an environment in which the supply and demand do not change abruptly. The experiments in Section 3.4.2 show that A-FL-agents always outperform other benchmarking strategy agents in various situations. The transaction price distribution of agents using different strategies shows that an A-FL-agent always sells (buys) goods at higher (lower) prices than agents using other strategies. Finally, in Section 3.4.3 we investigate to what extent the behaviour of A-FL-agents and the efficiency of the CDA market are affected by the increasing use of A-FL-agents. This investigation reveals that the profit of an individual A-FL-agent decreases at first and then increases steadily. We also show that with an increase in the number of A-FL-agents, the efficiency of the market is not significantly affected.

As well as being effective, we believe the FL strategy is practical for building autonomous agents for CDAs. The strategy we employ is intuitive and its embodiment in fuzzy rules means that it should be readily comprehensible to the agent's owner (as have other similar applications of fuzzy rules [Sosnowski, 2000, Yam and Koczy, 2000]). Moreover, the information required by the strategy can be readily obtained by monitoring market activities, such as the outstanding ask, the outstanding bid, and the accepted bids or asks in past transactions. In particular, this procedure does not require any information of the cost or valuation of other agents (cf. some of the approaches discussed in Section 3.2).

Having shown that agents can be developed for a particular auction setting, we now turn to the more complex problem of an agent bidding across multiple, concurrent auctions. However, this CDA work brings forward a number of important intuitions and insights (as well as specific technologies) to the multiple auction setting. Firstly, and most directly, the idea of using a reference price and of fuzzifying the relation with this reference price is used in the entertainment auctions of the trading agent competition (which are a CDA). Secondly, the idea of exploiting fuzzy reasoning techniques is also adopted in the hotel auctions of the competition, where there is a need for an efficient reasoning procedure. Finally, there is the importance of adapting bidding behaviour to the prevailing context in order to cope with dynamics and unpredictability.

Chapter 4

An Adaptive Trading Agent for Multiple Interrelated Auctions

Given the potential and the importance of using agents in on-line auction settings, there has been considered research endeavour in developing bidding strategies for different types of agents in different types of auctions (see Sections 3.5 and 2.2.5 for more details). Therefore, in order to develop a means of comparing and evaluating this work, it was decided to establish an International Trading Agent Competition (TAC) (similar in spirit to other initiatives such as RoboCup,[1] RoboCupRescue[2] and the Planning competition[3]). In this competition, software agents compete against one another in 28 simultaneous auctions in order to procure travel packages (flights, hotels and entertainment) for a number of customers (see Section 4.1 for more details of the roles).

The TAC has been set up so that there is no optimal bidding strategy that is guaranteed to always win. This is because an agent's decision making in the TAC involves uncertainty caused by the random features of the game, the opponents' strategies and the particular combination of opponents. Against this background, this chapter reports upon the design and implementation of our particular trading agent (called **SouthamptonTAC**) which participated in both the competition in 2001 (TAC-01) and in 2002 (TAC-02).

SouthamptonTAC was one of the most successful agents in both competitions (see Section 4.3) and this chapter details its design and implementation and evaluates when and why it is successful. In more detail, SouthamptonTAC is an adaptive agent that varies its bidding strategy according to its perception of the prevailing market conditions. Building upon the success of the FL strategy in Chapter 4, it uses fuzzy reasoning techniques to predict closing prices of the auctions, fuzzy recognition to assess the degree

[1] An international project that uses soccer as a central topic, see http://www.robocup.org for more details.

[2] RoboCupRescue is a new research domain which targets search and rescue in large scale disasters (such as earthquakes), see http://www.r.cs.kobe-u.ac.jp/robocup-rescue/index.html for more details.

[3] The International Planning Competition aims to provide a forum for empirical comparison of planning systems, see http://www.dur.ac.uk/d.p.long/competition.html for more details.

of competitiveness in the prevailing market context, and fuzzy set technique to control bidding behaviour.

This work advances the state of the art in two main ways. In terms of the TAC itself, we developed novel reasoning models and prediction methods that enable an agent to bid across multiple heterogeneous auctions that have inter-dependencies. Through the competition and our systematic evaluation this reasoning mechanism is shown to be both highly effective and practical (it has to operate in a time-constrained environment and has to cope with the uncertainty of operating over the Internet with its concomitant latency problems). In more general terms, we believe that a number of the technologies we developed can be used in other complex auction settings and our insights and experiences about building a successful trading agent will also transfer (see the discussion in Section 4.5 for more details). SouthamptonTAC addresses all the common issues discussed in Section 1.2: price prediction (Section 4.2.7), adaptation (Section 4.2.8), flexible bidding (Section 4.2.6), risk attitude adjusting (Section 4.2.8), and trade-off attributes (Section 4.2.5).

The remainder of this chapter is organised as follows. Section 4.1 describes the trading agent competition. Section 4.2 presents the details of the SouthamptonTAC agent. Section 4.3 evaluates the performance of SouthamptonTAC. Section 4.4 discusses the related work. Finally, Section 4.5 concludes.

4.1 The Trading Agent Competition

TAC-01 and TAC-02 involved 27 and 26 agents respectively, developed by universities and research labs from around the world [Wellman et al., 2002, Greenwald, 2002]. In each TAC trading game, there are 8 software agents (entrants to the competition) that compete against each other in a variety of auctions to assemble travel packages for their individual customers according to their preferences for the trip.[4] A valid travel package for an individual customer consists of (i) a round trip flight during a 5-day period (between TACtown and Tampa) and (ii) a stay at the same hotel[5] for every night between their arrival and departure dates. Moreover, arranging appropriate entertainment events during the trip increases the utility for the customers. The objective of each agent is to maximise the total satisfaction of its 8 customers (*i.e.*, the sum of the customers' utilities). Customers have individual preferences over which days they want to be in Tampa, the type of hotel they stay in, and which entertainment they want to attend. This data is randomly generated by the TAC server in each game (see Table 4.1 for an example).

[4] These packages are assembled by the agent bidding in a number of auctions in which the other bidders are other competition entrants.

[5] Customers are not allowed to change their hotels during the stay.

Table 4.1: SouthamptonTAC's customer preferences for game tac-5722.
PAD and PDD stand for preferred arrival and preferred departure date. HV stands for the reservation value of staying in the Tampa Tower hotel, and WV, PV and MV stand for the utility associated with attending Alligator Wrestling, the Amusement Park and the Museum.

Customer	PAD	PDD	HV	WV	PV	MV
1	Day 3	Day 5	80	178	183	136
2	Day 3	Day 4	129	165	134	36
3	Day 1	Day 3	104	131	110	109
4	Day 4	Day 5	146	27	22	28
5	Day 3	Day 4	80	126	33	81
6	Day 2	Day 5	136	191	143	24
7	Day 3	Day 4	92	180	63	154
8	Day 1	Day 4	148	31	7	177

Each agent communicates with the TAC server through a TCP-based agent programming interface in order to get current market information and to place its bids. An individual game lasts 12 minutes and involves 28 auctions. Each of the three good types are traded in an auction with different rules:[6]

- *Flights.* TACAIR is the only airline selling flights (placing asks). Tickets for these flights are unlimited and are sold in *single seller auctions*. There are 8 such auctions (TACtown to Tampa (day 1 to 4) and back (day 2 to 5)). Flight ask prices update randomly, every 24 to 32 seconds, by a value drawn from a range determined by the elapsed auction time and a randomly drawn value. Flight auctions clear continuously during the game. Thus, any buy bid an agent makes that is not less than the current ask price will match immediately at the ask price. Those bids not matching immediately remain in the auction as standing bids.
- *Hotels.* There are two hotels: Tampa Towers (T) and Shoreline Shanties (S). T is nicer than S. Hotel rooms are traded in 16th price multi-unit English auctions[7]. Overall, there are 8 hotel auctions (for each combination of hotel and night apart from the last one), that close randomly one by one at the end of every minute after the 4th. A hotel auction clears and matches bids when it closes (*i.e.*, 16 rooms are sold at the 16th highest price). While a given auction is open, its ask price is the current 16th highest price and this price is updated immediately in response to new bids. The price of other bids, such as the highest bid, is not known by agents. No withdrawal of hotel bids is allowed. Suppose the current ask price is a, when an agent submits a new bid, two conditions must be satisfied for it to be accepted: (i) it must offer to buy at least one unit at a price of $a+1$ or greater; (ii) if the agent's current bid would have resulted in the purchase of q units in the current state, the new bid must offer to buy at least q units at $a+1$ or greater.

[6] For full details, see http://www.sics.se/tac.
[7] This differs from a standard English auction where 16 units of goods are sold at the 16th highest price

Table 4.2: Market state of all auctions in game tac-5722.
The remaining time in the game is 6 minutes 42 seconds.

Auction	Bid Quote	Ask Quote	Status
Alligator Wrestling on day 1	1.1	86	running
Alligator Wrestling on day 2	70	150	running
Alligator Wrestling on day 3	65	139.38	running
Alligator Wrestling on day 4	66	107.51	running
Amusement Park on day 1	66	70	running
Amusement Park on day 2	66	90	running
Amusement Park on day 3	84	119	running
Amusement Park on day 4	50	89.74	running
Museum on day 1	50	99	running
Museum on day 2	50	146.73	running
Museum on day 3	50	175	running
Museum on day 4	50	80	running
Inflight day 1	–	328	running
Inflight day 2	–	337	running
Inflight day 3	–	381	running
Inflight day 4	–	299	running
Outflight day 2	–	400	running
Outflight day 3	–	370	running
Outflight day 4	–	434	running
Outflight day 5	–	270	running
Tampa Towers Hotel day 1	0	22.35	closed
Tampa Towers Hotel day 2	0	59.49	running
Tampa Towers Hotel day 3	0	33.60	running
Tampa Towers Hotel day 4	101	101.5	running
Shoreline Shanty day 1	0	0	running
Shoreline Shanty day 2	33	42	running
Shoreline Shanty day 3	22.2	29	running
Shoreline Shanty day 4	1	1	closed

- *Entertainment.* Each agent is randomly endowed with 12 entertainment tickets at the beginning of the game. All agents can trade their tickets in CDAs. Overall, there are 12 CDAs (for each kind of entertainment for each of days 1 to 4). Bids match at the price of the standing bid in the CDA. An entertainment package is feasible if none of the tickets are for events on the same day and all the tickets coincide with the nights the customer is in town. No additional utility is obtained for a customer attending the same type of entertainment more than once during the trip.

By means of illustration, Table 4.2 gives the market running state of all the auctions at a single moment in time of the game in tac-5722. A customer's utility from a valid travel and entertainment package[8] is given by:

$$\mathit{Utility} = \mathit{1000} - \mathit{TravelPenalty} + \mathit{HotelBonus} + \mathit{FunBonus},$$

[8] An invalid travel package receives zero utility.

Table 4.3: SouthamptonTAC's customer allocation from game tac-5722.
P, M, W stand for Alligator Wrestling, Amusement Park and Museum and the following number indicates the date of the entertainment.

Customer	AD	DD	Hotel	Entertainment	Utility
1	Day 3	Day 5	S	P3, M4	1319
2	Day 3	Day 4	T	P3	1263
3	Day 1	Day 3	T	W2, M1	1344
4	Day 4	Day 5	S	None	1000
5	Day 3	Day 4	T	W3	1206
6	Day 2	Day 5	S	W4, P2	1334
7	Day 3	Day 5	S	W3, M4	1234
8	Day 1	Day 4	T	M1	1325
				Total utility:	10025

where $TravelPenalty = 100 * (|AD - PAD| + |DD - PDD|)$ (here AD and DD are the customer's actual arrival and departure dates), *HotelBonus* is the bonus if the customer stays in T, and *FunBonus* is the sum of the reservation values of all the entertainment a customer receives. To illustrate this, the allocations and scores for SouthamptonTAC, given the preferences in Table 4.1, are shown in Table 4.3. For example, the utility of customer 3 is obtained by the following:

$$TravelPenalty = 100 * (|AD - PAD| + |DD - PDD|) = 0,$$
$$HotelBonus = 104,$$
$$FunBonus = 131 + 0 + 109 = 240,$$
$$Utility = 1000 - 0 + 104 + 240 = 1344.$$

At the end of each game, the TAC scorer (on the TAC server) allocates the agent's travel goods to its individual customers optimally. The value for a particular allocation is the sum of the individual customer utilities (*e.g.* 10025). The agent's final score is then the value of this allocation minus the cost of procuring the goods. For example, the agent's cost of obtaining the goods in game tac-5722 is shown in Table 4.4. Thus the score in this game is $10025 - 5767.30 = 4257.70$.

Designing a bidding strategy for the TAC auction context is a challenging problem.

- *There are inter-dependencies between auctions*. That is, what goods to buy and how many to buy in one auction are relative to the progress of other auctions. These inter-dependencies exist between different kinds of auctions. For example, flights will be useless if the hotel rooms are not available and if no customer stays in Tampa on a particular day, the entertainment ticket on that day will be useless. The inter-dependence also exists between different dates within the same kind of auction. For example, customers must stay in the same hotel during their stay.

Table 4.4: Expenditure for SouthamptonTAC in tac-5722.
A negative number means the agent obtains the indicated amount of utility by selling the good.

Good	Day 1	Day 2	Day 3	Day 4	Day 5	Total
AlligatorWrestling	0	1 at 91.9	1 at 108.9 1 at 100.5	1 at 120.5	0	421.80
AmusementPark	0	-2 at 80 -1 at 139.5	0	0	0	-299.5
Museum	-1 at 89	0	0	0	0	-89
Inflight	1 at 282 1 at 340	1 at 314	3 at 341 1 at 381	1 at 277	0	2617
Outflight	0	0	1 at 390	2 at 345 1 at 415	2 at 272 1 at 273	2585
TampaTowers Hotel	2 at 22.35	2 at 120	3 at 33.6	0	0	385.5
ShorelineShanty Hotel	4 at 1	1 at 47.5	3 at 30	5 at 1	0	146.5
					Total cost:	5767.3

Thus if a customer stays in T1,[9] the agent will also need to bid in the auctions of other days for T. Finally, inter-dependencies exist between the auctions of the same day, same kind counterpart auctions.[10] For example, if the price of T1 is high, the customer can change to S1.

- *The bidding involves uncertainty*. For example, flight prices start randomly and change continuously in a random fashion and one randomly selected hotel auction closes from the 4th to 11th minute.
- *The bidding involves trade-offs*. For example, in flight auctions, if an agent buys all the flight tickets very early, it may fail to buy the necessary hotel rooms that the flights require, while the flight prices may be quite high if it buys the flights later.

4.2 SouthamptonTAC

Our agent design for TAC-01 and TAC-02 is broadly similar. However we describe in this section SouthamptonTAC-02 since this is the agent built upon our experiences in TAC-01. The main differences between the two are as follows:

- SouthamptonTAC-02 is an adaptive agent that varies its bidding strategy according to its perception of the prevailing market conditions (see Section 4.2.3 for details). SouthamptonTAC-01 had a fixed strategy that it used in all contexts.
- SouthamptonTAC-02 does hotel closing price prediction differently. Although they both use the same basic technique, SouthamptonTAC-02 has two rule bases for predicting prices when the counterpart auction has closed (one for when it has just closed and one for when it has been closed for a longer period of time).

[9] We will use the abbreviation Tn and Sn ($1 \leq n \leq 4$) for staying in the indicated hotel on a particular day n.

[10] For the auction of the same day, T and S are called their counterpart auctions. For example, the counterpart auction of T1 is S1 and the counterpart of S1 is T1.

SouthamptonTAC-01 only had one such rule base (see Section 4.2.7 for more details).

We now deal, in turn, with each of the main components of our agent.

4.2.1 Classifying TAC Environments

Our post hoc analysis of the TAC-01 [He and Jennings, 2002, He and Jennings, 2003] shows that an agent's performance depends heavily on the risk attitude of its opponents. Here a *risk-averse* agent is one that buys a small number of flight tickets at the beginning of the game and that bids for hotels according to the situation as the game progresses. This kind of agent is highly flexible and copes well when there is a significant degree of competition and the hotel prices are high (see below). In contrast, a *risk-seeking* agent buys a large number of flight tickets at the beginning of the game and seldomly changes the travel plan of its customers during the game. This kind of agent does well in environments in which hotels are cheap. For example, when a hotel price goes up sharply, a risk-averse agent would stop bidding on that hotel (changing the stay to a counterpart hotel or reducing the trip period) (see Section 4.3.2). In contrast, a risk seeking agent will insist on bidding for that hotel, although the price is very high. In so doing it hopes that the price will eventually stabilise (hence the risk). The consequence of this variety is that for broadly the same situation, different agents can bring about widely varying final prices. Based on the analysis reported in [He and Jennings, 2002], we identify the following types of TAC environment:

- *Competitive environments* where the prices of the hotels are (very) high. This is caused by (a) the high bid prices that agents place; (b) the fact that some agents insist on bidding for hotels even when their ask price becomes high; and (c) the fact that some agents increase their bids sharply rather than gradually. For example, in game 4594, the prices of T (S) are (in the increasing order of day): 5 (6), 238 (557), 155 (102) and 40 (11). For most customers in this game, it is beneficial for an agent to reduce the stay to a single day (either day 1 or day 4). To achieve this, however, the agent needs to be flexible. Specifically, it cannot buy all the flights at the very beginning of the game, otherwise, when the hotel prices rise to high values, it has to give up the travel package for some customers or pay these high prices for hotels. Being predictive is also important. By predicting the price of the hotels, the agent can make alternative plans to cope with the very high prices.
- *Non-competitive environments* where there is very little competition for hotels and an agent can obtain the rooms it wants at low prices. For example, in game 6341, there is very little competition and the closing prices for T (S) are 7 (12), 92 (27), 70 (53) and 62 (7). In this situation, the best strategy is to buy all flights earlier; since the agents can always get the hotels they want.

- *Semi-competitive environments* where prices are medium. There is competition, but it is not very severe. For example, in game 444, the clearing prices for T (S) are 5 (2), 128 (71), 128 (60) and 116 (3).

4.2.2 The Agent Architecture

Given the uncertainty and unpredictability involved in the TAC, it is desirable for the agents to be responsive to their prevailing situation during the course of bidding. To this end, Figure 4.1 overviews the SouthamptonTAC agent. The time period from when the agent polls the TAC server to get the most up to date asks/bids of all auctions to when it submits its bids to the TAC server is called a *round*. SouthamptonTAC connects to the server in a continuous series of such rounds (the length of a round depends on the location of the TAC server as well as the server load, but it typically varies between 2 and 30 seconds). In each round, the agent first processes this ask or bid information in Bids Preprocessor to get the prices of different goods, number of goods it actually owns and may possibly own (only for hotel rooms) and its current active bids. Then, Hotel Price Predictor is used to predict the likely clearing price of each hotel auction (see Section 4.2.7). All of this information is then input to the Allocator which calculates the optimal distribution of goods to customers given the current situation (see Section 4.2.4 for more details). Given this assignment, the agent then determines its subsequent bidding actions. Flight Categoriser uses updated flight prices to classify each flight auction according to its expected change of price and takes the output of Allocator to determine how many trips to bid for (see Section 4.2.5). For example, it may delay buying the flight tickets if it believes the price change will be small, so that it has flexibility in choosing the hotel rooms. Hotel Bid Adjustor takes the Allocator's output, the agent's current active bids, the hotel auction's ask prices, as well as the predicted prices and decides whether to increase the price of its bids or to "withdraw" (see Section 4.2.7) the current bids and turn to other auctions (see Section 4.2.7). Entertainment Bid Processor determines the type and the amount of entertainment tickets to bid for (see Section 4.2.6). Where SouthamptonTAC differs from its predecessor is in having the Environment Sensor in the architecture. This component (described in more detail in Section 4.2.3) aims to determine what type of environment the agent is presently situated in (as detailed in Section 4.2.1). The reason for doing this is so that the agent can adapt its bidding strategy accordingly (see Section 4.2.8).

Among these components, the most important ones are the Environment Sensor, Hotel Price Predictor, and Allocator. The Environment Sensor aims to determine the degree of competitiveness in the environment both before a game starts and during the course of a game. It does this because the agent will use different bidding strategies in the different situations. The Hotel Price Predictor is important because it lets the agent forward plan about how best to draw up travel plans for the customers. The Allocator is important

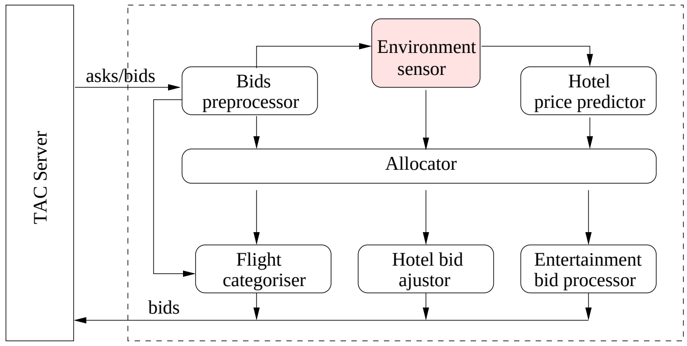


Figure 4.1: Overview of the SouthamptonTAC agent.

because it can allocate the goods the agent has bought to its customers in an optimal way and because it highlights what goods still need to be bought.

SouthamptonTAC divides a game into three stages: the *probing* stage (up to minute 4), the *decisive* stage (from minutes 5 to 11) and the *finalisation* stage (minute 12). Hotel auctions are the most uncertain part of the game. This uncertainty stems both from the random nature of the customers' preferences and from the way opponents deal with their hotel bidding. Nevertheless, a rational agent should have submitted all its hotel bids before the end of the 4th minute (otherwise a hotel auction may close and the agent will miss out on those rooms). Thus, during the first 4 minutes the demand of the hotel market is unpredictable. Given this, SouthamptonTAC uses the probing stage to buy some flights which it has a high possibility of needing, to place buy and sell bids in the entertainment auctions, and to place initial hotel bids. The agent bids for not only what it needs, but also for extra rooms in the hotels with low ask prices (since the additional outlay is comparatively small and gives the agent greater flexibility). As the decisive stage progresses, the demand of the various auctions becomes clearer and rooms are actually allocated which means the agent can more accurately decide which hotels to go for. The finalisation stage represents the agent's last chance to transact on entertainment tickets and to buy any remaining flights that are needed. There is no longer any uncertainty in this stage and so Allocator can find the optimal allocation and the appropriate bids are generated.

4.2.3 TAC Environment Recognition

We treat the environment recognition problem as one of fuzzy pattern recognition since it is impossible to precisely determine the type while the game is running. To this end, we apply the maximum similarity principle [Pedrycz, 1990] in the Environment Sensor component of the agent architecture. This recognition process is used in two cases: before a game starts and during a game. Before a game starts, the agent calculates the average hotel closing prices of the previous 10 games[11] from the price history and uses

[11] We chose ten (based on experience of playing the game) as a suitable indicator that is sufficiently stable to not be influenced by atypical game outcomes, but sufficiently adaptive to respond to genuine changes in the patterns of games.

the maximum average price as a reference price to classify the environment in a given game. We use the maximum price in this fashion since if one price is high it is likely that others will also be high and so the environment is competitive (*mutatis mutandis* when the reference price is low or medium). During a game, the agent continuously monitors the current hotel prices and records the current maximum price to see if the environment type changes from its initial prediction. More formally, let T_i (S_i) represent T (S) on day i, and P_{T_i} (P_{S_i}) represent the current price if the agent is monitoring a running auction or the average history price if it is making its initial assessment of the environment of T_i (S_i). Suppose the prices of $T_1, \cdots, T_4, S_1, \cdots, S_4$, are $P_{T_1}, \cdots, P_{T_4}, P_{S_1}, \cdots, P_{S_4}$. Then, P_{max} (the maximum hotel price) is simply:

$$P_{max} = \max(P_{T_1}, P_{T_2}, P_{T_3}, P_{T_4}, P_{S_1}, P_{S_2}, P_{S_3}, P_{S_4}).$$

Let E_c, E_s, and E_n correspond to the fuzzy sets that represent competitive, semi-competitive and non-competitive environments respectively (see Figure 4.2). Now the type of environment (ε) can be determined by ascertaining which of the fuzzy sets the reference price has the strongest membership to. Thus if:

$$E_x(P_{max}) = \arg\max\{E_c(P_{max}), E_s(P_{max}), E_n(P_{max})\},$$

then ε is of environment type E_x, where $x \in \{c, s, n\}$ and $E_c(x)$, $E_s(x)$, and $E_n(x)$ are the similarity functions for the fuzzy sets E_c, E_s, and E_n. These similarity functions (denoted μ) capture how much the hotel price belongs to each of the different environments and they are defined as follows:

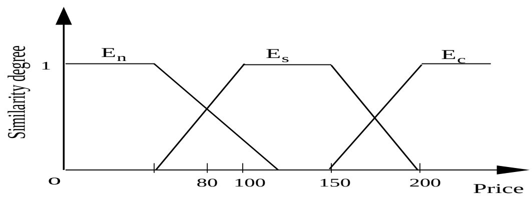


Figure 4.2: Fuzzy sets of the three environment types.

$$\mu_{E_N}(x) = \begin{cases} 1 & x < 50, \\ 0 & x > 120, \\ \frac{120-x}{70} & 50 \leq x \leq 120. \end{cases}$$

$$\mu_{E_S}(x) = \begin{cases} 1 & 100 < x < 150, \\ 0 & x > 200 \text{ or } x < 50, \\ \frac{x-50}{50} & 50 \leq x \leq 100, \\ \frac{200-x}{50} & 150 \leq x \leq 200. \end{cases}$$

$$\mu_{E_C}(x) = \begin{cases} 1 & x > 200, \\ 0 & x < 150, \\ \frac{x-150}{50} & 150 \leq x \leq 200. \end{cases}$$

4.2.4 Allocator

The Allocator component of our agent operates in two different modes: Allocator-1 deals with the allocation of available and unavailable goods and outputs what flights and hotels to buy during the first 11 minutes of a game and Allocator-2 does the same for flights and entertainment tickets in the final minute when the hotel situation is finalised. By means of illustration, a sample output from the Allocator is:

buy inflight: (1, 3, 2, 2)	buy outflight: (0, 2, 3, 3)
buy good hotel: (0, 3, 3, 3)	buy bad hotel: (1, 1, 1, 0)
customer 1: (4, 5, 1, 0, 0, 0)	customer 2: (1, 4, 0, 3, 2, 1)
customer 3: (3, 5, 1, 3, 0, 0)	customer 4: (3, 4, 1, 3, 0, 0)
customer 5: (4, 5, 1, 0, 0, 0)	customer 6: (2, 3, 1, 0, 0, 0)
customer 7: (2, 3, 1, 0, 0, 0)	customer 8: (2, 4, 1, 3, 2, 0)

This indicates the agent needs to buy 1 inflight ticket on day 1, 3 on day 2 and 2 on days 3 and 4; buy 2 outflight tickets on day 3, and 3 on days 4 and 5; that good hotels are needed for day 2 (3 rooms), day 3 (3 rooms) and day 4 (3 rooms); that bad hotels are needed for day 1 (1 room), day 2 (1 room) and day 3 (1 room); and that the plans for the individual customers are currently as follows: customer 1 goes on day 4, returns on day 5 and will stay in the good hotel (1 in third element of tuple), customer 2 will go on day 1, return on day 4, stay in the bad hotel (0 in third element of tuple), go to wrestling on day 3, the amusement park on day 2 and the museum on day 1; and so on for each of the remaining customers.

Both Allocator-1 and Allocator-2 use the linear programming package *LPsolver*[12] to generate their solutions. The solution found is optimal and never took more than 1 second to generate on a 1.33 GHz Pentium during all games played.

Dealing first with Allocator-1, in total this has 92 constraints and 272 variables.[13] In more detail, each customer can choose from inflight day (1 to 4) and outflight day (2 to 5) and hotel type (T or S). This means that in total there are 20 valid packages for each customer (see Appendix A for more details). Given 8 customers, this leaves 160 combinations (160 variables in allocators). For each customer, the agent can choose one from 12 entertainment tickets (3 types for 4 days). Thus there are 96 variables for 8 customers. Moreover, there are 8 variables to denote the number of flight tickets needed for each flight auction and 8 variables to denote the hotel rooms needed for each hotel

[12] *LPsolver* is based on lp_solve, a simplex-based code for linear and integer programming problems by M. Berkelaar. The source is available at ftp://ftp.es.ele.tue.nl/pub/lp_solve/lp_solve.tar.gz.

[13] Our approach is based on that used in ATTac-2000 [Stone et al., 2001] but we improve upon their method. We use only 92 constraints while ATTAC-2000 has 188 constraints. Thus our Allocator greatly improves the speed of finding a solution.

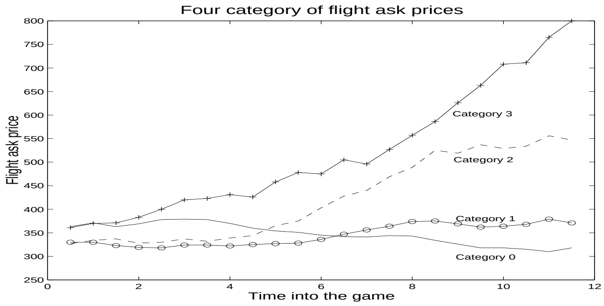


Figure 4.3: Four categories of flight ask prices.

auction. Since from the 4th minute, one hotel auction closes, the number of variables decreases from 272 to 264 at the end of the 11th minute. The 92 constraints come from the fact that: each customer only gets one valid package (8 constraints), flight tickets/hotel rooms/entertainment tickets allocated must be less than the number the agent has or is going to buy (28 constraints), each customer can only use each type of entertainment ticket once (24 constraints), and each customer must use its entertainment tickets between the days he stays in TACtown (32 constraints).

The Allocator-2 deals with the allocation of available and unavailable goods and outputs what flights and entertainments to buy in the last minute (this solver has 276 variables and 92 constraints). The first 264 variables are the same as the above, except that the last 12 variables denote the entertainment ticket numbers for each type each day (12 variables). The constraints are the same as those above.

4.2.5 Flight Auctions

The flight price is perturbed every 24 to 32 seconds by a value drawn uniformly from -10 to $x(t)$. The final upper bound x (called the *flight's determinant factor*) on perturbations is a random variable chosen independently from $[10, 90]$ for each flight for each game. The upper bound on perturbations at time t is a linear interpolation between 10 and the determinant factor x:

$$x(t) = 10 + \frac{t \times (x - 10)}{t_{max}}, \tag{4.1}$$

where t_{max} is the time period of a game (720 seconds) and x is not known to the participants. Figure 4.3 shows four different categories of flights in game tac2-7011 and leads to a number of observations. Firstly, during the first 4 minutes of the game, the distribution of the prices vary in a small range. This is because the current time is early and time is the dominant factor determining the range of prices according to formula (4.1). Secondly, the price changes increase or decrease gradually during the game since time is continuously changing. Thirdly, at the end of each game, some flight prices are lower than their initial price (Category 0), some rise slowly (Category 1), some rise quickly (Category 2) and some rise very rapidly (Category 3). Here the differences are mainly due to the different final bound values of x.

Given these observations, we believe it is important to try and classify the various flight auctions at run-time since this categorisation should lead to different bidding behaviour. To this end, our agent observes the changes in prices and puts each flight auction into one of four categories:

$$\mathcal{F}_j = \{f \mid f's \text{ determinant factor } x \in [L_j, U_j]\} \quad (j = 0, 1, 2, 3), \tag{4.2}$$

where f represents a flight auction: L_j and U_j represent the lower and upper limits of the flight's determinant factor respectively: when $j = 0$, $L_0 = 10$ and $U_0 = 15$; when $j = 1$, $L_1 = 15$ and $U_1 = 30$; when $j = 2$, $L_2 = 30$ and $U_2 = 60$; and when $j = 3$, $L_3 = 60$ and $U_3 = 90$.[14] We found that although the increase or decrease is randomly drawn, if x is small, the price does not rise quickly; conversely, if x is high, the price will rise rapidly. The agent computes its increase or decrease so far and classifies each flight. We believe it is unnecessary to find the precise x, because even though the increase or decrease is randomly drawn, an x that is close to the real 'x' is sufficient to approximate the expected range of change. This categorisation is computed as follows:

$$\arg \min_{0 \le j \le 3} \left| \frac{1}{n} \sum_{i=1}^{n} \delta_i - M_j \right|, \tag{4.3}$$

where n is the number of times the price changed in the auction; δ_i is the ith price change; and M_j, called the centre of $\mathcal{F}_j$, is given by

$$M_j = \frac{1}{U_j - L_j + 1} \left(\sum_{x=L_j}^{U_j} \left(\frac{1}{n} \sum_{k=1}^{n} \frac{(x-10)t_k}{2t_{max}} \right) \right), \tag{4.4}$$

where t_k is the time at which the kth price change is quoted and t_{max} is the time period of a game (720 seconds).

Formula (4.3) computes the average price change of the flight and classifies it into the closest category. However, since the price change δ is drawn from a range, whenever δ is larger than the upper limit of the range, the flight must belong to a category with a bigger x. For each $\mathcal{F}_j$, the upper limit of the range that random changes are drawn from rises with time. Thus, $U_j = g_j^u(t)$ where $g_j^u(t)$ is the upper limit of the determinant factor of $\mathcal{F}_j$ at time t. Then, suppose a flight k is currently categorised as $\mathcal{F}_j$ and the current price change is δ, if $\delta > g_j^u(t)$ and $\delta \le g_{j+1}^u(t)$, then flight k should be reclassified as $\mathcal{F}_{j+1}$.

The flight categorisation is updated in each round. Clearly, as the game progresses the categorisation becomes more accurate (see Section 4.3.3). However, for most flights the prices rise during the game. However, if the agent buys flight tickets very early, it may

[14] These values were first picked based on our experience with the games. Then, they were tested by a large number of games and shown to produce reasonable classifications.

fail to buy the necessary hotel rooms (leading to some invalid travel packages). Thus, what we need is a good trade-off (recall the importance of trade-offs noted in Section 1.2) between buying flights earlier at lower prices and buying them later to ensure they fit with the hotels that have been bought. To achieve this, our agent first needs to decide what type of TAC environment it is situated in (see Section 4.2.3). For non-competitive games, the agent will buy all the flight tickets once when the game starts. For competitive or semi-competitive environments, the agent buys a number of flight tickets at the beginning of the game (8-10 for competitive and 12-14 for semi-competitive games) because it knows it will need some flights. Then, when and how many of the remaining ones to buy of a particular flight is based on the flight categorisation. This delay in buying flight tickets ensures there is a degree of flexibility. Then, whenever an $\mathcal{F}_3$ auction is sensed, the agent will buy flights immediately. However, for an $\mathcal{F}_0$ auction, it will buy the tickets in the last minute since the price at the end of the game will be similar or less than the initial price. An $\mathcal{F}_1$ flight will be bought when the corresponding hotel rooms are guaranteed or the demand of the hotels involved with the flight is not very high. An $\mathcal{F}_2$ flight is bought immediately after the probing stage. This is because during the probing stage the expected price change is quite small (recall $g_j^u(t)$ rises with time). However after the probing stage, the increase is more significant (see discussion in Section 4.2.8 for the rationale).

4.2.6 Entertainment Auctions

The entertainment CDAs involve two kinds of bids: buys and sells. That is, an agent can place both bids to buy and asks to sell. It determines the amount of bids or asks to place as well as the price of bids or asks. The Entertainment Bid Processor handles entertainment bidding. A buy (sell) bid will immediately match the lowest price standing sell (highest price standing buy) bid that has a price at or below (above) the price of the buy (sell) bid. Bids match at the price of the standing bid in the auction.

- *Number of buy bids*. Place buy bids for a particular customer so that the allocation of the entertainment tickets for that customer is maximally satisfactory. For example, if a customer will stay for one day, the agent will buy the tickets with the highest preference value for that day; if the customer stays for two days, buy tickets with the two highest preference values for each day, and so on. Here the agent places extra bids to increase the chance of obtaining a ticket. Whenever an agent is successful in buying a particular ticket for a given day, it withdraws its buy bids for that entertainment on the other days.
- *Buy bid reservation price*. Let $v_{i,j}$ be the preference valuation of customer i for entertainment j. The agent only buys a good if it can make a profit from it, *i.e.*, $v_{i,j}$ must be larger than the price of buying that good. Thus, the buy bid reservation price *bid* is given by: $bid = v_{i,j} - \psi(t)$ ($\psi(t) > 0$), where $\psi(t)$ is the profit the agent

can obtain if the good is transacted at *bid*. Here $\psi(t)$ is a decreasing function with time meaning that the later it is, the lower the profit the agent is willing to accept.

- *Number of sell bids*. Sell any unallocated entertainment tickets[15] and any allocated tickets that the agent can get more profit for by selling rather than by allocating to a customer.
- *Sell bid reservation price*. The reservation price *ask* is given by $ask = cost + \phi(t)$ ($\phi(t) > 0$), where $\phi(t)$ is a decreasing function of time and *cost* is the preference value for an allocated ticket and a predefined value for unallocated ones. The latter value varies according to the agent's context. If it has *n* unallocated tickets, the cost will be at descending prices (from 80 to 50) meaning that the more goods the agent has, the quicker it wants to sell them (thus, the sell price is lower). $\phi(t)$ decreases with time meaning the later it is, the lower the price the agent is willing to sell the good for (since selling for a small profit is better than not selling at all).

SouthamptonTAC does not wait until the ask price decreases to or the bid price rises to exactly its reservation buy or sell price. Rather, the agent continuously observes the market and when it finds an ask or bid price very close to its reservation price it will decrease or increase its ask or bid to match it. For example, if a bid is 79.5 and the ask price of our agent is 80, it evaluates the bid and decides to decrease the ask to 79.5 so as to accept the bid. This strategy is achieved using fuzzy sets (the idea comes from the FL-strategy in Chapter 3). The strategy is simple but effective since it avoids missing transactions where the bid and ask are quite close. Figure 4.4 shows the fuzzy sets used to decide when to accept the current asks or bids, where θ_b and θ_s are the thresholds of the degree that an agent would like to relax its constraints on buy or sell bids. Suppose two fuzzy sets B and S are characterised by the membership function $\mu_B : X \longrightarrow [0,1]$ and $\mu_S : Y \longrightarrow [0,1]$. $\mu_B(x)$ and $\mu_S(y)$ are interpreted as the degree of membership of x, (*i.e.*, current asks placed by sellers) in fuzzy set B for each $x \in X$ and y, (*i.e.*, current bids placed by buyers) in fuzzy set S for each $y \in Y$ respectively. Formulae (4.5) and (4.6) are the similarity membership functions for bids and asks.

$$\mu_B(x) = \begin{cases} 1 & \text{if } x < b_r, \\ 0 & \text{if } x > b_0, \\ \frac{b_0 - x}{b_0 - b_r} & \text{if } b_r \leq x \leq b_0; \end{cases} \tag{4.5}$$

$$\mu_S(y) = \begin{cases} 1 & \text{if } y > a_r, \\ 0 & \text{if } y < a_0, \\ \frac{y - a_0}{a_r - a_0} & \text{if } a_0 \leq y \leq a_r. \end{cases} \tag{4.6}$$

[15] Unallocated tickets are caused by having multiple tickets for the same event or the same day for a given customer or by having no customer staying on the night of the entertainment.

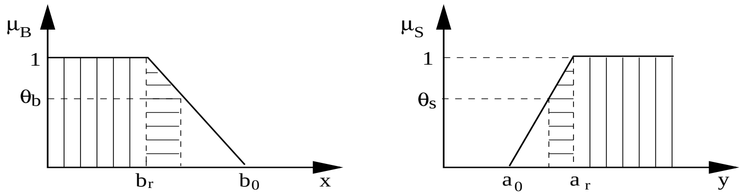


Figure 4.4: Fuzzy sets used in entertainment CDAs.
Here b_0/a_0 is the current highest bid/lowest ask; b_r/a_r is the agent's reservation price and θ_b/θ_s is the agent's threshold of accepting the ask/bid. The region with vertical lines represents the original price acceptance range. Using fuzzy sets, the acceptance range also includes the region with horizontal lines.

4.2.7 Hotel Auctions

Hotel auctions are the most important, uncertain and difficult part in TAC. Since customers can have no entertainment and flights tickets are available throughout the game, only hotel auctions are uncertain. Moreover, failure in a single hotel auction can cause the failure of an entire travel package. To deal with this complexity, several strategies are used: (i) fuzzy reasoning to predict the likely clearing prices (see Section 4.3.3 for an evaluation of the prediction method);[16] (ii) "withdraw" non-profitable hotel bids; and (iii) reasoning to determine when to switch between bidding for the different hotels. Each of them are dealt with in turn.

Fuzzy reasoning about hotel closing prices

According to the basic laws of supply and demand theory [Perloff, 1998], the higher demand there is in a market, the higher the price of the goods. Thus the competition among the agents on a particular hotel auction leads to a rise in the price of the hotels. For example, T2 and T3 are in greatest demand, since staying in the good hotel gets the higher utility and day 2 and day 3 are part of most customers' stays. Therefore their prices are always the highest. This information, as well as the price changes, are factored into the agent's reasoning about price prediction. The reasoning utilises fuzzy rules to predict the clearing prices of hotels. The motivation for using fuzzy rules is as per Section 1.2. Through observation, we find that the factors that effect the price of hotels are: the price of the hotel (P), the price of the counterpart hotel (CP), the price change in the previous minute (C) and the previous price change of the counterpart hotel (when it closed) (CC). Figure 4.5 shows the relation between the closing of several hotels in game tac2-5960. Here S3 closes first (at the fourth minute). Then the price for T3 rises very quickly because S3's closure means some agents fail to get S3 and so they have to bid in T3. Usually, the price of S2 is high, but since S2 closes early, its price is relatively low. Also, the rooms of day 2 have a close relationship with those of day 3 because many customers stay for successive days. Thus, the price of T2 also rises quickly.

[16] Note that the same type of controller for the reasoning is used as was used in in Section 3.1.2.

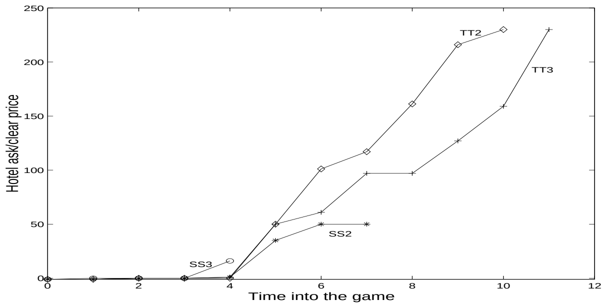


Figure 4.5: Hotel prices changes in game tac2-5960.

To capture reasoning of this kind, we use the Sugeno controller[17] [Sugeno, 1985, Zimmermann, 1996], since it is easy to use and has already been shown to be effective (Chapter 3). In more detail, the fuzzy reasoning inference mechanism employed here adheres to the following fuzzy reasoning pattern:

$$
\begin{array}{ll}
\mathcal{R}_1: & \text{if } x_1 \text{ is } A_{11} \text{ and } \cdots \text{ and } x_m \text{ is } A_{1m} \text{ then } \Delta_1 = c_1 \\
\textbf{also} & \\
\mathcal{R}_2: & \text{if } x_2 \text{ is } A_{21} \text{ and } \cdots \text{ and } x_m \text{ is } A_{2m} \text{ then } \Delta_2 = c_2 \\
\textbf{also} & \\
\vdots & \\
\textbf{also} & \\
\mathcal{R}_n: & \text{if } x_1 \text{ is } A_{n1} \text{ and } \cdots \text{ and } x_m \text{ is } A_{nm} \text{ then } \Delta_n = c_n \\
\textbf{fact:} & x_1 \text{ is } x_1^{'} \text{ and } \cdots \text{ and } x_m \text{ is } x_m^{'} \\
\hline
\textbf{consequence:} & \Delta^{'}
\end{array}
$$

where $A_{11}, \cdots, A_{nm}$ are fuzzy sets, and $\Delta_1, \cdots, \Delta_n$ are variables indicating the predicted price increase. The output of the individual rule is denoted as Δ_i and $c_i \in \{$*small*, *medium*, *big*, *very-big*$\}$ ($i \in \{1, \cdots, n\}$) are fuzzy parameters predefined by the designer.

The firing level α_i of the rules $\mathcal{R}_i$ is computed by the *Min* operator. That is,

$$\alpha_i = \min\{A_{i1}(x_1^{'}), \cdots, A_{im}(x_m^{'})\}, \tag{4.7}$$

where $A_{i1}(x_1^{'}), \cdots, A_{im}(x_m^{'})$ are the membership functions of the corresponding fuzzy sets A_{i1} and A_{im}, respectively. According to the Sugeno controller's definition, the crisp control action (*i.e.*, the predicted increase of hotel closing price) of the rule base is obtained by:

$$\Delta^{'} = \frac{\sum_{i=1}^{n} \alpha_i \Delta_i}{\sum_{i=1}^{n} \alpha_i}. \tag{4.8}$$

[17] The use of other fuzzy logic controllers, such as the conventional Mamdani controller [Zimmermann, 1996] is also possible. However we chose Sugeno because it is shown to be effective in the CDA (Chapter 3 and empirical results show that it is also effective in the TAC environment.

When predicting the closing price of a hotel auction, three cases are considered:[18] (i) when the good and bad hotels are open; (ii) when the counterpart hotel auction had just closed (within the previous minute) and (iii) when it had been closed for a longer period of time. The corresponding rule bases are shown in Tables 4.5 to 4.7.

In these tables, the hotel ask prices (*P* and *CP*) are expressed in the fuzzy linguistic terms: *very-high*, *high*, *medium*, and *low* (Figure 4.6 and Table 4.8) and the price changes (*C* and *CC*) in the fuzzy linguistic terms *quick*, *medium*, and *slow* (Figure 4.7 and Table 4.9). The output of the rule base is the prediction Δ' of how much the price of the given hotel is likely to increase, thus $\Delta' \in \{small, medium, big, very\text{-}big\}$ is the increase that is added to the current price to obtain the predicted clearing price.

Table 4.5: Fuzzy rule base when counterpart auction is open.

IF *P* is *high* and *C* is *quick* THEN Δ is *big*.
IF *P* is *high* and *CP* is *high* and *C* is *not-quick* THEN Δ is *big*.
IF *P* is *high* and *CP* is *not-high* and *C* is *not-quick* THEN Δ is *medium*.
IF *P* is *low* and *CP* is *high* THEN Δ is *medium*.
IF *P* is *low* and *CP* is *not-high* THEN Δ is *small*.
IF *P* is *medium* and *CP* is *high* THEN Δ is *medium*.
IF *P* is *medium* and *CP* is *not-high* and *C* is *not-slow* THEN Δ is *medium*.
IF *P* is *medium* and *CP* is *not-high* and *C* is *slow* THEN Δ is *small*.

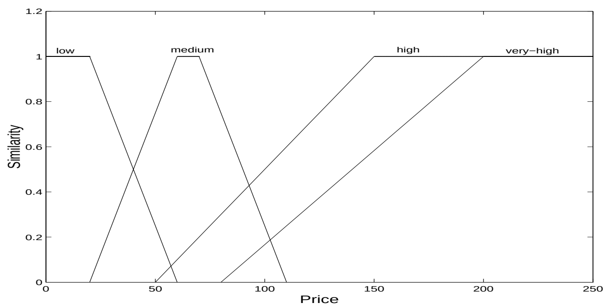


Figure 4.6: Fuzzy sets for price state of Tampa Towers/Shoreline Shanties.

[18]SouthamptonTAC-01 used two rule bases to make its predictions: (i) for when both the good and bad hotels are open and (ii) for when the counterpart auction was closed. However, we found that our predictions in the latter case could be improved if we separated out the cases in which the counterpart auction had just closed (within the last minute) and when it had been closed for a longer period of time. This difference occurs because hotel prices change more rapidly and with a different pattern when the counterpart has just closed. When the counterpart auction has been closed for a longer period, the changes are smaller.

Table 4.6: Fuzzy rule base when counterpart auction just closed.

IF P is *high* and C is *not-slow* and CC is *quick* THEN Δ is *very-big*.
IF P is *high* and C is *not-slow* and CC is *not-quick* THEN Δ is *big*.
IF P is *high* and C is *slow* and CC is *quick* THEN Δ is *big*.
IF P is *high* and C is *slow* and CC is *not-quick* THEN Δ is *medium*.
IF P is *medium* and C is *quick* THEN Δ is *big*.
IF P is *medium* and C is *medium* and CC is *quick* THEN Δ is *big*.
IF P is *medium* and C is *medium* and CC is *not-quick* THEN Δ is *medium*.
IF P is *medium* and C is *slow* and CC is *quick* THEN Δ is *medium*.
IF P is *medium* and C is *slow* and CC is *not-quick* THEN Δ is *small*.
IF P is *low* and C is *slow* and CC is *quick* THEN Δ is *medium*.
IF P is *low* and C is *not-slow* THEN Δ is *medium*.
IF P is *low* and C is *slow* and CC is *not-quick* THEN Δ is *small*.

Table 4.7: Fuzzy rule base when counterpart auction has closed for more than one minute.

IF P is *high* and C is *not-slow* THEN Δ is *big*.
IF P is *high* and C is *slow* THEN Δ is *medium*.
IF P is *medium* and C is *quick* THEN Δ is *big*.
IF P is *medium* and C is *medium* THEN Δ is *medium*.
IF P is *not-high* and C is *slow* THEN Δ is *small*.
IF P is *low* and C is *not-slow* THEN Δ is *medium*.

Table 4.8: Similarity functions of fuzzy sets for price of T/S.

$\mu_{low}(x) = \begin{cases} 1 & \text{if } x < 20 \\ 0 & \text{if } x > 60 \\ \frac{60-x}{40} & \text{if } 20 \leq x \leq 60 \end{cases}$	$\mu_{medium}(x) = \begin{cases} 1 & \text{if } 60 \leq x \leq 70 \\ 0 & \text{if } x < 20 \text{ or } x > 110 \\ \frac{x-20}{40} & \text{if } 20 \leq x \leq 60 \\ \frac{110-x}{40} & \text{if } 70 \leq x \leq 110 \end{cases}$
$\mu_{high}(x) = \begin{cases} 0 & \text{if } x < 50 \\ 1 & \text{if } x > 150 \\ \frac{x-50}{100} & \text{if } 50 \leq x \leq 150 \end{cases}$	

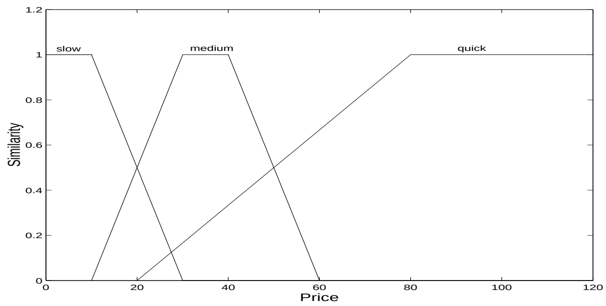


Figure 4.7: Fuzzy sets for price change.

Table 4.9: Similarity functions for fuzzy sets of price change.

$\mu_{slow}(x) = \begin{cases} 1 & \text{if } x < 10 \\ 0 & \text{if } x > 30 \\ \frac{30-x}{20} & \text{if } 10 \leq x \leq 30 \end{cases}$	$\mu_{medium}(x) = \begin{cases} 1 & \text{if } 30 \leq x \leq 40 \\ 0 & \text{if } x > 60 \text{ or } x < 10 \\ \frac{x-10}{20} & \text{if } 10 \leq x \leq 30 \\ \frac{60-x}{20} & \text{if } 40 \leq x \leq 60 \end{cases}$
$\mu_{quick}(x) = \begin{cases} 0 & \text{if } x < 20 \\ 1 & \text{if } x > 80 \\ \frac{x-20}{60} & \text{if } 20 \leq x \leq 80 \end{cases}$	

Hotel bid withdrawal

Turning now to the notion of withdrawing bids. TAC does not allow hotel bid withdrawal during a game (as described in Section 4.1). Nevertheless our agent can effectively achieve withdrawal by the following means. The agent decides which bids to continue with, which bids to withdraw and where new bids are needed based on the output of the Allocator and Hotel Bid Adjustor. The agent decides to withdraw a bid either because the hotel rooms cannot be used or because the hotel price is predicted to rise sharply. Suppose the current ask price of a hotel auction is a and our agent has already placed a bid higher than a. The agent calculates the predicted clearing price of each hotel auction (from Equation 4.8). The idea is that the agent submits a bid (for the appropriate quantity) at the price of $a+1$. In so doing, the agent believes that new bids from other agents will top its withdraw bid and thus will remove its commitment to those rooms. This does not violate the rules of the TAC auction, but avoids getting high price hotels. This method proved very effective and meant our agent could withdraw bids before the ask price rose too high. This ability to withdraw bids make it possible to consider switching the bidding between the different hotel types (to be discussed below) and so increased the flexibility of our bidding strategy.

Switching between hotels

SouthamptonTAC's initial allocation of hotel rooms starts early in the game, while the ask prices of all the hotels are very low. At this time, the agent uses the reference hotel prices as input to the Allocator. This reference price comes from the average prices of the various hotels in the previous 10 games. While during the game, it is sometimes useful for an agent to change from trying to buy one sort of hotel to going for another, there are risks associated with this: (i) it may fail to get rid of its existing bids for its original hotels (thus it may double bid); and (ii) it is possible that the price of the hotel that is changed to rises very quickly while the price of the old type remains unchanged (thus the agent may want to switch its bidding back to its original hotel).

To manage the process of determining when to change the type of hotel to bid for, our agent employs the following process. The output of the Allocator is an optimised solution which means that if a new allocation (involving a different hotel) produces one more unit of profit than the current one, it will be suggested. However, blindly following this recommendation may cause the agent to oscillate in its behaviour and lead it to having unwanted hotel rooms at the end of the process. To avoid this, our agent makes sure that any change in behaviour is likely to have a worthwhile effect on its score. In more detail, given a change threshold θ, suppose a customer is currently allocated to stay in Hotel A giving it a utility of U_A. Now assume the Allocator suggests placing this customer in Hotel B giving a utility of U_B (where $U_B > U_A$). The rule for enacting this decision is:

$$\textbf{IF } U_B - U_A - q_A \times p_A > \theta \textbf{ THEN} \text{ change to } B \textbf{ ELSE} \text{ stay in } A,$$

where $q_A \times p_A$ indicates the loss if the agent cannot withdraw its existing bids in hotel auction A. Here q_A is the quote price of auction A and p_A is the possibility of not withdrawing the bid (this can be approximated from the speed with which bids have changed in the last minute). Adhering to this rule means we only withdraw those bids that have a continuously changing bid history in the past minute and those where much more profit can be obtained by changing the hotel type.

4.2.8 Varying the Bidding Strategy

After our experiences in TAC-01, we came to believe that there is no single best strategy that can deal with all the different types of TAC environment (see Section 4.3 for more details). For example, a risk-seeking agent that always allocates the optimal travel package for its customers and buys flights earlier is highly effective in non-competitive environments. This is because there is little competition in hotel bidding and the agent can always obtain what it wants. On the other hand, delaying buying flights and shortening the stay of customers works well in competitive games. For this reason, SouthamptonTAC dynamically varies its bidding strategy according to its assessment of the environment type (see Section 4.3.4 for an evaluation of the effectiveness of being able to do this). In games it deems non-competitive, SouthamptonTAC buys all of its flight tickets at the beginning of the game (the number of flight tickets to buy is a parameter which can be adjusted according to the agent's attitude to risk, see as per Section 1.2) at the beginning of the game and never changes the travel plan of its clients (unless it senses a change in the environment). In this way, it avoids buying extra hotels which cost extra money. Also, the agent can receive optimal utility by not shortening the stay of its customers. In competitive games, our agent buys flights according to its assessment of the flight category (as discussed in Section 4.2.5). In these games the agent may alter its customers' travel plans in order to avoid staying in expensive hotels for long periods. In semi-competitive games, the agent behaves inbetween these two strategies; it buys most of the flights earlier and will only change travel plans if a significant improvement can be obtained.

4.3 Evaluation

Our evaluation of SouthamptonTAC is composed of two components: (i) the results from the TAC-01 and TAC-02 competitions and (ii) our post-hoc systematic analysis in a range of controlled environments.

4.3.1 TAC Results

TAC consisted of a preliminary round (mainly used for practice and fine tuning), a seeding round, the semi-finals and the final round. The seeding round determined groupings for the semi-finals. The top 16 agents were organised into two "heats" for the semi-finals based on their position in the seeding round and the first four teams in both heats entered into the final round.

In TAC-01, SouthamptonTAC-01 obtained the highest score in the seeding round (see Table 4.10). Table 4.11 shows the result of the final round, here SouthamptonTAC-01 had the 3rd highest score.[19] Overall, during the course of the competition some 600 games were played and SouthamptonTAC-01 had the highest mean score and lowest standard deviation.

For TAC-02, Table 4.12 shows the seeding round result of each agent's relative score to SouthamptonTAC. Note there is less than 2 points difference between ATTac and SouthamptonTAC and given the random features of the game their performance should be considered as broadly similar. Table 4.13 shows the scores (again relative to our agent) of all the agents in this final round. Again the difference between SouthamptonTAC and the top agent is small, less than 0.8% (when all the games in the competition are considered, SouthamptonTAC is 2.2% better than whitebear).

When both competitions are considered, there are 12 agents that participated in both TACs (some of the agents are designed by the same group but with different agent

[19] This score was calculated without game 7315, where there was a crash due to the network platform failure for SouthamptonTAC. Details can be found in http://auction2.eecs.umich.edu/tac01-scores-finals/.

Table 4.10: Result of TAC-01 seeding round.

Rank	Agent	Avg(-10 worst)	Std Dev	Games played
1	SouthamptonTAC	3163.8	855.3	315
2	whitebear	3163.8-43	881.4	318
3	Urlaub01	3163.8-88.3	1197.8	319
4	livingagents	3163.8-151.6	1251.7	305
5	TacsMan	3163.8-180	1065.7	315
6	CaiserSose	3163.8-294	1219.7	315
7	polimi_bot	3163.8-306.2	980.8	316
8	umbctac	3163.8-399	1288.4	313

Table 4.11: Result of TAC-01 final round.

Rank	Agent	Avg	Std Dev	Games played
1	livingagents	3530.6+139.4	622.3	24
2	ATTac	3530.6+91	691.6	24
3	SouthamptonTAC	3530.6	568.8	23
4	whitebear	3530.6-17.4	700.1	24
5	Urlaub01	3530.6-109.4	698.3	24
6	Retsina	3530.6-178.8	668.2	24
7	CaiserSose	3530.6-456.5	656.2	24
8	TacsMan	3530.6-671.3	1054.3	24

Table 4.12: Result of TAC-02 seeding round (440 games).

Rank	Agent	Avg(-10 worst)	Avg
1	ATTac	3129.5+1.8	3033.5+4.2
2	SouthamptonTAC	3129.5	3033.5
3	UMBCTAC	3129.5-11.1	3033.5-16.6
4	livingagents	3129.5-38.1	3033.5-24.9
5	cuhk	3129.5-74	3033.5-62.1
6	Thalis	3129.5-129.8	3033.5-131.9
7	whitebear	3129.5-163.9	3033.5-158.2
8	RoxyBot	3129.5-274.2	3033.5-300.8

names). Four of these twelve qualified for the final rounds in both competitions. For these four agents, their average scores over both competitions (some 1200 games) are: SouthamptonTAC (3229), whitebear (3119), livingagents (3016) and Thalis/CaiserSose (2863). Thus, our agent is the most successful of these.

Through both competitions, we believe that this large number of games and the very nature of the competition mean that the difference in the trader's scores reflect true differences in the performance of the agents' strategies. Thus we believe SouthamptonTAC performs successfully in a wide range of TAC situations.

4.3.2 Controlled Experiments

To evaluate the performance of our agent in a more systematic fashion than is possible in the competition, we decided to run a series of controlled experiments. To do this we

Table 4.13: Result of TAC-02 final round (32 games).

Rank	Agent	Avg(-worst)	Avg
1	whitebear	3492+64.4	3385.5+27.3
2	SouthamptonTAC	3492	3385.5
3	Thalis	3492-140.8	3385.5-139.2
4	UMBCTAC	3492-171.4	3385.5-149.9
5	Walverine	3492-176.4	3385.5-175.9
6	livingagents	3492-182.2	3385.5-204.6
7	kavayaH	3492-242.2	3385.5-286.0
8	cuhk	3492-244.2	3385.5-316.7

devised two competitor agents that adopt strategies consistent with the broad classes of behaviour that were observed in the competition:

- **Risk-seeking agent (RS-agent)** This is based on the behaviour of the livingagents, UMBCTAC, and Walverine agents (see Section 4.4 for more details). This agent buys all the flights tickets at the beginning of the game, bids aggressively in hotel auctions and never changes the plans for its customers.
- **Risk-averse agent (RA-agent)** This is based on the behaviour of SouthamptonTAC-01, Retsina, and sics agents (see Section 4.4 for more details). This agent buys a small number of flight tickets at the beginning of the game to leave some flexibility and it will change the customers' travel plans according to how the game unfolds.

For both these types of agents, as well as for SouthamptonTAC, a record is kept of the closing price history and the initial travel plans for the customers are calculated based on the average price of this history.

The set-up of the experiment is shown in Table 4.14 where it can be seen that there are 36 different cases which cover all possible combinations of SouthamptonTAC, RA-agents and RS-agents given that there can only be eight agents in one game. For example, in the case where the number of RA-agents is 2 and the number of RS-agents is 1, there will be 5 SouthamptonTAC agents. For each case, between 50–100 games[20] were played to test the performance of each kind of agent. In this way, it is possible to produce a wide range of environments, from competitive to non-competitive, and to evaluate the corresponding performance of the different types of agents and the broad behaviour trends. The following conjectures were used as an approximate guide for designing the experiments.

Conjecture 1: The more RS-agents there are in the game, the more competitive (see Section 4.2.1) it will be and the more RA-agents there are, the less competitive it will be.

Conjecture 2: RS-agents will do well in non-competitive environments and RA-agents will do well in competitive ones.

[20]This number differs from game to game. The experiment for a single case stops when the relative scores of the agents become stable.

Table 4.14: Experiment set-up for controlled experiments.
The light grey area indicates competitive environments and dark grey non-competitive ones.

Number of RS−agents	Number of RA−agents: 0	1	2	3	4	5	6	7
0	8	7	6	5	4	3	2	1
1	7	6	5	4	3	2	1	
2	6	5	4	3	2	1		
3	5	4	3	2	1			
4	4	3	2	1				
5	3	2	1					
6	2	1						
7	1							

Number of SouthamptonTAC agents

In terms of Conjecture 1, the average hotel clearing price for those environments marked as competitive (in Table 4.14) is 240 and for those marked as non-competitive it is 67 and for the remainder it is 125. Thus Conjecture 1 can be seen to hold.

We now start to analyse the performance in the different environments. Figure 4.8 shows the performance surface of SouthamptonTAC in the different cases. The x-axis and y-axis represent the number of RS-agents (N_{RS}) and RA-agents (N_{RA}), thus, the number of SouthamptonTAC agents is $8 - N_{RS} - N_{RA}$. The z-axis shows the average score[21] of SouthamptonTAC. The higher the score, the better the agent performs. Figures 4.9 and 4.10 show the performance of the RS-agents and RA-agents on similar graphs.

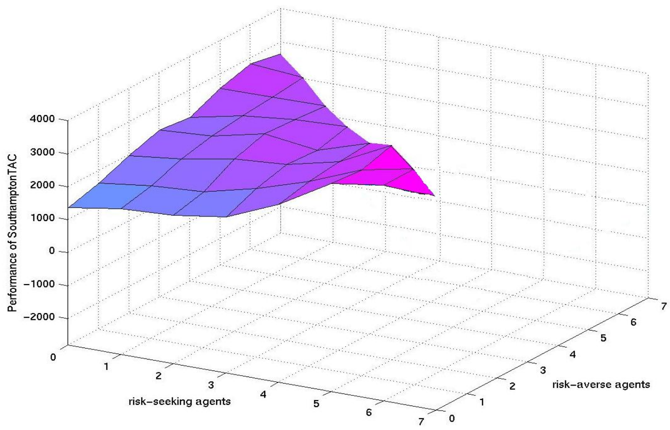


Figure 4.8: Performance of SouthamptonTAC in different environments.

As shown in Figure 4.8, SouthamptonTAC does best, obtains the highest score, in competitive games (*i.e.*, where the number of RS-agents is big). This is due to the adaptive nature of its strategy. When it finds the game competitive, it alters its strategy in the direction of being risk-averse. In non-competitive environments, SouthamptonTAC also does well since it adapts its strategy to bid aggressively because it can always obtain the goods it wants. Both of these observations are consistent with Conjecture 2. The worst

[21] Suppose there are m SouthamptonTAC agents, and the average scores of these agents are s_1, s_2, $\cdots$, s_m. Then the average score shown on the z-axis is $\frac{\sum_{i=1}^{m} s_i}{m}$.

situation for SouthamptonTAC is when all the players are like itself. This is because the competitive tendency of the agents causes the hotel prices to rise to moderate levels and then many of the agents change their customers' travel plans at approximately the same time. This switching behaviour causes the counterpart hotel prices to rise (because of increased competition) and the agents to have unused flights or hotel rooms bought on account of their previous travel plans. For RS-agents, as shown in Figure 4.9, the results also support Conjecture 2. RS-agents behave very well in non-competitive games and their performance decreases rapidly as the number of RS-agents increases. This happens because as more agents bid aggressively, the hotel closing prices get higher. RA-agents behave best in competitive environments when there are many RS-agents, perform adequately in non-competitive games and worst in semi-competitive games when there are a few RS-agents and SouthamptonTAC agents (see Figure 4.10). In the latter two cases, RA-agents change their customers' travel packages reasonably often and this causes them to buy extra hotels and flights that they cannot subsequently use.

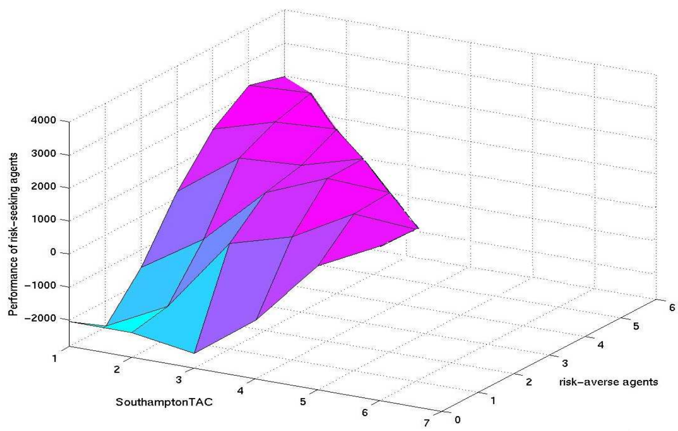


Figure 4.9: Performance of risk-seeking agents in different environments.

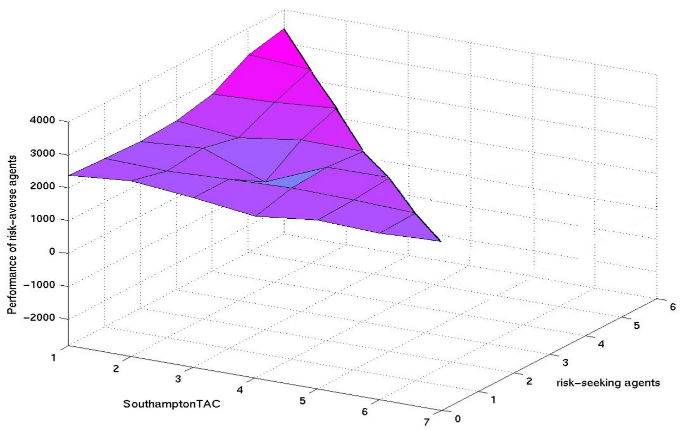


Figure 4.10: Performance of risk-averse agents in different environments.

Moreover, from Figures 4.8 to 4.10, we find that the range of scores for each kind of agent are different; for SouthamptonTAC it is $[1372, 3737]$, for RS-agents it is $[-2742, 2374]$

and for RA-agents it is $[1709, 3445]$. Thus the RA-agent has the narrowest score range and is the most stable agent. The RS-agent has the widest score range since its performance depends heavily on the environment it is situated in. SouthamptonTAC is in between, less stable than RA-agents (but able to obtain higher scores) but with a better worst performance than RS-agents.

While Figures 4.8 to 4.10 show the performance of a single type of agent in various environments, Figure 4.11 compares their scores. There are 8 subfigures and each of them represents several cases of the above experiments.[22] We found that when the number of SouthamptonTAC agents is small (less than 4), they can always outperform both RS-agents and RA-agents (as shown in (e) to (h) and some cases in (a) to (d)). This is because SouthamptonTAC can successfully adapt itself in competitive games and become aggressive in non-competitive ones. However, as we discussed previously, when the number of SouthamptonTAC agents is above 4, the agents exhibit similar behaviour and make the market less efficient. Generally, from (a) to (h), it can also be seen that profits for all agent types increase as the number of RA-agents increases (because these agents keep the hotel prices low).

4.3.3 Predicting Hotel Prices

Most of the agents in TAC engage in some form of hotel price prediction (see Section 4.4). Since, generally speaking, the more accurately the agent can predict these prices the more easily it can identify profitable actions. To this end, Table 4.15 shows the accuracy of SouthamptonTAC's predictions on a minute by minute basis for a single game (randomly chosen) in the final. The figures in the table are the difference between the predicted price and the actual price. Thus, a positive number means over prediction and a negative one means under prediction. As we can see, the trend is that the further into the game the predictions are made the more accurate they are. This is because at the beginning the agent can only work based on the price history of previous games. However as the game progresses, more information is revealed (such as the closing order of the hotels, the current hotel prices and the relation between the hotels). This, in turn, means more accurate predictions can be made. This is important for our agent since it enables its flexible decision making to be based on more or less accurate information.

In most cases, our agent tends to over predict the hotel closing prices. If the hotel prices are not very high, the agent will not suffer since it will not change the plan for its customers; whereas if the prices are very high, the agents may change the travel plans for its customers and therefore obtain a lower score (since it may have bought flights or rooms that it cannot now use). However, when hotel prices rise very quickly, our agent tends to under predict which can cause it to buy highly priced hotels (so reducing its profit).

[22] For example, in figure (c), there are 2 RS-agents, thus the horizontal axis represents the number of RA-agents and the vertical axis is the average score of the different agent types.

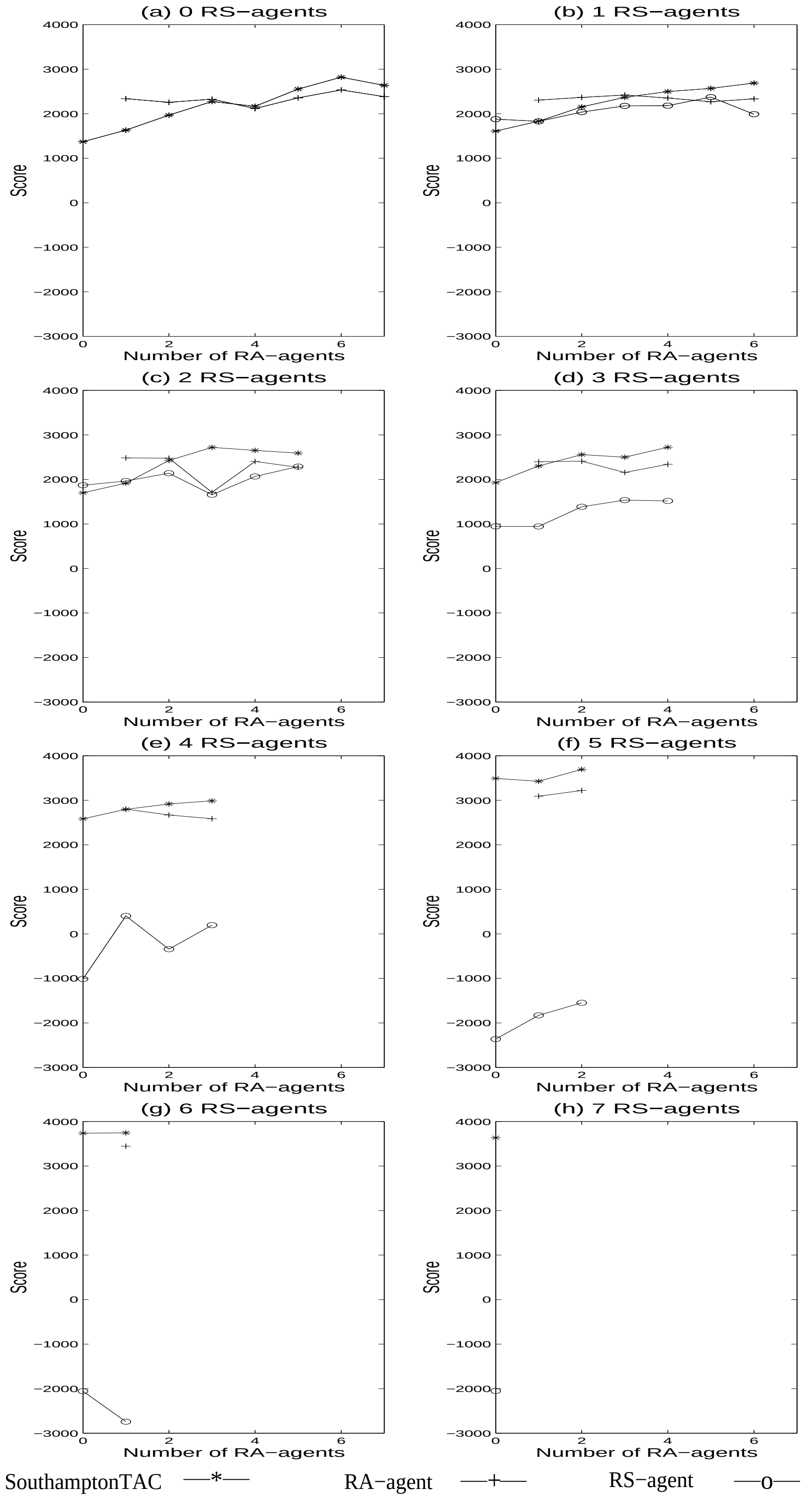


Figure 4.11: Relative performance of the agents in different environments.

Furthermore, Table 4.16 shows the difference between the predicted and actual hotel closing prices for the order in which they closed in the final. For example, for the hotel

Table 4.15: Actual vs. predicted hotel prices.
Positive figure means over prediction and negative figure means under prediction.

Hotel	4	5	6	7	8	9	10	11
T3	86							
T2	40	23						
S3	76	-62	50					
S4	39	9	9	9				
S2	83	11	-32	-53	-17			
S1	45	9	8	9	9	9		
T1	79	9	8	8	8	8	8	
T4	87	10	10	10	10	10	10	-17

Table 4.16: Hotel closing price prediction in final round.

Closing order	Avg difference	Max difference	Min difference
1	64	174	8
2	30	103	2
3	29	144	8
4	38	149	1
5	32	115	8
6	30	111	3
7	27	87	8
8	20	62	9

that closed first (whatever that happened to be in a particular game), the average difference is 64, the maximum difference is 174[23] and the minimum is 8. These results are consistent with those of Table 4.15 and show that the later a hotel closes, the more accurate our agent's prediction is.

4.3.4 Strategy Adaptation

To test the value of the agent being able to adapt its strategy during the course of a game, we compare the performance of our agent with a non-adaptive variant (called na-SouthamptonTAC) that is identical apart from the fact that it cannot change its strategy once a game has started running. In each game, there was one SouthamptonTAC, one na-SouthamptonTAC and the remaining agents were drawn randomly from a pool of RS-agents and RA-agents. We ran this configuration for 164 games and computed the average score of each agent type. Our results were that the adaptive agent received an average score of 3138, the non-adaptive one an average of 2937 and the other agents an average of 1657 (RS-agents) and 2649 (RA-agents). This shows that being adaptive does indeed improve the agents' performance.[24]

[23]This number is large and it occurred at the beginning of the final where the price history data was based upon the seeding round (which had very different outcomes from the final round).

[24]A t-test showed that the adaptive agent's performance is significantly better than the non-adaptive one, where $p < 0.05$.

4.4 Related Work

There are a number of strands of work that are related to what we have described in this chapter. Firstly, there is the work on agents bidding in multiple simultaneous auctions. Secondly, the work using fuzzy techniques to manage an agent's interactions, especially in bilateral auctions (however this is dealt with in Section 2.2.5). Finally, other agents developed for the TAC.

First we consider bidding in multiple simultaneous auctions. Preist designed an algorithm for agents that participate in multiple English auctions [Preist et al., 2001]. The algorithm proposes a co-ordination mechanism that can be used in cases where all the auctions terminate simultaneously, and a learning method to tackle auctions that are terminating at different times. However, their strategy is targeted at a single auction protocol (English) and the goods are not inter-dependent. Byde also describes a dynamic programming approach for agents that participate in multiple English auctions to buy a single item [Byde, 2001b]. Moreover, in [Byde, 2001a], the dynamic programming approach is compared with other algorithms in order to determine its quality. As a result, the dynamic programming approach is shown to be effective. However this method is only for one kind of auction (English) and it only deals with purchasing one item. We also believe it is difficult to extend this approach to a time constrained environment, such as the TAC, because of the heavy computational demands of this technique.

The framework presented in [Byde et al., 2002] enables an agent to make rational decisions across multiple heterogeneous auctions (English, Dutch, First-price sealed-bid and Vickrey auctions). It uses a fixed-auction strategy and a fixed threshold strategy to estimate the expected utility of a bid and then use a heuristic algorithm to approximate this decision making behaviour. However again there is no notion of purchasing interrelated goods.

Anthony *et al.* [Anthony and Jennings, 2002] also propose an approach for agents to bid for a single item in English, Dutch, and Vickrey auctions. The agent decides what to bid based on four parameters: (i) the remaining time; (ii) the number of remaining auctions; (iii) the desire for bargain; and (iv) the desperateness of the agent. The overall strategy is to combine these four tactics using a set of relative weights provided by the user. The agent also has a deadline for obtaining the good, but only one item is purchased. In an extension to this model [Anthony and Jennings, 2003], a genetic algorithm is used to search the effective strategies so that an agent can behave appropriately according to its assessment of its prevailing circumstances. Nevertheless, it still does not deal with interrelated goods.

Combinatorial auctions (discussed in Section 2.2) do deal with interrelated goods in that they allow bidders to bid for combinations of items. In contrast with the multiple auctions scenarios above, however, combinatrial auctions place the complexity on dealing with the interrelated aspect of the bidding on the auctioneer rather than on the

Table 4.17: Comparison among agents.
RS means risk seeking, RA risk averse and RN risk neutral (between RS and RA), — means information unavailable.

Agent	Price Prediction	Allocator	Attitude
ATTac	machine learning	ILP	RN
cuhk	average prices	heuristic search	RN
livingagents	average prices	search	RS
PainInNEC	—	Genetic algorithm	RA
Retsina	price matrix	Markov chain	RA
RoxyBot	price distribution	heuristic search	RA
sics	price distribution	branch-and-bound search	RA
SouthamptonTAC	Fuzzy reasoning	ILP	RN
UMBCTAC	average price	heuristic search	RS
Walverine	Walrasian competitive equilibrium	ILP	RS
whitebear	average price	greedy search	RN

bidding agent. Thus the auctioneer needs a winner determination algorithm to select a set of non-conflicting bids that maximise its revenue. This problem has been shown to be NP-complete [Fujishima et al., 1999] and, accordingly, a number of algorithms have been developed to achieve this according to various criteria (*e.g.*, anytime solutions [Fujishima et al., 1999], polynomial solutions [Dang and Jennings, 2002] and optimal solutions [Sandholm, 2002]).

In terms of other agents developed for the TAC, Table 4.17 compares the most successful agents from both competitions. *ATTac* uses machine learning techniques to obtain a model of the price dynamics based on the past data (*e.g.*, the data in the seeding round) to predict the closing prices of the hotels in the future. It also uses mixed-integer linear programming (ILP) to find the optimal allocation of the goods [Stone et al., 2001]. *cuhk* agent is composed of a cost estimator, an allocation and acquisition solver and bidders. It uses a greedy, heuristic search to find the travel packages for customers. *livingagents* [Fritschi and Dorer, 2002] bases its decisions on closing price data for the various hotels in past games and it buys all the flights needed at the beginning of the game. It also buys/sells entertainment tickets at a fixed price of 80. It makes bids for the needed hotels only once during the game again at a fixed price (of 1001). *PainInNEC*'s strategy is a combination of heuristics and a genetic algorithm based optimisation method, which outputs the goods to buy and sell given the predicted auction clearing prices and customers' preferences. *Retsina* uses a Markov Chain Monte Carlo approach to allocate the goods to its customers and it uses a matrix learned from past games to predict the hotel's future prices. *RoxyBot* [Greenwald and Boyan, 2001] decides the goods to bid for based on heuristic search techniques and applies a marginal utility calculator to determine the value of the goods. *sics* uses *pricelines* for price prediction and the optimiser performs branch-and-bound search for the best solutions. *UMBCTAC* balances the minimal risk and the maximum return to find the best travel plan for its customers. *Walverine* predicts the hotel closing prices by calculating the Walrasian competitive equilibrium of the game. *whitebear* uses a randomised greedy algorithm to calculate the price of each commodity bought or sold and uses Bayesian analysis to compute the minimum and average value of the flight's determinant factor.

As can be seen from the above discussion, there are a number of commonalities between the designs. Firstly, a variety of AI techniques including fuzzy reasoning, machine learning, planning, Markov decision making and heuristic search are used for making predictions about the likely future state of affairs. Thus most agents keep a record of the hotel closing prices and use a variety of methods to predict subsequent hotel closing prices in order to allocate travel packages to customers. Secondly, a number of the agents adapt their bidding behaviour in response to environmental changes. Such adaptation includes our agent varying its bidding behaviours (as described in Section 4.2.8), ATTac which varies the number of flights it buys at the beginning of the game, and whitebear which postpones some flight ticket purchases until after it learns the hotel prices.

4.5 Summary

This chapter details the design, implementation and evaluation of SouthamptonTAC, an agent that successfully participated in both the TAC-01 and the TAC-02 competitions and that employs a range of fuzzy techniques at its core. Specifically, it uses fuzzy pattern recognition to determine the type of environment it is situated in and then uses an adaptive bidding strategy to change its strategy depending on this assessment. In entertainment auctions, the agent continuously observes the current market asks/bids and uses fuzzy set techniques to extend the asks/bids it will accept by decreasing the similarity degree of the fuzzy sets. Moreover, the agent uses a fuzzy reasoning technique (Sugeno controller) to predict the hotel closing prices given the prevailing market conditions.

SouthamptonTAC has been shown to be successful across a wide range of TAC environments (in both competitions, as well as in our controlled experiments). Naturally the strategies that have been employed are tailored to the specific auction context of the competition (as is any agent strategy for any other auction context!). Nevertheless, we believe that the TAC domain exhibits a number of characteristics that are common to many real-world, on-line trading environments. These attributes include a time constrained environment, network latency, unpredictable opponents, multiple heterogeneous auction types and the need to purchase interrelated goods. Given this, we believe that a number of technologies and insights from our work are applicable in a broader agent-mediated e-commerce context.

In more detail, we believe the following methods can also be used in other areas of agent mediated e-commerce. Firstly, our fuzzy reasoning methods for predicting hotel closing prices can be reused in other multiple auction applications (as we demonstrate in

Chapter 5). Secondly, our method for an agent assessing its environment is applicable in more general settings. We believe that when agents have to use heuristic strategies it is likely that no one heuristic is likely to be best for all cases. Therefore such agents need the ability to tailor their strategy according to their assessment of the prevailing situation. To do this they need to be able to determine what environment they are in so that they can best respond. For example, in the general case of multiple auctions, the environment sensor can work based on the previous clearing prices, the type of participants in the auctions, the number of participants in the auctions and so on. Thirdly, through adaptation the agent can adjust its behaviour between three kinds of environments: competitive, non-competitive and semi-competitive. While attempting to settle these things in advance and not responding to the prevailing context may sometimes work (even in repeated encounters), it can produce brittle behaviour that is not robust in a wide variety of circumstances. Nevertheless some degree of prior analysis is essential to set the basic parameters to approximately correct values otherwise the agent may take a long time before it starts to perform effectively. Thus, our agent was tuned between rounds based on past performance and the risk attitude of its opponents in the competition. This was possible because the opponents were known in advance for the semi-finals and finals and because the opponents' behaviours can be studied in previous rounds. Fourthly, while an agent should certainly be responsive to its prevailing context; it should not respond to each and every minor perturbation in the environment. For example, in hotel bidding, the current ask prices of each auction change on a minute by minute basis. Now it may be that the agent believes it can obtain an improvement in utility by switching its customers between the good and the bad hotels. However if this improvement is only small then the agent should not switch because its estimation is based on uncertain predictions and if these predictions are slightly out then there may not be a real improvement. Moreover, by making such a switch the agent is taking a risk because it may not be able to off-load those hotels that it has already bought and so it may have to pay for hotels that it cannot use.

TAC has certainly shown itself to be an interesting challenge problem, however one of its major shortcomings is that it does not incorporate the standard English auction and this is by far the most widely used auction type on the Internet today. To rectify this, we decided to generalise TAC to the case where an agent has to buy multiple goods for multiple English auctions. Another limitation of TAC is that the preferences of the customers are specific numbers (*e.g.*, staying in a good hotel has a utility of 150). However in many real life scenarios a user often has a more flexible preference structure. For example, a user may want to buy a laptop for about $1,500$ pounds. This flexibility can be achieved by fuzzifying the preferences of the customer. To deal with these issues we need to extend our agent design by allowing it to be more adaptive to its prevailing circumstances. This we detail in Chapter 5.

Chapter 5

A Neuro-Fuzzy Bidding Strategy for Buying Multiple Goods in Multiple English Auctions

Having designed the strategy for the TAC, we now move to a more general multiple auction context in which an agent seeks to buy multiple good for multiple English auctions. The ensuing agent designs build upon SouthamptonTAC but introduce still greater adaptability into the agents strategy so that it can respond effectively to changes in the marketplace.

5.1 Overview

We focus on English auctions because they are by far the most common protocol for agent-mediated e-commerce. In our particular case, each such English auction sells a single unit of the desired good and this good may be described by multiple attributes[1] (*e.g.*, in auctions selling flights, the goods may be described by their dates of departure and return, by their carrier, and the class of ticket being bought). Also we assume the goods are independent; thus the failure to obtain one of the goods does not influence the availability of other goods.[2] Bids for these goods must be at least $h(a)$ pounds larger than the previous price to be valid, where $h(a)$ is the increment of bids decided by the auctioneer. If an agent bids successfully, it becomes the active agent which is holding the bid. It may, of course, be subsequently out-bid. Auctions respond to any bid before they close and their good is allocated to the active bid holder when the auction closes.

In a stand-alone English auction, the bidding strategy is simple. The agent's dominant strategy is to bid a small amount more than the current highest bid and stop when

[1] The goods have multiple attributes, but the buyer agents only bids on price. Thus, our work differs from the case where agents bid on multiple attributes (as discussed in section 5.6).

[2] Dealing with inter-dependent goods (as per the TAC) involves significant amount of domain knowledge (*e.g.*, the relationship between the auctions, some are replacements and some are compensate) and thus it is more difficult to generate generic strategies and results.

the user's valuation is reached (Section 2.2.5). By adhering to this dominant strategy, the good is always allocated to the bidder who values it most highly. However, when there are multiple auctions running at the same time, bidding is much more complex and has much greater uncertainty. This is because: (i) the auctions have varying start and end times (thus comparisons need to be made between auctions that are nearing completion (and probably have a high ask price) and auctions that are near the beginning (and probably have a low ask price); (ii) the participating agents are likely to adopt a variety of different strategies (*e.g.*, some may bid aggressively in the earlier auctions in order to ensure they get the desired goods, while others may wait until later auctions to see if they can obtain bargains), and (iii) the agents are likely to have different sets of auctions that they consider bidding in (driven by their own deadlines), thus the set of agents in a given auction will vary and so, consequently, will the supply/demand. In some variants of the English auction, there is a strict deadline for when the auction will finish and this promotes a strategy of trying to place a bid at the last moment so that no other agent has a chance to place a higher bid (this is called "sniping" [Rust et al., 1991]). To counteract such end effects, our auctions have a soft deadline; that is, they do not close until a fixed period after the last bid is placed (as per Yahoo!Auctions (http://auctions.shopping.yahoo.com) and Auction Universe [Miller, 1999]). This means sniping is not effective and the auctions are akin to the standard English one.

Moreover, we consider the case where the agent has to purchase multiple goods from the ongoing auctions which, we believe, is more realistic than just having to purchase a single good from this set. Finally, and again for maximum generality in our solution, we assume the agent may be acting on behalf of multiple customers. This means it needs to allocate the goods it has purchased or is currently holding to the various customers in order to maxmise its return (as per TAC).

Given this context, we developed (for the first time) a bidding algorithm that buys multiple units of the desired good from the available auctions. This algorithm operates in the following way. It calculates what it believes is the best set of auctions to bid in (it does this by predicting the auctions' closing prices, using a fuzzy neural network, allocating the goods to its various customers and then calculating the satisfaction degree of the allocation). However the prediction of this optimal set of auctions is highly uncertain because it depends on the strategies, profiles and reservation prices of an arbitrary set of agents. Therefore rather than just bidding in this set, an agent could also decide to bid in other auctions that are likely to have broadly the same outcome because by doing so its chances of obtaining the goods are increased (more places to buy from) and the satisfaction degree compared with what is believed to be the optimal is still reasonably high. Specifically, our algorithm adopts the heuristic of bidding in this expanded set of auctions (the best set, plus those that have a similar satisfaction degree) in the order of increasing auction end time. Thus we term it an *Earliest Closest First* (ECF) algorithm. Moreover, as the goods are composed of multiple attributes, the agent may have to make *trade-offs* between them in its bidding in order to best satisfy the users' preferences (as discussed in Section 1.2). Thus, for example, a user may ideally wish to fly out on a Saturday, return the following Wednesday and fly with British Airways, but would be willing to accept (for a lower price) flight dates of Friday and Wednesday with Quantas (a BA partner). To allow such flexibility (or imprecision), we choose to model preferences using fuzzy sets (for the reasons outlined previously). Specifically, the agent's preferences are private information that include: (i) valuations v for the good (expressed as fuzzy sets); (ii) the ratings for different values of the good's attributes (expressed as fuzzy sets); and (iii) the weights which balance the valuation and the other attributes. By means of an example, consider the case of a student who wants to buy a flight ticket to New York. She prefers to buy a cheap ticket ("cheap" is a fuzzy term). Thus the lower the price, the higher her degree of satisfaction. Ideally she wants to depart on Saturday, but it is acceptable to go on Friday or Sunday. Here, the date can be denoted as a triangular fuzzy number (Section 3.1.2), where "Saturday" has the highest satisfaction degree, and "Friday" and "Sunday" have lower ones. Also, the airlines she likes can also be a fuzzy number where the memberships of the fuzzy set "like" are given by her satisfaction on the airlines. Finally, she can express the relative importance of the attributes of price, date and airline, by assigning them the appropriate weights.

To cope with the uncertainty inherent in the multi-auction context and to make trade-offs between the different variants of the goods available is a complex decision making problem. Ideally the closing price of the auctions would be known first in order to calculate the satisfaction degree of a bid. However, since the closing price is only known after the auction is closed, it is important for the agent to make predictions about the likely closing prices. By so doing, the agent can determine whether it should place a bid at the current moment or it should delay because better deals may subsequently become available. In our previous work, we successfully used adaptive fuzzy inference methods for this task in continuous double auctions (CDAs) (Chapter 3) and the TAC (Chapter 4). However, in both cases, the parameter adaptation of the fuzzy rules was limited. For example, our agent for the CDA can only adapt its parameters in a single direction of change (*e.g.*, all the parameters are bigger in a competitive environment, see Section 3.4 for more details). However this is inappropriate for the multi-auction context because each parameter in the strategy should be adjusted according to its actual direction of change (*e.g.*, the centre of the membership function for the fuzzy set "medium" may need to go up and the corresponding width may need to go down to reflect the fact that this fuzzy set should cover a smaller range of higher values). To rectify this, we exploit *fuzzy neural networks* (FNNs) [Jang, 1993] since these can do the fuzzy reasoning and, through learning (both off-line and on-line), can adjust the parameters of the fuzzy terms and the consequent output as the auctions progress. This adaptation enables the agent's bidding behaviour to better reflect the current state of its environment (hereafter we call this strategy FNN, and the agent the FNN Agent).

The work described in this chapter advances the state of the art in the following ways. First, we develop, for the first time, a practical agent bidding algorithm (ECF) for obtaining multiple goods in multiple overlapping English auctions. Second, fuzzy neural networks are developed that can make predictions about the closing prices and adapt the parameters in the neural network through off-line and on-line learning to suit the environment the agent is situated in. Through empirical evaluation the agent which uses ECF combined with the FNN is shown to be effective in a wide range of situations. Finally, by exploiting a fuzzy set representation of the user's preferences, the strategy is able to make effective trade-offs between the various attributes of the goods the agent purchases and can cope with the inherent imprecision/flexibility that often characterises a user's preferences. All the common issues discussed in Section 1.2 are addressed here: price prediction (Section 5.3.1), adaptation (Section 5.3.2), flexible bidding (Section 5.2), risk attitude adjusting (Section 5.2), and trade-off attributes (Section 5.3.3).

The rest of this chapter is structured as follows. Section 5.2 describes the ECF bidding algorithm and Section 5.3 describes how the bidding strategy operates. Section 5.4 gives an example of a flight auction scenario in which the operational effectiveness of our algorithm is evaluated. Section 5.5 actually provides the systematic empirical evaluation and benchmarks our strategy against a number of others that have been proposed in the literature. Section 5.6 discusses the related work in multiple auction bidding, multi-attribute auctions, and fuzzy-based bidding methods. Finally, Section 5.7 concludes.

5.2 The Earliest Closest First Bidding Algorithm

This section details the ECF algorithm. First, we introduce some specific terms in our auction context and then the algorithm is described.

There are multiple auctions in the market, each selling one unit of good. There are three states for each auction: (i) *waiting*: before its start time, nothing happens in this auction; (ii) *running*: the auction is open for bids; and (iii) *closed*: the auction finishes when the market time is bigger than the auction's closing time and there have been no active bids in the auction for a fixed period.

Each agent aims to buy multiple goods, thus it considers bidding if and only if the sum of the bids it holds (*i.e.*, those auctions in which it is the active bidder) and owns (closed auctions in which the agent won) is less than the number of good it desires.[3] If it decides to bid, the agent needs to determine which auctions it should bid in. To do so, it first determines the auctions that it believes best satisfy the user's preferences (calculation detailed in section 5.3.1) given its expectation about the closing prices (calculation

[3]The case where the agent bids in more auctions than the number of goods it wants is not considered here.

detailed in section 5.3) of each auction. Then, rather than placing a bid in the selected auctions immediately, it bids in auctions that close earlier than the selected auctions and have an evaluation “close” (a fuzzy term) to that of the selected auctions. The intuition here is that given the significant degrees of uncertainty that exist, precise calculations about the closing prices are simply not reliable and an auction that appears slightly less promising may well turn out to be better. Given this, the agent should consider bidding in auctions that have broadly similar expected returns so as to increase its chances of obtaining the item (by participating in more auctions), while ensuring the likely return is one of the highest. Thus, for each good it desires, if there are such close auctions, the agent will bid in the selected auctions in order of increasing closing time (*i.e.*, bid first in the one that is going to close first, then in the one that will close next, and so on) (hence the name Earliest Closest First). The degree of closeness that is required to trigger bidding is captured by the threshold ($\lambda \in [0,1]$). Then if the difference is within λ, the agent will bid in the auction. In this sense, the choice of λ represents the risk attitude of the user (Section 3.3.2). If λ is high, the agent can be viewed as being risk averse because it bids in many more auctions in order to maximise its chance of getting the good (although it is likely to get a less satisfactory set of goods because it may accept a higher ask price). If λ is low, the agent is taking a greater risk because it is trying to obtain a high degree of satisfaction (but by not bidding in as many auctions it has a lower chance of actually being successful). If λ is inbetween, the agent is striking a balance between the two positions (which is here termed risk neutral).

In more detail, the decision making algorithm ECF is given in Figure 5.1. In this algorithm, n_{active}, n_{own}, and n_{demand} are the number of goods the agent holds, owns and desires respectively. An explanation of the algorithm’s key functions are as follows:

- The function $AuctionRunning()$ (line 1) returns $true$ if there are still available auctions to bid in, $false$ otherwise.
- The function $update()$ (line 2) returns changes in the auctions since they were last monitored. Such changes include whether the agent is holding an active bid or has obtained the good, the updated ask price of each auction, the transaction price for any auctions that recently closed, and the number of auctions left to bid in.
- The function $predict()$ (line 3) predicts all auction’s closing prices given the current market situation and the history transaction prices. The agent uses a FNN to predict the closing prices in this chapter (see section 5.3.1).
- The function $allocate()$ (line 4) allocates all the goods the agent owns and possibly owns to its user according to their preferences. The later includes the goods the agent holds in waiting or running auctions. The way that we assign the auction to the user is through an assignment algorithm discussed in section 5.3.4.
- The function $toBid(g)$ (lines 6, 7 and 16) returns the id of the auction in which good g is believed to have the highest degree of satisfaction given the prediction

PROCEDURE $ECFbid()$

while $n_{active} + n_{own} < n_{demand}$ or $AuctionRunning()$ **do**
    $update()$ *//get updated market information*
    $predict()$ *//predict auctions' closing prices*
    $allocate()$ *//allocate goods to user*
    for all $g \in G$ **do**
        if $toBid(g) \geq 0$ **then**
            $s_{best}(g) \leftarrow evaluate(toBid(g), g)$
        else
            $s_{best}(g) \leftarrow 0$
        end if
    end for
    $L \leftarrow RunningAuctions()$ *//order the auctions by their closing time*
    for all $a \in L$ **do**
        for all $g \in G$ **do**
            $s \leftarrow evaluate(a, g)$
            if $toBid(g) \geq 0$ **then**
                $\lambda \leftarrow chkThreshold()$ *//determine the risk attitude*
                if $(s \geq s_{best}(g))$ or $(s_{best}(g) - s \leq \lambda)$ **then**
                    $bid(a)$ *//submit a bid in auction a*
                    break
                end if
            else if $s > 0$ **then**
                $bid(a)$
                break
            end if
        end for
    end for
end while

Figure 5.1: The Earliest Closest First bidding algorithm.

of the closing prices. This is the output of the allocation in $allocate()$. If there is a best auction for buying good g given the allocation, the returned value will be the auction id; otherwise, the agent will bid in any auction in which the current satisfaction degree is positive.

- The function *RunningAuctions*() (line 12) returns a list L of all the currently running auctions in ascending order of their end times.
- The function $evaluate(a,g)$ (lines 7 and 15) returns the evaluation of auction a given the agents' preference for good g at its current ask price. This evaluation balances both price and the other attributes of the goods using a fuzzy aggregation method (see section 5.3.3).
- The function $chkThreshold()$ (line 17) returns the threshold parameter for the agent given the current situation of the auction market. The threshold λ is determined by the number of auctions that have a positive satisfaction for the agent at that particular moment in time. Here the general rule for choosing λ is: the fmore such auctions there are, the smaller λ should be. This captures the intuition that if there are many chances for the agent to win the good, it can have a higher threshold so that it will have a higher satisfaction degree for the purchased goods (and vice versa). The experiments in 5.5.4 shows the effect of different λs on the performance of the agent.
- The function of $bid(a)$ (lines 19 and 23) places a bid in auction a. The price to place is the ask current ask price of the auction plus the bid step $h(a)$.

To realise this algorithm, a number of prediction techniques are needed (line 3). Here we use fuzzy neural networks and their application is discussed in sections 5.3.1 and 5.3.2. Moreover, an evaluation method is needed for ranking the various auctions (section 5.3.3), and a good allocation method is needed to decide which auctions should be assigned to which user (section 5.3.4).

5.3 The FNN Strategy

This section details the FNN strategy. First, we describe how the FNN is structured and how it operates to obtain the predicted closing price. Second, the learning algorithm of the FNN is described. Third, the way of evaluating auctions are introduced. Finally, the allocation method that allocates the goods to the user is given.

5.3.1 FNN Prediction

To reason about the expected closing price of each auction, the FNN agent considers a per auction reference price (p_{ref}), the order in which the auctions are due to close ($o_{auction}$), and the auction's current ask price (p_{now}). Here the reference price represents a likely value at which the auction will close for that particular variant of the good. It is computed by considering the transaction prices of auctions that have previously sold the

FUNCTION $getRefPrice(i)$

if $getRelPrice(i) \geq 0$ **then**
    Return $getRelPrice(i)$ *//return the average price in a single game*
else
    Return $getAvgPrice(i)$ *//return the historical average price in past games*
end if

Figure 5.2: Reference price (p_{ref}) calculation for the FNN. $getRelPrice(i)$ returns the average transaction price of all the auctions (with the same attributes as auction i) that have closed since the agent started bidding. $getAvgPrice(i)$ returns the historical average transaction price in the history for auctions (with the same attributes as auction i).

specified good and the average transaction price in the history records for the specified good (see function $getRefPrice(i)$ in Figure 5.2). The FNN agent records such data examples and removes some of the oldest data to ensure it only learns based on the latest data. Every time the FNN agent detects that an auction has closed, it will re-train the neural network on the updated set of data (the online updating takes less than one second on a standard PC) which involves adjusting the parameters to better reflect the prevailing circumstances (see section 5.3.2).

R_1:	If p_{ref} is high and $o_{auction}$ is early then p_{close} is very_high.
R_2:	If p_{ref} is high and $o_{auction}$ is medium then p_{close} is high.
R_3:	If p_{ref} is high and $o_{auction}$ is late then p_{close} is medium.
R_4:	If p_{ref} is medium and $o_{auction}$ is early then p_{close} is high.
R_5:	If p_{ref} is medium and p_{now} is high then p_{close} is high.
R_6:	If p_{ref} is medium and $o_{auction}$ is medium and p_{now} is medium then p_{close} is medium.
R_7:	If p_{ref} is medium and $o_{auction}$ is medium and p_{now} is low then p_{close} is medium.
R_8:	If p_{ref} is medium and $o_{auction}$ is late and p_{now} is medium then p_{close} is medium.
R_9:	If p_{ref} is medium and $o_{auction}$ is late and p_{now} is low then p_{close} is low.
R_{10}:	If p_{ref} is low and p_{now} is high then p_{close} is high.
R_{11}:	If p_{ref} is low and p_{now} is medium then p_{close} is medium.
R_{12}:	If p_{ref} is low and $o_{auction}$ is early and p_{now} is low then p_{close} is low.
R_{13}:	If p_{ref} is low and $o_{auction}$ is medium and p_{now} is low then p_{close} is low.
R_{14}:	If p_{ref} is low and $o_{auction}$ is late and p_{now} is low then p_{close} is low.

Table 5.1: The FNN agent's rule base.

To predict the closing prices of the auctions, fuzzy reasoning is used (as per Sections 3.2 and 4.2.7). Through analysing our experimental data, we found that the reference price, auction's closing order and current price are closely correlated with the actual closing prices. Thus, fuzzy rules (defined in table 5.1) are designed to capture the relation among these factors. In particular, p_{ref} is expressed using the fuzzy linguistic terms: *high*, *medium* and *low*; $o_{auction}$ is expressed by *early*, *medium* and *late*; and p_{now} is expressed by *high*, *medium* and *low*. The consequent output is expressed as the integer numbers *very_high*, *high*, *medium* and *low*.

Thus, the FNN agent takes three inputs (p_{ref}, $o_{auction}$, p_{now}) and has one output (the expected auction closing price p_{close}). According to the rule base above, we developed a FNN with 5 layers (as shown in Figure 5.3). The input variables correspond to the nodes in layer 1. The nodes in layer 2 correspond to individual rules for reasoning about the closing prices of the auctions. Nodes in layer 3 calculate the relative importance (weight) of each of these rules. The nodes in layer 4 combine the output of each rule into the overall output. Finally, the node in layer 5 sums up all the outputs in layer 4 and gives the predicted auction closing price. In more detail:

- **Layer 1**: each node in this layer generates the membership degrees of a linguistic label for each input variable (*e.g.*, reference price is low or there are a big number of auctions left). Specifically, the *i*th node performs the following (fuzzification) operation:

$$O_i^{(1)} = \mu_{A_i}(x) = e^{-\frac{(x-c_i)^2}{2\delta_i^2}} + \gamma, \tag{5.1}$$

  where $O^{(1)}$ is the output of layer 1 (*i.e.*, the membership degree with respect to the corresponding fuzzy sets), x is the input to the ith node, and A_i is the linguistic value (high, medium, low, etc.) associated with this node. The set of parameters (c_i, δ_i) determines the shape of the membership function.[4] These parameters can be adapted by learning (as we will explain in section 5.3.2). γ is a very small number (here we choose 0.00001) that avoids the output in layer 1 becoming zero.
- **Layer 2**: each node in this layer calculates the firing strength (the product of all the inputs, and it is in the range of $[0,1]$) of each rule (in table 5.1) via the multiplication operation:

$$O_i^{(2)} = w_i = \Pi_{j \in S_i^{(1)}} \{\mu_{A_j}\}, \tag{5.2}$$

  where $S_i^{(1)}$ is the set of nodes in layer 1 which feed into node i in layer 2, and w_i is the output of this node (*i.e.*, the strength of the corresponding rule).
- **Layer 3**: the *i*th node of this layer calculates the ratio of the *i*th rule's firing strength to the sum of all rules' firing strengths:

$$O_i^{(3)} = w_i' = \frac{w_i}{\sum_{j \in S^{(2)}} w_j}, \tag{5.3}$$

[4] Here we assume that this is a Gaussian function. This is because it has non-zero derivatives throughout the universe of discourse and is therefore easy to implement. Also, the derivatives of a Gaussian function are continuous and smooth, thus, it can produce good training performance.

where $S^{(2)}$ is the set of nodes in layer 2. This ratio indicates the relative importance of each rule.

- **Layer 4**: the ith output of the node is calculated by:

$$O_i^{(4)} = r_i \sum_{j \in S_i^{(3)}} w'_j, \tag{5.4}$$

where $j \in S_i^{(3)}$ is the set of nodes in layer 3 that feed into node i. The output of this layer combines all the outputs of the rules that have the same consequent output.

- **Layer 5**: the single node in this layer aggregates the overall output of the FNN (*i.e.*, p_{close}) as the summation of all incoming signals:

$$O^{(5)} = \sum_{i \in S^{(4)}} \left(r_i \sum_{j \in S_i^{(3)}} w'_j \right), \tag{5.5}$$

where $j \in S^{(4)}$ is the set of nodes in layer 4.

Given the nature of this decision making task, it is important that the various parameters of the FNN algorithm fit the prevailing context as accurately as possible. This is achieved via off-line and on-line learning (see section 5.3.2 for more details). In the former case, a number of simulated games[5] are used to set the initial parameters of the FNN. After this, the agent can be used in an operational setting to actually purchase goods. In the latter case, the agent keeps track of the various auctions and when changes occur (*e.g.*, when an auction has closed or the ask prices change) they are fed into the FNN as new training examples. These examples are weighted more highly than older ones and so enable the agent to better reflect prevailing circumstances (but without being completely reactionary to the last set of changes).

To illustrate the operation of this architecture consider the following example. Let the reference price be $p_{ref} = 20$, the current price be $p_{now} = 15$, and assume the auction is the fourth one to close (*i.e.*, $o_{auction} = 4$). Thus $(20, 4, 15)$ is fed into the FNN. Let these values be assigned the following membership degrees of the following fuzzy set: $medium(20) = 0.8$, $late(4) = 0.7$ and $low(15) = 1$. These values are then the respective outputs of nodes 2, 6 and 9 in layer 1. Then, taking R_9 as an example, the output of node R_9 in layer 2, by equation (5.2), is $0.8 \times 0.7 \times 1 = 0.56$. Then, suppose the sum of all the

[5]The initial parameters of the FNN can either be set directly by the user or can be set by playing simulated games. A simulated game is played by strategies or humans in a test environment where the various agents compete but money does not actually change hands. In the cases where such games are not available, users can monitor real multiple auction websites and collect relevant data (since the trading history of most auction sites is readily available). However, if the user is confident about their parameter settings, the agent can be put directly into practice without going through the simulated games. For example, on Yahoo!Auctions, the final outcomes of the auctions are stored. Thus the prices paid for *digital cameras* can be found at: http://csearch.auctions.shopping.yahoo.com/csearch?sb=desc&alocale=0us&acc=us&desc=digital+camera.

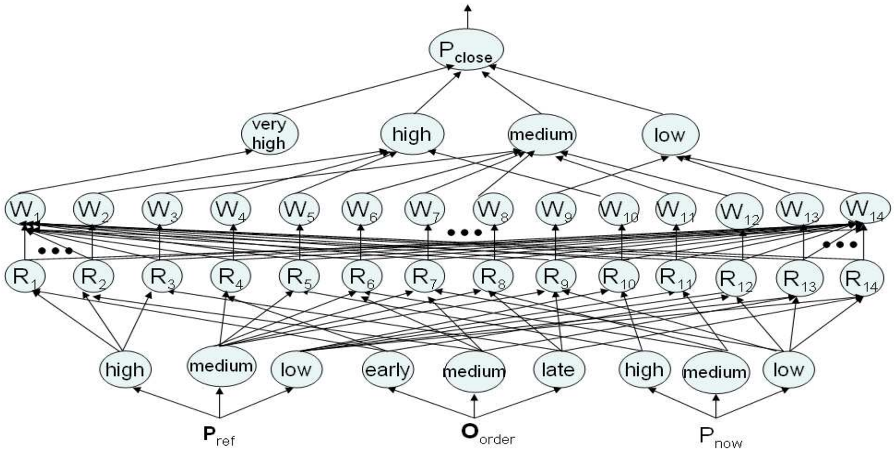


Figure 5.3: Overview of the FNN architecture.

firing strengths in layer 2 is 3. The weight of R_9 (output of w_9) will be $0.56/3 = 0.187$. Thus, R_9 contributes 0.187 among all the rules. After this, in layer 4, there are 3 rules that have the same consequent output, which is "low" (*i.e.*, R_9, R_{13}, and R_{14}). Let $r_{low} = 18$, $w_{13} = 0.3$, and $w_{14} = 0.5$, in which case the output of node "low" is, by equation (5.4), $18 \times (0.187 + 0.3 + 0.5) = 17.7$. Finally, the output of layer 5 is the sum of all the outputs of layer 4 (one of which will be the 17.7 coming from the "low" node).

Finally, to guarantee the predicted closing price is alway higher than current ask price, the predicted auction closing price will be

$$P^*_{close} = \max\{p_{close}, p_{now}\}, \tag{5.6}$$

where p_{close} is the output of the FNN and p_{now} is the current ask price of the auction.

5.3.2 FNN Learning

The FNN agent involves two types of learning: off-line and on-line. In either case, however, the same basic method is used. Given the training data x_i ($i = 1, 2, 3$), the desired output value Y, and the fuzzy logic rules (from table 5.1), the parameters of the membership functions for the FNN's input variables are adjusted by supervised learning. Here the goal is to minimise the error (E) function for all the training patterns:

$$E = \sum_j \frac{1}{2}(Y_j - O_j^{(5)})^2, \tag{5.7}$$

where Y_j is the actual closing price of pattern[6] j and $O_j^{(5)}$ is the predicted closing price of the FNN for pattern j. For each set of training data, starting at the input nodes, a forward pass is used to compute the activity levels of all the nodes in the network. Then starting at

[6] Here a pattern is a training example (*e.g.*, for the previous example, a pattern might be $\{\{20, 4, 15\}, 17.7\}$, where given the inputs $\{20, 4, 15\}$ the actual output is a price of 17.7).

the output nodes, a backward pass is used to compute $\frac{\partial E}{\partial O}$ for all the hidden nodes. In our FNN agent, the parameters that get adjusted during learning are the consequent output of each rule (r_i in layer 4) and the centre and width of the Gaussian membership functions for each of the fuzzy terms (c_i and δ_i in layer 1). For the parameters of r, the learning rule of the FNN agent is based on gradient descent optimisation [Rumelhart et al., 1986]:

$$r(t+1) = r(t) + \eta\left(-\frac{\partial E}{\partial r}\right), \tag{5.8}$$

where η is the learning rate ($\eta \in [0.001, 0.01]$).

Thus, the learning rule for adjusting the parameters of r_i in layer 4 and (c_i, δ_i) in layer 1:

$$\frac{\partial E}{\partial r_i} = \frac{\partial E}{\partial O^{(5)}}\frac{\partial O^{(5)}}{\partial O_i^{(4)}}\frac{\partial O_i^{(4)}}{\partial r_i} = (O^{(5)} - Y)\sum_{j\in S_i^{(3)}} w'_j, \tag{5.9}$$

Hence r_i is updated by:

$$r_i(t+1) = r_i(t) - \eta(O^{(5)} - Y)\sum_{j\in S_i^{(3)}} w'_j. \tag{5.10}$$

For parameters c_i and s_i, the conjugate gradient algorithm [Johansson et al., 1990] is used since it is faster than the gradient descent optimisation. Suppose α is the parameter we are interested in and the gradient of the tth iteration of the learning is g_t ($t > 1$), then the new search direction is to combine the new steepest descent direction with the previous one, that is,

$$p_t = -g_t + \beta_t p_{t-1}, \tag{5.11}$$

where by using Fletcher-Reeves [Fletcher and Reeves, 1964] update,

$$\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}}. \tag{5.12}$$

Thus, the adaptive rule of c_i in layer 1 is as follows (where $S_{-i}^{(2)}$ means the set of nodes in layer 2 that are connected with node i in layer 1):

$$\begin{aligned}\frac{\partial E}{\partial c_i} &= \sum_{m\in S_{-i}^{(2)}}\left(\frac{\partial E}{\partial O_m^{(2)}}\frac{\partial O_m^{(2)}}{\partial O_i^{(1)}}\frac{\partial O_i^{(1)}}{\partial c_i}\right)\\ &= \sum_{m\in S_{-i}^{(2)}}\left(\sum_{k\in S_{-m}^{(3)}}\left(\frac{\partial E}{\partial O_k^{(3)}}\frac{\partial O_k^{(3)}}{\partial O_m^{(2)}}\right)\frac{\partial O_m^{(2)}}{\partial O_i^{(1)}}\frac{\partial O_i^{(1)}}{\partial c_i}\right),\end{aligned} \tag{5.13}$$

where

$$\frac{\partial E}{\partial O_k^{(3)}} = (O_1^{(5)} - Y) r_k; \tag{5.14}$$

$$\frac{\partial O_k^{(3)}}{\partial O_m^{(2)}} = \begin{cases} \frac{\sum_k w_k - w_m}{(\sum_k w_k)^2} & \text{if k=m,} \\ \frac{-w_m}{(\sum_k w_k)^2} & \text{otherwise;} \end{cases} \tag{5.15}$$

$$\frac{\partial O_m^{(2)}}{\partial O_i^{(1)}} = \frac{w_m}{O_i^{(1)}}; \tag{5.16}$$

$$\frac{\partial O_i^{(1)}}{\partial c_i} = e^{-\frac{(x_i - c_i)^2}{2\delta_i^2}} \frac{(x_i - c_i)}{\delta_i^2}. \tag{5.17}$$

So the adaptive rule of c_i is:

$$c_i(t+1) = c_i(t) + \eta p_t, \tag{5.18}$$

where $p_t = -g_t + \beta_t p_{t-1}$ and $g_t = \frac{\partial E}{\partial c_i}$.

Similarly, from (5.14), (5.15), and (5.16) the adaptive rule of δ_i is derived as:

$$\frac{\partial E}{\partial \delta_i} = \sum_{m \in S_{-i}^{(2)}} \left(\sum_{k \in S_{-m}^{(2)}} \left(\frac{\partial E}{\partial O_k^{(3)}} \frac{\partial O_k^{(3)}}{\partial O_m^{(2)}} \right) \frac{\partial O_m^{(2)}}{\partial O_i^{(1)}} \frac{\partial O_i^{(1)}}{\partial \delta_i} \right), \tag{5.19}$$

where

$$\frac{\partial O_i^{(1)}}{\partial \delta_i} = e^{-\frac{(x_i - c_i)^2}{2\delta_i^2}} \frac{(x_i - c_i)^2}{\delta_i^3}. \tag{5.20}$$

Hence the adaptive rule of δ_i becomes:

$$\delta_i(t+1) = \delta_i(t) - \eta p_t, \tag{5.21}$$

where $p_t = -g_t + \beta_t p_{t-1}$ and $g_t = \frac{\partial E}{\partial \delta_i}$.

5.3.3 *Evaluating the Auctions*

Given the expected auction closing prices, the agent needs to make a decision about which auctions to bid in. For ease of expression, we present this evaluation function[7] for the case where only price and one other attribute of the good are considered (but the concepts are equally applicable for arbitrary numbers of attributes). Given the user's preference on price and other attributes (as defined in section 5.4.1), the evaluations of the various factors need to be integrated. In fuzzy theory, the process of combining such individual ratings for an alternative into an overall rating is referred to as *aggregation*

[7] Such an evaluation function is used to evaluate the bidding strategy that considers more than one of the good's attributes in making its bidding choice. Thus, for example, all the benchmark strategies in section 5.5 exploit such a function.

[Yager, 1994]. Let w_p and w_q be, respectively, the weight of price and the other attribute that the agent is concerned with and u_p be the evaluation with respect to price and u_q the evaluation with respect to the other attribute. Intuitively, the role of the aggregation operator is to balance u_p and u_q and obtain an overall evaluation $u_{p,q}$ somewhere between the two values. There are three main aggregation operators that are commonly used and each of them has different semantics (conforming to different user objectives):

- Weighted average operator:

$$u_{p,q} = u_p w_p + u_q w_q. \tag{5.22}$$

  Using this operator means that even if one of the evaluations is very low, the overall output can still be reasonably high. For example, if the user does not like the time of the flight but it is very cheap, the overall evaluation can still be high.

- Weighted Einstein operator [Luo et al., 2003c, Luo et al., 2003d]:

$$u_{p,q} = \frac{u_p^{'} u_q{}'}{1 + (1 - u_p^{'})(1 - u_q^{'})} \tag{5.23}$$

  where $u_p^{'} = (u_p - 1)w_p$ and $u_q^{'} = (u_q - 1)w_q$. The above equation satisfies the characteristics of T-norms operators [Yager and Filev, 1994]. That is, if one evaluation is not satisfied (*i.e.*, $u_p = 0$ or u_q=0), the overall evaluation is 0. Intuitively, this corresponds to the situation where both evaluations must be more or less satisfied. For example, even if the flight ticket is free, the user cannot accept it since traveling after a specific date is totally useless (*e.g.*, he has very import meeting at that specific date).

- Weighted uninorm operator [Yager and Rybalov, 1996]:

$$u_{p,q} = \frac{(1-\tau)u_p^{'} u_q^{'}}{(1-\tau)u_p^{'} u_q^{'} + \tau(1 - u_p^{'})(1 - u_q^{'})}, \tag{5.24}$$

  where $u_p^{'} = \frac{(u_p - 1)w_p}{\max\{w_p, w_q\}} + 1$ and $u_q^{'} = \frac{(u_q - 1)w_q}{\max\{w_p, w_q\}} + 1$, $\tau \in (0, 1)$ is the unit element of this operator. The unit element can be regarded as a threshold: if both the evaluations are above the threshold, the overall evaluation is enhanced; if both are less than the threshold, the overall evaluation is weakened by each other; if there is a conflict between the two evaluations, the overall evaluation is a compromise. For example, if the user likes the date and price, the overall evaluation is very high; if the user hates the date and price, the overall evaluation is even lower; if the user likes the date but hates the price, then some intermediate value is chosen.

Since these operators are all plausible means of aggregating price and the other attributes, and none is necessarily superior in all cases, we need to empirically evaluate the impact of these operators on the performance of the agents. This we do in section 5.5.2.

5.3.4 Goods Allocation

The agent needs to allocate the goods it owns and potentially owns to its various customers in order to maximise the overall satisfaction degree. Thus good allocation takes place each time the agent accesses the market and when the market information has been updated. The goods are allocated to the agent's users optimally so as to maximise the sum of the users' satisfaction. Here this allocation process is regarded as an assignment problem which we solve using a shortest augmenting path algorithm[8] [Jonker and Volgenant, 1987].

In more detail, suppose D is a $d \times d$ square,[9]

$$D = \begin{pmatrix} \mathrm{s}_{11} & \cdots & \mathrm{s}_{1d} \\ & \cdots & \\ \mathrm{s}_{d1} & \cdots & \mathrm{s}_{dd} \end{pmatrix}$$

where d is the number of auctions, and s_{ij} is the satisfaction degree of the user's jth requirement for auction a_i:[10]

$$s_{ij} = \begin{cases} -evaluate(a_i, g_j) & \text{if } i \leq n_{demand}, \\ 99 & \text{if owns or holds a good in } a_i \text{ and } j > n_{demand} \\ 0 & \text{otherwise,} \end{cases}$$

where 99 is used to avoid the owned goods being allocated to the dummy nodes.

Given this input, suppose an allocation X is:

$$X = \begin{pmatrix} \mathrm{x}_{11} & \cdots & \mathrm{x}_{1d} \\ & \cdots & \\ \mathrm{x}_{d1} & \cdots & \mathrm{x}_{dd} \end{pmatrix}$$

where each row and each column has only a single 1 (*i.e.*, each auction is only allocated to one user). Thus,

$$\sum_{j=1}^{d} x_{ij} = 1,$$

[8] There are many ways to solve linear assignment (http://citeseer.ist.psu.edu/burkard98linear.html). By experience, however, we found this one is among the best. The important observation is that choosing how to assign goods to customers is an assignment problem and so can be solved efficiently (in $O(n^3)$ time) [Jonker and Volgenant, 1987].

[9] Here a square is needed in order to use the algorithm. The row represents the auction and the column represents the goods the agent desires. To use the algorithm, some dummy nodes may need to be added to make a square so that row number is equal to the column number.

[10] Our problem here is a maximisation problem, but in order to use the shortest augmenting path algorithm, we need put a minus before the evaluation value.

$$\sum_{i=1}^{d} x_{ij} = 1,$$

and the objective function to minimise is:

$$\sum_{j=1}^{d}\sum_{i=1}^{d}(s_{ij}x_{ij}),$$

where $x_{ij} = 0$ or 1.

Using this method, the agent can decide how to allocate the goods it owns and holds to its users optimally given the ask price or the predicted price of the auctions. This assignment method is also used to calculate the performance of the agents at the end of the game. Compared with the allocators used in the TAC (Section 4.2.4), this one is a linear assignment problem where the allocating problem in TAC is a NP problem. Thus, the solution we used here is much faster than that in TAC.

5.4 Flight Auction Scenario

This section provides an intuitive scenario[11] in which the operation of our algorithm can be exemplified and its performance empirically assessed (see section 5.5). Here we consider how to model the user's preferences as fuzzy sets, outline the environmental setting for realising the scenario, and present the training results for the FNN agent.

In more detail, there are a number of airlines selling flight tickets through auctions. Each auction is selling one flight ticket. Each software agent is acting on behalf of one user who has multiple requirements and they are informed of the user's preferences about prices and travel dates.[12] The aim of the agent is to obtain the goods that maximise the sum of its users' satisfaction.

5.4.1 Users' Preference Settings

We describe the valuation v of a user for a ticket as a trapezoid shape fuzzy number $(l_{bottom}, l_{top}, r_{top}, r_{bottom})$, where $l_{bottom} = 0$ and $l_{top} = 0$, and r_{top} and r_{bottom} are the values where the satisfaction starts to decrease and where it becomes 0. In this case, the higher the price of the good, the lower the satisfaction degree. When the price increases to the valuation of the agent, the satisfaction degree is 0. The travel date q is represented as a triangular fuzzy number[13] (l_q, c_q, r_q), where c_q is the preferred date and l_q and r_q are the

[11]We choose (for reasons of familiarity) a flight auction scenario where an agent is trying to buy multiple flight tickets on behalf of a user. This is a real problem that we often meet in real life and it builds upon the TAC scenario.

[12]For reasons of simplicity, we focus on the two attribute case. However, the principle is similar with more attributes.

[13]Any kind of fuzzy number can be used here, *e.g.*, trapezoid or bell-shaped fuzzy number. We choose a triangular fuzzy number simply because it is the most commonly used.

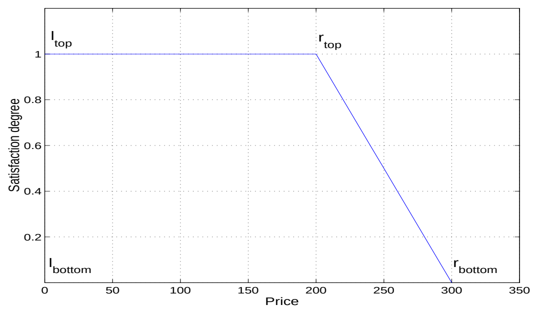


Figure 5.4: Customer's preference about price.

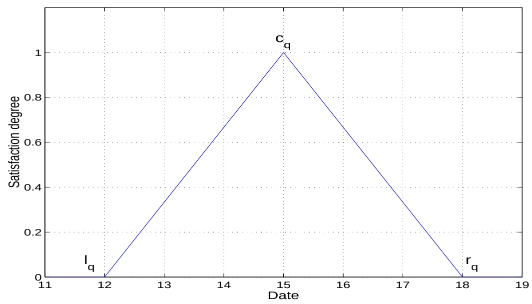


Figure 5.5: Customer's preference about travel date.

left and right limits respectively.[14] Using fuzzy numbers to express the preferences is more flexible than the fixed preference value as used in TAC (Section 4.1).

By way of illustration, suppose a customer's valuation about the ticket is about 300 pounds and she wants to travel on about the 15th of December. These preferences are expressed as fuzzy sets by the respective membership functions μ_P and μ_Q given in (5.25) and (5.26) and are shown graphically in Figures 5.4 and 5.5.

$$\mu_P(x) = \begin{cases} 1 & \text{if } x \leq 200, \\ \frac{300-x}{100} & \text{if } 200 < x < 300, \\ 0 & \text{if } x \geq 300. \end{cases} \tag{5.25}$$

$$\mu_Q(y) = \begin{cases} \frac{y-12}{3} & \text{if } 12 \leq y \leq 15, \\ \frac{18-y}{3} & \text{if } 15 \leq y \leq 18, \\ 0 & \text{if } y \leq 12 \text{ or } y \geq 18. \end{cases} \tag{5.26}$$

5.4.2 Experimental Settings

The experiments aim to cover a broad range of scenarios. All the parameters about the environment are assigned at the beginning of the game. Here we suppose that all auctions start at a price of 0 and all have a bid increment of 10 pounds. Also:

[14] If a user has a crisp preference, for example, he has to travel on the 15th of Dec, the similarity degree of 15th is 1 and 0 otherwise.

- A day in the game equals $\zeta = 20$ seconds of real time.[15]
- An auction i's starting time t_i^{start} is randomly chosen from a uniformly distributed range $(0, (q_i - 5)\zeta)$. This ensures all the auctions start at least five days before the travel date.[16]
- Auction i's end time is randomly chosen from a uniformly distributed range $(t_i^{start} + 2\zeta, q_i - 3\zeta)$. This guarantees that the auctions close at least three days before the travel date.[17]
- Each agent is randomly assigned to a customer which has n requirements, $n \in [1, 5]$;
- All the agents start bidding at the beginning of the game.[18]
- Auction i's flight date q_i is chosen randomly from a uniformly distributed range $(11, 19)$.
- The valuation of the goods for a customer are randomly chosen from a uniformly distributed range $(170, 370)$.
- A customer's preferred travel date is randomly chosen from a uniformly distributed range $(12, 18)$.[19]

5.4.3 *The FNN Agent's Learning Algorithm*

As discussed in section 5.3.2, the agent engages in a period of off-line learning in order to provide initial parameters for the FNN agent. In more detail, Figure 5.6 shows the curve of the root mean square error with respect to the number of training epochs.[20] After 200 training epochs, it can be seen that the error between the target output and the actual output reaches its lowest point and so the parameter settings of this point are those used when the agent is made operational. Specifically, Figures 5.7 and 5.8 show, respectively, the comparison of the FNN parameters before and after training. As can be seen, the parameters for each of the three inputs are adjusted from the original settings defined by the field experts.

5.4.4 *Parameter Adaptation in Different Environments*

This section compares the parameter adaptation in two environments where the supply is high (25 auctions) and low (15 auctions). Specifically, Figures 5.9 to 5.12 show how the parameters are adjusted differently in different environments. In Figure 5.10, when supply is high, the closing prices tend to be low and, thus, the consequent parameters

[15]The length of one day can be shorter if it can be guaranteed that all the agents have time to respond in the market.

[16]We choose five to ensure the agent has a reasonable time to transact the ticket before the travel date.

[17]We choose three to ensure the start time of the auction is before the end time and there is a reasonably long time for the auction.

[18]This is because we want to evaluate all types of agents fairly. If some agents start bidding late, they will be at a disadvantage compared with those that start early.

[19]This range is smaller than the range of the auctions' flight dates because this preferred travel date is a fuzzy number. Thus when defuzzified it will actually cover the full range of the flight's dates.

[20]The reference price and ask price are scaled by dividing by 10 during learning. However, this does not affect the result in any way. In the real world scenario, the price can be any number.

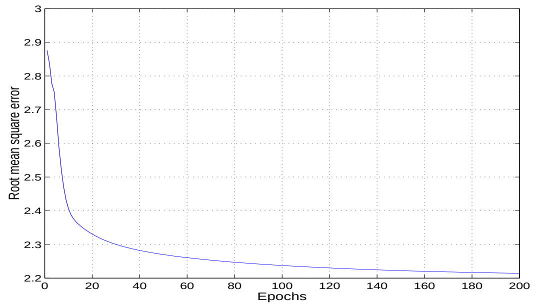


Figure 5.6: Learning curve: root mean square error versus time.

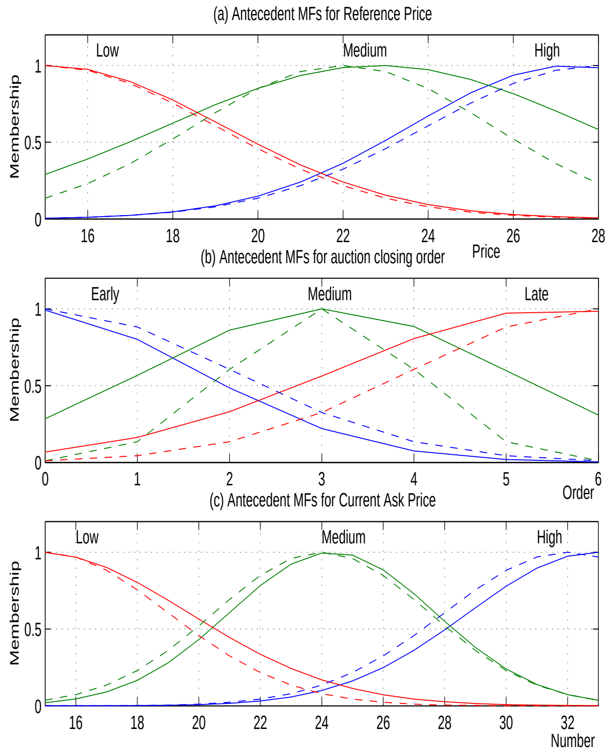


Figure 5.7: Comparing antecedent membership functions (MFs) before (dashed line) and after (solid line) off-line learning.

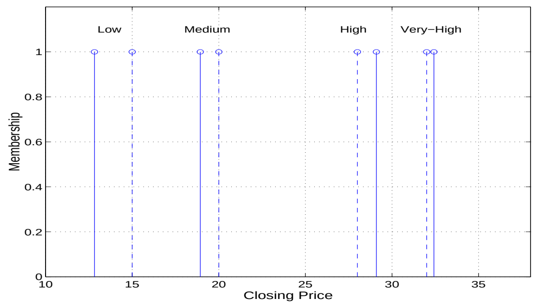


Figure 5.8: Comparing consequent membership functions before (dashed line) and after (solid line) off-line learning.

are lower than the initial ones. In contrast, in Figure 5.12, when supply is low, the closing prices tend to be high, and the consequent parameters are higher than the initial parameters.

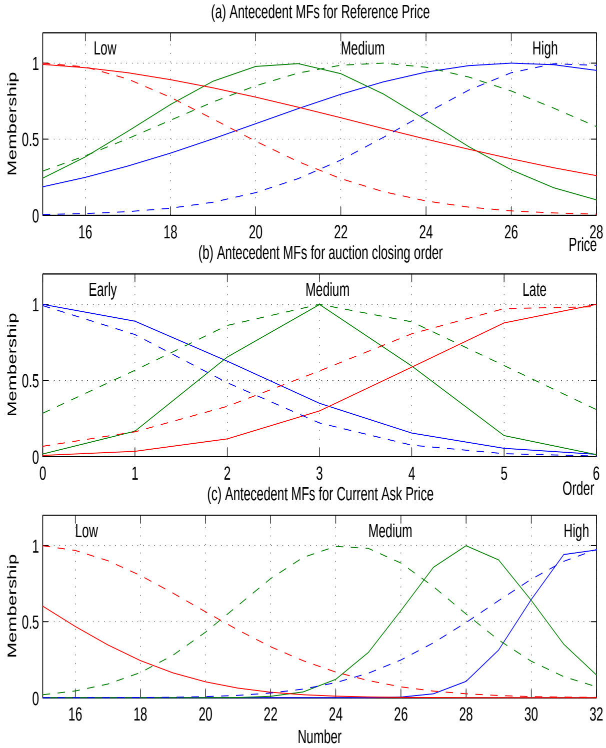


Figure 5.9: Comparing antecedent membership functions (MFs) before (dashed line) and after (solid line) off-line learning in high supply environment.

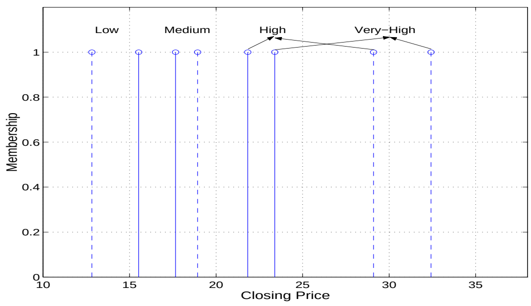


Figure 5.10: Comparing consequent membership functions before (dashed line) and after (solid line) off-line learning in high supply environment.

5.5 Empirical Evaluation

This section evaluates the FNN agent by comparing it in a variety of environments, with other agents that use bidding strategies proposed in the literature. In particular we are interested in assessing the performance of each kind of agent in different environments. There are three main groups of experiments and there are a number of sessions which

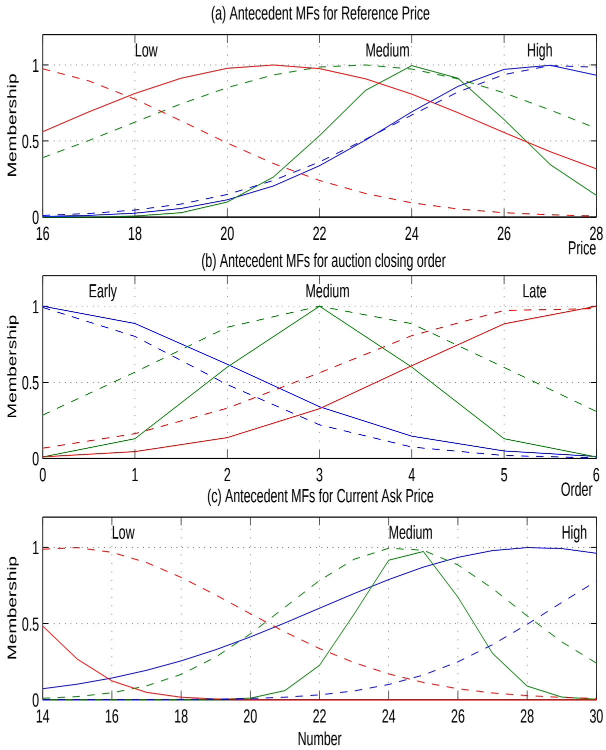


Figure 5.11: Comparing antecedent membership functions (MFs) before (dashed line) and after (solid line) off-line learning in low supply environment.

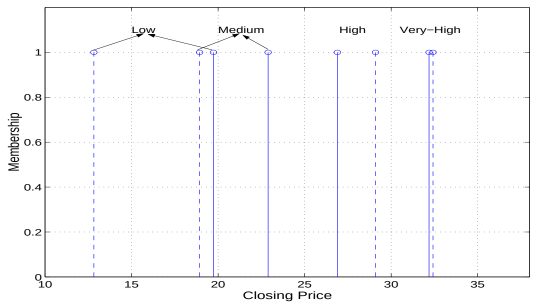


Figure 5.12: Comparing consequent membership functions before (dashed line) and after (solid line) off-line learning in low supply environment.

correspond to experiments with different settings (as per section 5.4.2). For each session, at least 200 games[21] are played among the agents.

Since the number of agents in each experiment varies, the performance ρ_K of a particular type K of agent (*e.g.*, FNN) is calculated as the average satisfaction degree per agent of the kind K, that is:

$$\rho_K = \frac{\sum_i^{n_K} \sum_j^{m_i} u_j^{(i)}}{n_K}, \tag{5.27}$$

[21] A t-test showed that 200 games are sufficient to give a significant ranking among the agents. A *p value* of $p < 0.05$ is reported for all the experiments.

where n_K is the number of type K agents in the same game, $u_j^{(i)}$ means the evaluation of customer j and m_i is the number of customers of agent i. Since most of the extant multi-auction bidding strategies are concerned solely with price (see section 5.6), we had to extend them to deal with bidding for goods that are characterised by multiple attributes. Thus, in all cases, the agents used the aggregation operators specified in section 5.3.3 in order to make trade-offs between price and travel date. To deal with multiple goods, the allocation function can decide which user bids in which auction. The specific benchmark strategies we used are:

- *Greedy* (GRD) strategy (adapted from [Byde, 2001a]): while the sum of the number of goods held and owned is less than what the agent needs, bid in auctions where the auction's current satisfaction has the highest evaluation (as defined in section 5.3.3);
- *Fixed Auction* (FIX) strategy (adapted from [Byde et al., 2002]): select at the beginning of the game the auctions in which bids will be placed and then only bid in these auctions. The auctions chosen here are those where the sum of the users' satisfaction is highest for date (at this time none of them have a value for price). The agent continues bidding in its selected auction until the price satisfaction degree equals zero in which case it will switch to another auction (until all those in the fixed set have been tried).
- *Average* (AVG) strategy: AVG also uses the ECF algorithm, but it uses a much simpler prediction function based on the past history transaction prices to predict the closing prices. In more detail, it calculates the average closing prices of all the auctions for each kind of good from the recent games. Suppose in the latest N games, there are m auctions with attribute i, then the predicted closing price of an auction with attribute i is:

$$\tilde{p}_{close}^{(i)} = \frac{\sum_j^m p_j^{(i)}}{m},$$

  where $p_j^{(i)}$ is the real closing price of auction j with attribute i.

5.5.1 Varying Agent Populations

This experiment aims to compare the performance of the different types of agents when there are varying numbers of the other agent types in the population (here the population size is fixed). In this experiment, we studied three environments: when supply is low (15 auctions), medium (20 auctions) and high (25 auctions). When the weighted average operator is used and the weight ratio is $w_p : w_q = 1 : 1$, Figure 5.13 shows the results when there are a fixed number of each type of agent in a session (a), and when one type dominates numerically (b) to (e).[22]

[22] In (a), there are equal numbers of each agent and we have 4 kinds of agents, thus there are 8 agents. In (b) to (e), there are 4 of one kind of agent and 2 of other three kinds. Thus, there are 10 agents in total.

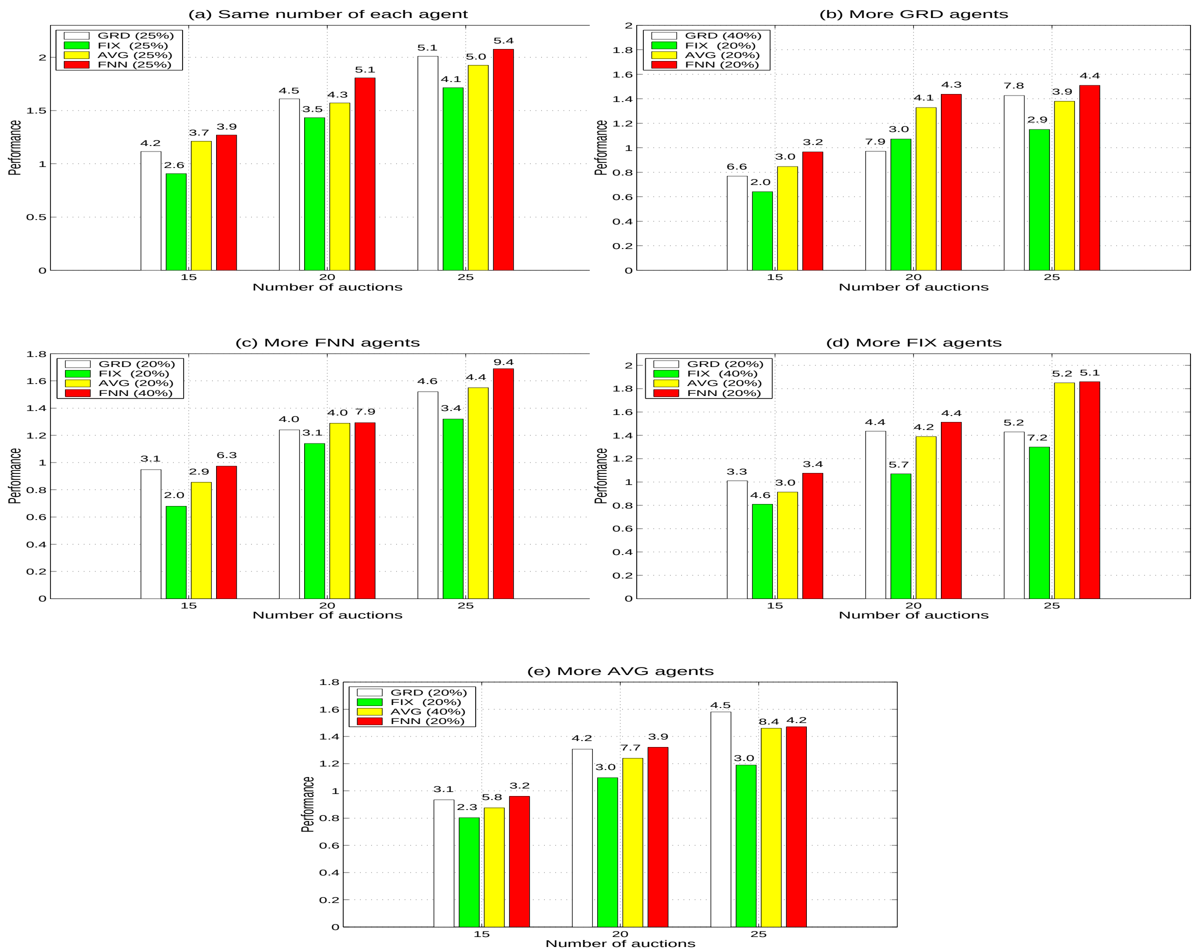


Figure 5.13: Performance of agent with various agent populations.
The horizontal axis shows the total auction number in the market. The vertical axis represents the performance of the kind of agents in that session. The number on top of each bar is the number of transactions made per agent by that kind of agents in the session. In (a) there are 8 agents in total and 10 for (b) to (e).

From this, it can be seen that the FNN agents perform better than AVG agents in most cases considered. We attribute this success to their ability to be able to select the auctions to bid in according to the relatively correct prediction on the closing prices of the auctions. The FNN agent is better than GRD agents in most cases except in (e) where there is a high supply and 40% AVG agents. This is because in this case there are many agents that use the ECF strategy. Thus, it is likely to be the case that a number of FNN agents and AVG agents are waiting for specific auctions to bid and some of them failed to obtain the goods. The GRD agent endeavours to make a transaction whenever it can. Its main shortcoming is that it only considers ongoing auctions (it ignores those that have not yet started and so fails to consider the full set of potential purchasing opportunities when making bidding decisions). Thus, it sometimes buys a good at the user's valuation price, when, if it waited, it may well find subsequent auctions with lower closing prices.

The FIX agent performs worst because it only bids in auctions where it knows, a priori, that it can get high satisfaction on the flight date. This leads to a poor overall performance because it misses auctions that have a high evaluation on price but a lower one on date. As is shown in the figure, FIX agents also have the smallest transaction numbers, because they only bid in small number of auctions. This shows the advantage of our heuristic of wanting to bid in more than just the *n* most promising ones.

It can also be seen from all the subfigures in Figure 5.13 that when the supply is high, all the agents have a higher performance value than when there is a low supply. This is because, in general, the auctions close at a lower price when the supply is high (because there is less competition). In Figure 5.13(b), GRD agents dominate the market. This is because these agents are greedy which means they often make the transaction price of some auctions very high because they will top up the ask price whenever it is lower than their reserve prices. It can be seen from the relative performance in Figure 5.13(b) that GRD agents do indeed have the lowest performance value in this case. When FIX agents dominate the market (see Figure 5.13(d)), all the other agents have a higher performance value compared to other cases in Figure 5.13. This is because the FIX agents only bid in a small number of auctions. Thus other auctions have less competition and, consequently, lower prices.

From Figure 5.13, we can also see the number of transactions made by each kind of agent. Generally, the greater the number of agents, the larger the number of transactions made (*e.g.*, more GRD in (b), more FNN in (c), more FIX in (d), and more AVG in (e)). When the agent numbers are the same and when the performance of one kind of agent is better than the others, but the number of transactions is smaller, this means the agent made some good deals (*e.g.*, it had some transactions at low prices). For example, in (a) when there are only 15 auctions, the FNN agent had the highest performance, but the average number of transaction is only 3.9 which is smaller than the corresponding value for the GRD agents (4.2).

5.5.2 Varying Aggregation Operators

This experiment studies the impact on the different types of agents of the different ways of trading-off the price and travel date (see Figure 5.14). To do this, the number of auctions is generated randomly in the range of $[15, 25]$ and the number of agents is set to 8. We also fix the weight of $w_p : w_q = 1 : 1$ for each operator (see section 5.5.3 for experiments with differently weighted attributes). This time, the numbers of each agent type in a given game are randomly generated.

As can be seen, the FNN agents behave the best and AVG agents behave second in all cases above. In fact, the order of performance of the four kinds of agent does not change for different aggregators. This shows that our FNN agent performs best whatever

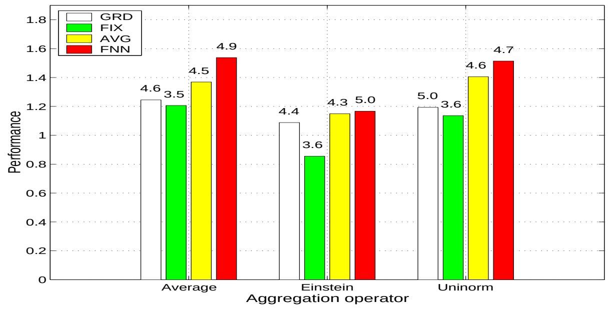


Figure 5.14: Performance of various agents using different aggregation operators.

aggregation operator is used. Again, we attribute this success of FNN agents to the efficiency of the ECF algorithm and its ability to predict the auction closing price and the parameter adaptation through learning. It is also shown that the performance of agents using the Einstein operator is, relatively speaking, lower than that of using average and uninorm operators. This is because the Einstein operator uses T-norm operators [Yager and Filev, 1994] which means the overall evaluation is less than the value of each of the constituent attributes.

5.5.3 Varying Preference Weights

This experiment evaluates the performance of the different agents when they use varying weights for the different attributes of the goods. This is an important issue to consider because, again, various weightings may lead to a different ranking of the strategies. Thus, for each operator, we conducted experiments to test the impact of the weights on the performance of various strategies. In this case, the number of auctions is generated randomly from $[15,25]$ and Figure 5.15 shows the performance of the agents with the three representative weights we consider. As can be seen, the order of each kind of agent does not change for the different weight ratios. Again, in all cases, FNN agents perform the best followed by AVG agents. This superior performance is again due to the efficiency of ECF algorithm.

However, in Figures 5.15(a) and (b), we see that the GRD agents also perform well when $w_p : w_q = 1 : 3$. This is because such agents always choose the currently best candidate and the one with a high satisfaction on “date” is usually chosen. Note that GRD agents accept any price within their budget line, thus the satisfaction degree on price is not always high. Thus, when “date” is valued more, GRD agents tend to perform well. However, this advantage disappears when price is valued more. This feature is less apparent when we use the uninorm operator (see Figure 5.15(c)), since even when the satisfaction for one attribute is high, the overall evaluation is not always high. For FNN and AVG agents, when the attribute of “price” is valued more, the agents perform better and have a larger number of transactions. Again, this shows that the ECF algorithm enables the agents to make more transactions. The FIX agents also perform better when

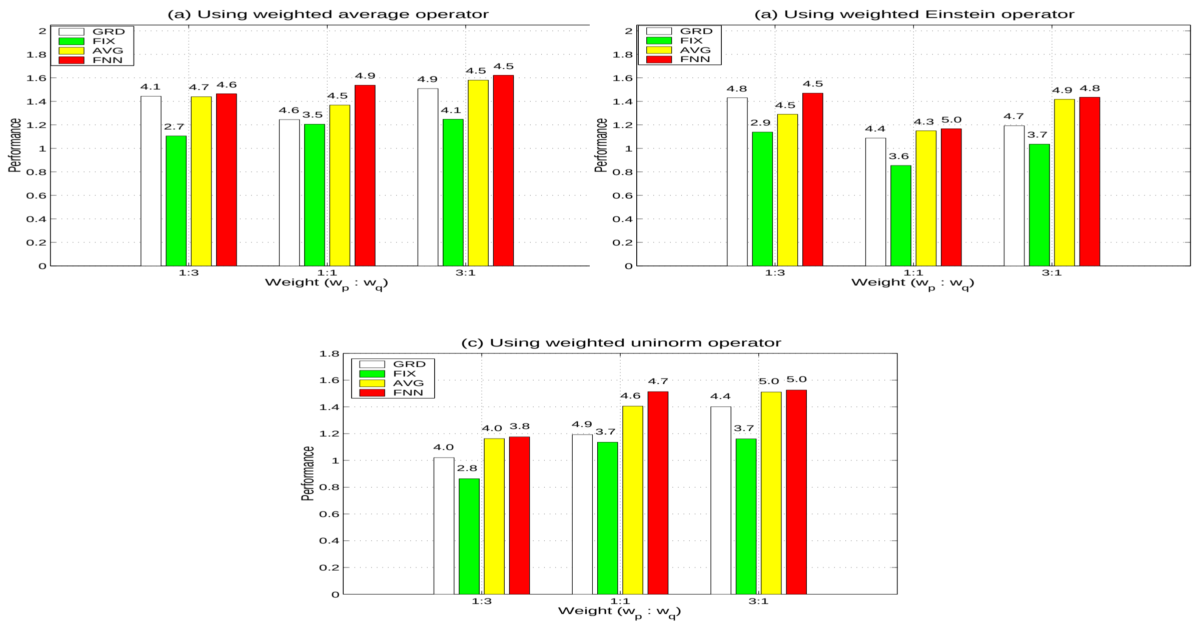


Figure 5.15: Using weighted operators with varying weights.

"price" is valued more, because the behaviour of the FNN and AVG agents make the transaction price low in this situation.

5.5.4 *Varying Attitude to Risk*

This experiment aims to study the influence of the risk attitude on the performance of the FNN agent given the same aggregation operators (because for the same operators, different thresholds for the risk attitudes may lead to different performance characteristics for the agents). In particular, we aim to find the best configuration of risk attitude for an agent to adopt. Thus, for each kind of aggregation operator, three risk attitudes are compared: risk-seeking, risk-neutral and risk-averse. For example, when the weighted average operator is used, the threshold (λ in Figure 5.1) is 0.1 for a risk-seeking agent, 0.2 for a risk-neutral agent and 0.3 for a risk-averse agent.

From Figure 5.16, we can see that in general terms the higher the supply there is, the better the agent performs (as was shown in the experiment in section 5.5.1). However, the general trend is that risk-seeking agents perform better when supply is high and risk-averse agents behave better when supply is low. This is because when supply is high, there is little competition among the agents and a risk-seeking agent which has a big threshold can almost always win in the auction it selects. This, in turn, leads to high profit per transaction. In contrast, when supply is low, the game is very competitive and a risk-averse agent with a small threshold can win a good whenever it is within the threshold. However it cannot make much profit per transaction.

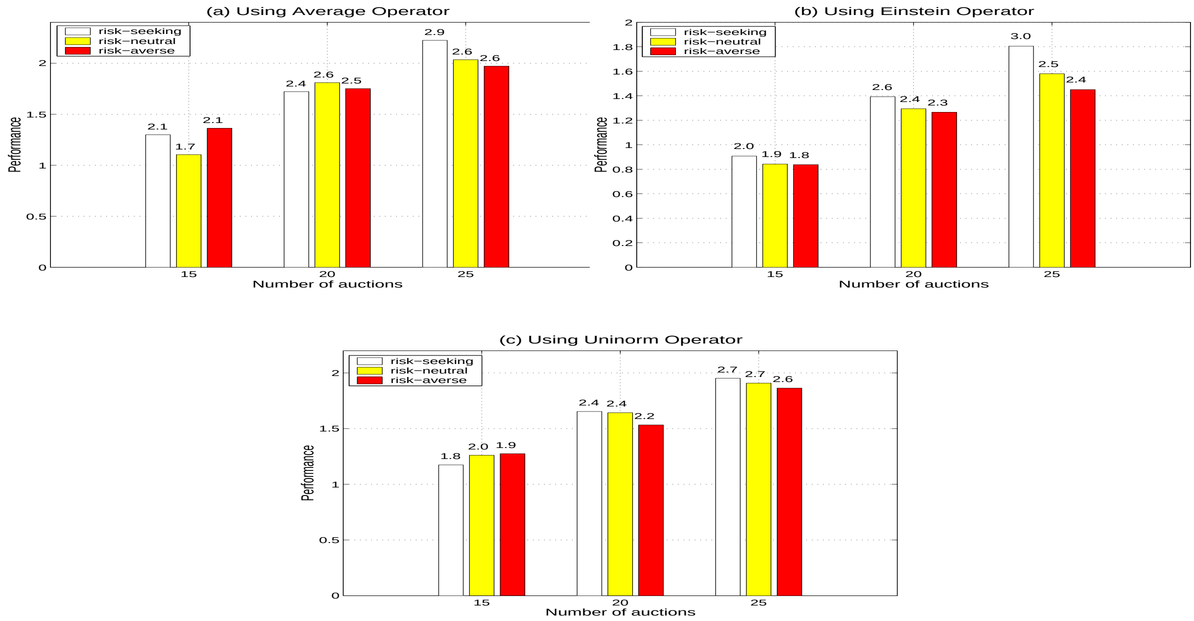


Figure 5.16: Performance of FNN agents with different risk attitudes.

5.6 Related Work

There are several strands of work that are directly related to what we have described in this chapter. There is work on agents bidding in multiple overlapping auctions (which has already been discussed in Section 4.4) and work on using fuzzy techniques to manage agent interactions (which has also been discussed previously (in Section 3.5). The related work which has not yet been considered is that on multi-attribute auctions [Che, 1993, Bichler et al., 1999, Jennings et al., 2000a, David et al., 2002] (Section 2.3.3). However, in these works, the buyer is the auctioneer who calls for bids along multiple attributes from sellers, while in our work the sellers are the auctioneers. There are multiple auctions in our context where they have one auction. Our agent bids on price only although it needs to consider multiple attributes while the agents in these works needs bid on multiple attributes.

5.7 Summary

This chapter developed a new algorithm that guides an agent's bidding behaviour in multiple overlapping English auctions for multiple items characterised by multiple attributes. The Earliest Closest First algorithm we developed first calculates the auctions that best fit the users' preferences and then bids in order of increasing end time in any auctions that have a satisfaction degree that is reasonably close to the best ones. Specifically, the FNN strategy uses neuro-fuzzy techniques to predict the expected closing prices of the English auctions and to determine which auction the agent should bid in at what time. The use of a fuzzy neural network also allows the decision making criteria of our agent to be adapted to the situation in which it finds itself. The adaptation is based on the learning of the neural network where the parameters in the fuzzy sets and consequent output can be adjusted. Moreover, we benchmarked our algorithm against two common alternatives available in the literature and the strategy which also uses ECF but with a different prediction function. In most cases we considered, the FNN strategy is superior to the others. This shows the effectiveness of the ECF and the adaptation ability of the FNN agent. Our algorithm can also make trade-offs in its bidding behaviour between the different attributes that characterise the desired good in order to maximise the user's satisfaction.

Chapter 6

Conclusions

With the increasing automation of e-commerce, we believe ever greater amounts of trading will be conducted in on-line auctions by software agents. However, to make progress in this area one of the key problems that needs to be addressed is that of developing effective and efficient bidding strategies that agents can use to achieve their negotiation objectives. This is exactly the aim of this work and, to this end, we have developed such strategies for continuous double auctions (Chapter 3), for multiple inter-related auctions with multiple protocols (Chapter 4), and for a general multiple English auction market (Chapter 5). In all cases, these strategies have been benchmarked against the leading other strategies that have been proposed in the literature and our strategies have been shown to be effective in a wide variety of circumstances.

In more detail, we first developed a strategy that guides an agent's buying and selling behaviour in a continuous double auction. Our strategy, the FL-strategy, uses heuristic fuzzy rules and a fuzzy reasoning mechanism to decide what bids or asks to place. We then enhanced the basic strategy so that it can adapt its behaviour to the supply (demand) in the market (this revised strategy is called the adaptive FL-strategy). Our strategies were then shown to outperform the main strategies that have previously been proposed for CDAs. This result is especially promising since the benchmark strategies we evaluate against have been shown to outperform human bidders in experimental settings [Das et al., 2001].

Based on the success of our work in CDAs, we then used fuzzy logic techniques in the multiple auctions context. In particular, we developed a trading agent, SouthamptonTAC, that participated in the International Trading Agent Competition in 2001 and 2002. This agent uses several of the techniques devised in the CDA work and can adapt its bidding behaviour according to its assessment of situation in which it finds itself. SouthamptonTAC has been shown to be successful across a wide range of TAC environments in both competitions, as well as in our controlled experiments. In more detail, SouthamptonTAC does best, obtains the highest score, in competitive games (*i.e.*, where the number of risk seeking agents is large). It also performs well in non-competitive environments because it can adapt its strategy to bid aggressively for the goods it wants.

To build a more general multiple auction model and improve the preference representation of the customer, we focused on the problem of an agent bidding across multiple, simultaneous English auctions. For this case, an Earliest Closest First algorithm was proposed that bids in the auctions which have a close satisfaction degree with what are believed to be the optimal set. To realise this algorithm, we designed an agent that uses neuro-fuzzy techniques to predict the expected closing prices of the auctions. The parameters involved in the strategy can be adapted according to standard learning algorithms in neural networks. Thus, our agent is able to adapt its bidding strategy to reflect the type of environment in which it is situated. As before, we compared our agent with a range of other strategies from the literature (in a flight auction scenario). The result shows that, in most cases, the FNN agents outperform other strategies.

Looking back at the research aims outlined in Section 1.2, the research objectives that were laid out have been met:

- We successfully used fuzzy reasoning methods (Sections 4.2.7 and 5.3.1) to predict the auctions' likely closing prices in a dynamic market. The effectiveness of both agents show that the prediction is sufficiently accurate to make reasonable decisions.
- All the agents' strategies were adaptive to some degree. The CDA agent (Section 3.4) adapts its bidding behaviour according to the transaction frequency, SouthamptonTAC (Section 4.2.8) adapts its behaviour between three different kinds of TAC environments, and the FNN agent (Section 5.3.2), which exhibits the greatest degree of adaptivity, can vary the parameters involved in the bidding strategy to reflect the environment in which it is situated.
- The agents bid flexibly and relax their constraints where appropriate. Both the CDA agent and SouthamptonTAC realised this in the bidding in a continuous double auction (Sections 3.2.2 and 4.2.6). Flexible bidding was shown in the earliest closest first strategy of the FNN Agent (Section 5.2) by the fact that it chooses the closest auctions to the optimal set to bid in.
- The attitudes towards risk are varied in all the agents (Sections 3.3.2, 4.2.8 and 5.2) and we found it had a significant beneficial impact on their performance.
- Both SouthamptonTAC (Section 4.2.5) and the FNN agent (Section 5.3.3) were able to make trade-offs when bidding in auctions.

In addition to developing the strategies themselves, we also believe that our work is significant both for the areas of agent-mediated e-commerce and fuzzy logic. In the former case, we developed novel bidding strategies for a number of auction contexts. In the latter case, we showed how fuzzy logic can be employed in agent-mediated e-commerce settings. Specifically, fuzzy logic theory and fuzzy neuro-network techniques are shown to be suitable ways to design a strategy for a bidding agent. Thus we successfully used fuzzy reasoning to predict the likely closing price of an auction; adaptive bidding behaviour is achieved through neuro-fuzzy techniques, where the parameters in the membership function of the fuzzy sets are adapted through error propagation; and flexible bidding enables the agent to relax constraints when bidding in order to make more transactions.

When taken together, these contributions make an important step towards realising the full potential of agent mediated e-commerce. Specifically, we have extended the state-of-the-art in strategies for continuous double auctions, developed robust bidding strategies for the trading agent competition, and developed and implemented, for the first time, algorithms that can purchase multiple goods from multiple English auctions.

Undertaking this work, we identified a number of commonalities between the methods of approach in the various contexts. These steps, outlined below, can be viewed as an embryonic version of a methodology for building trading agents using soft computing techniques.

1. *Determine what to reason about in the auction setting.* The chosen issues are usually the key determinant of what to bid in the auction. For example, it is the predicted closing price in the TAC (Chapter 4) and the bid price in the CDA (Chapter 3). These issues vary in different auctions and they are typically difficult to predict because of the uncertainty of the game.
2. *Choose the factors that should be used in the fuzzy rules.* If there is no history data available, these factors need to be chosen by the experience and intuition of the designers. If there is history data, the most relevant factors can be chosen by analysing the data set.
3. *Structure the fuzzy rules.* This is the key part of the fuzzy reasoning. There are two ways to sort out the rules. First, if there is a lot of history data, unsupervised learning can be used to abstract of the rules automatically [Lin, 1994]. The second way is to design the rules according to the relationship of the factors to the reasoning value. This, again, requires the designer to have rich experience and intuition in the environment. In some cases, if the problem is too complicated, more than one rule base can be used to deal with different situations (see the fuzzy rules in Chapter 4). In either case, however, the designer needs to decide which controller to use. In many cases, Sugeno and Mamdani [Mamdani, 1974] are commonly used: if the output of each rule clearly corresponds to some constants, the Sugeno controller is used; otherwise if the output is described as fuzzy sets, the Mamdani controller is more appropriate.
4. *Decide how to adapt the parameters in the fuzzy rules.* The initial parameters in

the fuzzy rules are usually set by the domain experts. In the case where neuro-fuzzy techniques are used, the adaptation can easily be achieved through supervised learning [Rumelhart et al., 1986]. However, for the fuzzy reasoning itself, the adaptation is more complicated because it often involves the adaptation of the parameters in the output of the fuzzy rules. These parameters can be adapted according to the risk attitudes of the agent, whether the auction is competitive or not, or whether it is urgent to obtain the goods.

Despite these achievements, more work is needed. In particular, there are six promising directions for further research based on this thesis:

- In both the CDA and TAC, we used the Sugeno controller, but it may be the case that other fuzzy logic controllers, such as the conventional Mamdani controller [Mamdani, 1974, Zimmermann, 1996], could also be used. This is a challenge since we need to determine if our fuzzy reasoning methods are suitable for controllers besides the Sugeno and it may be the case that another type of controller could improve the performance of our algorithms still further.
- In TAC, the pattern recognition procedure used to classify the degree of competitiveness of the environment could be enhanced by incorporating into the decision process further variables that are indicative of the trading evolution. Relevant examples include the prices of the hotel auctions in TAC the rate of change of the increase of the prices and the bidding behaviour of other agents. We believe that by incorporating such additional factors we may be able to develop a more robust knowledge base because this will lead to a more accurate classification result.
- For the FNN agent, we only consider the case where the multiple goods are independent. However, it is increasingly the case that people will buy multiple interdependent goods (as per the TAC). To deal with this case, the fuzzy rule base will need to be expanded to deal with the relation among attributes. Moreover, other related factors which are significant to the auction closing prices need to be considered because there are constraints among the goods.
- The FNN structure of the FNN agent (*e.g.*, the size of the linguistic terms and the fuzzy rules) is currently designed by domain experts. Ideally, however, this structure should be obtained by self-organised learning [Lin, 1994]. This is an important extension because it means that our method could more easily be used because it would be free from human involvement.
- The basic ideas of the ECF algorithm can be extended to other auction protocols (*e.g.*, Continuous Double Auctions or Dutch auctions). To achieve this, the optimal asks and bids can be calculated first and then, according to the current ask or bids, the agent can decide whether to accept the ask/bid or submit a fuzzified ask/bid.
- Based on our work on multiple auctions, it is a challenge to further generalise the neuro-fuzzy based prediction method so that it can be used in other auction

types (including Vickrey, Dutch and First-Price Sealed Bid auctions). In this case, the decision problem is more complicated since each auction protocol has its own price updates rules and it will involve more fuzzy sets and fuzzy rules.

Bibliography

[Adams et al., 1996] Adams, G., Rausser, G., and Simon, L. (1996). Modelling multilateral negotiations: an application to california water policy. *Journal of Economic behavior and organization*, 30(1):97–111.

[Andersson and Sandholm, 1998] Andersson, M. and Sandholm, T. (1998). Leveled commitment contracting among myopic individually rational agents. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 26–33.

[Anthony and Jennings, 2002] Anthony, P. and Jennings, N. R. (2002). Evolving bidding strategies for multiple auctions. In van Harmelen, F., editor, *Proceedings of the 15th European conference on Artificial Intelligence*, pages 178–182, Amsterdam. IOS Press.

[Anthony and Jennings, 2003] Anthony, P. and Jennings, N. R. (2003). Developing a bidding agent for multiple heterogeneous auctions. *ACM Trans. on Internet Technology*, 3(3):185–217.

[Badea, 2000] Badea, L. (2000). Learning trading rules with inductive logic programming. In de Mantaras, R. and Plaza, E., editors, *Machine Learning: ECML 2000*, volume 1810 of *Lecture notes in artificial intelligence*, pages 39–46.

[Bandemer and Gottwald, 1995] Bandemer, H. and Gottwald, S. (1995). *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods with Applications*. Wiley.

[Bapna et al., 2001] Bapna, R., Goes, P., and Gupta, A. (2001). Insights and analyses of online auctions. *Communications of the ACM*, 44(11):42–50.

[Barnes-Vieyra and Claycomb, 2001] Barnes-Vieyra, P. and Claycomb, C. (2001). Business-to-business e-commerce: models and managerial decisions. *Business horizons*, pages 13–20.

[Beam and Segev, 1996] Beam, C. and Segev, A. (1996). Automated negotiation: A survey of the state of the art. CITM working Paper 96-WP-1022.

[Bichler, 2000] Bichler, M. (2000). An experimental analysis of multi-attribute auctions. *Decision Support Systems*, 29:249–268.

[Bichler and Kaukal, 1999] Bichler, M. and Kaukal, M. (1999). Design and implementation of a brokerage service for electronic procurement. In *Proceedings of the Tenth International Workshop on Database and Expert Systems Applications*, pages 618–622.

[Bichler et al., 1999] Bichler, M., Kaukal, M., and Segev, A. (1999). Multi-attribute auctions for electronic procurement. In *Proceedings of the first IBM IAC workshop on Internet based negotiation technologies*, Yorktown Heights, NY.

[Bichler and Segev, 1999] Bichler, M. and Segev, A. (1999). A brokerage framework for Internet commerce. *Distributed and Parallel Databases*, 7:133–148.

[Billsus and Pazzani, 1998] Billsus, D. and Pazzani, M. (1998). Learning collaborative information filters. In *Proceedings of the fifteenth international conference on machine learning*, pages 46–54.

[Binmore, 1992] Binmore, K. (1992). *Fun and games: a text on game theory*. Lexington: D.C. Heath.

[Bonissone and Decker, 1986] Bonissone, P. and Decker, K. (1986). Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. In Kanal, L. N. and Lemmer, J. F., editors, *Uncertainty in Artificial Intelligence*, pages 217–247. Elsevier Science Publishers B.V.

[Brenner et al., 1998] Brenner, W., Zarnekow, R., and Witting, H. (1998). *Intelligent Software Agents*. Springer.

[Brewer, 1999] Brewer, P. (1999). Decentralized computation procurement and computational robustness in a smart market. *Economic theory*, 13:41–92.

[Byde, 2001a] Byde, A. (2001a). A comparison among bidding algorithms for multiple auctions. Technical report, HP Research Labs.

[Byde, 2001b] Byde, A. (2001b). An optimal dynamic programming model for algorithm design in simultaneous auctions. Technical Report HPL-2001-67, Hewlett Packard Research Labs, Bristol, UK.

[Byde et al., 2002] Byde, A., Preist, C., and Jennings, N. R. (2002). Decision procedures for multiple auctions. In *Proceedings of the first International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 613–620.

[Caplice, 1996] Caplice, C. (1996). *An optimization based bidding process: a new framework for shipper-carrier relationships*. PhD thesis, School of engineering, MIT.

[Cason and Friedman, 1991] Cason, T. and Friedman, D. (1991). An empirical analysis of price formation in double auction markets. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 253–283. Addison-Wesley.

[Chaib-draa, 1995] Chaib-draa, B. (1995). Industrial applications of distributed artificial intelligence. *Communications of the ACM*, 38(11):47–53.

[Che, 1993] Che, Y.-K. (1993). Design competition through multidimensional auctions. *RAND Journal of Economics*, 24(4):668–680.

[Cliff, 2001] Cliff, D. (2001). Evolutionary optimization of parameter sets for adaptive software-agent traders in continuous double auction markets. Technical report, Hewlett Packard Research Labs, Bristol, England.

[Collins et al., 1998] Collins, J., Youngdahl, B., Jamison, S., et al. (1998). A market architecture for multi-agent contracting. In Sycara, K. and Wooldridge, M., editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 285–292.

[Dang and Jennings, 2002] Dang, V. and Jennings, N. R. (2002). Polynomial algorithms for clearing multi-unit single item and multi-unit combinatorial reverse auctions. In van Harmelen, F., editor, *Proceedings of the 15th European conference on Artificial Intelligence*, pages 23–27, Amsterdam. IOS Press.

[Das et al., 2001] Das, R., Hanson, J., Kephart, J., and Tesauro, G. (2001). Agent-human interactions in the continuous double auction. In Nebel, B., editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pages 1169–1176, Washington.

[Dastani et al., 2001] Dastani, M., Jacobs, N., et al. (2001). Modeling user preferences and mediating agents in electronic commerce. In Dignum, F. and Sierra, C., editors, *Agent Mediated Electronic Commerce*, volume 1991 of *Lecture Notes in Computer Science*, pages 163–193. Springer.

[David et al., 2002] David, E., Azoulay-Schwartz, R., and Kraus, S. (2002). An english auction protocol for multi-attributes items. In Shehory, O. and Walsh, W., editors, *Agent Mediated Electronic Commerce*, volume 2531 of *Lecture Notes in Artificial Intelligence*, pages 52–68. Springer.

[Dou and Chou, 2002] Dou, W. and Chou, D. (2002). A structural analysis of business-to-business digital markets. *Industrial marketing management*, 31(2):165–176.

[Dubois and Prade, 1978] Dubois, D. and Prade, H. (1978). Operations on fuzzy numbers. *Int. J. Systems Sci*, 9:613–626.

[Easley and Ledyard, 1993] Easley, D. and Ledyard, J. (1993). Theories of price formation and exchange in double oral auctions. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 63–97. Addison-Wesley.

[Easwaran and Pitt, 2000] Easwaran, A. and Pitt, J. (2000). An agent service brokering algorithm for winner determination in combinatorial auctions. In Horn, W., editor, *Proceedings of the Fourteenth European Conference of Artificial Intelligence*, pages 286–290.

[Faratin, 2000] Faratin, P. (2000). *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary College.

[Faratin et al., 1998] Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous System*, 24(3-4):159–182.

[Faratin et al., 2002] Faratin, P., Sierra, C., and Jennings, N. R. (2002). Using similarity criteria to make trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237.

[Fisher and Ury, 1981] Fisher, R. and Ury, W. (1981). *Getting to Yes: Negotiating an agreement without giving in*. Random House.

[Fletcher and Reeves, 1964] Fletcher, R. and Reeves, C. (1964). Function minimization by conjugate gradients. *The Computer Journal*, 7.

[Foss, 1998] Foss, J. (1998). Brokering the info-underworld. *British Telecommunications Engineering*, 17(2):202–206.

[Fox et al., 2000] Fox, M., Barbuceanu, M., and Teigen, R. (2000). Agent-oriented supply-chain management. *The International journal of flexible manufacturing*, 12:165–188.

[Fraichard and Garnier, 2001] Fraichard, T. and Garnier, P. (2001). Fuzzy control to drive car-like vehicles. *Robotics and Autonomous Systems*, 34(1):1–22.

[Friedman, 1993] Friedman, D. (1993). The double auction market institution: A survey. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 3–25. Addison-Wesley, Reading, MA, USA.

[Friedman and Rust, 1992] Friedman, D. and Rust, J. (1992). *The Double Auction Market: Institutions, Theories and Evidence*. Addison-Wesley.

[Fritschi and Dorer, 2002] Fritschi, C. and Dorer, K. (2002). Agent-oriented software engineering for successful TAC participation. In *Proc. 1st Int. Joint Conf. on Autonomous Agents and Multi-agent Systems*, pages 45–46, Bologna.

[Fujishima et al., 1999] Fujishima, Y., Leyton-Brown, K., and Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553.

[Gallego et al., 1998] Gallego, I., Delgado, J., and Acebron, J. (1998). Distributed models for brokerage on electronic commerce. In Lamersdorf, W. and Merz, M., editors, *TREC'98*, volume 1402 of *lecture notes in computer science*, pages 129–140. Springer-Verlag.

[Gamvroulas et al., 2000] Gamvroulas, S., Polemi, D., and Anagnostou, M. (2000). A secure brokerage network for retail banking services. *Future generation computer systems*, 16:423–430.

[Garfinkel et al., 2001] Garfinkel, S., Spafford, G., and Russell, D. (2001). *Web security, privacy and commerce*. O'Reilly.

[Giménez-Funes et al., 1998] Giménez-Funes, E., Godo, L., Rodríguez-Aguilar, J., et al. (1998). Designing bidding strategies for trading agents in electronic auctions. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 136–143.

[Gjerstad and Dickhaut, 1998] Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auction. *Games and Economic Behavior*, 22:1–29.

[Gleizes et al., 1999] Gleizes, M. et al. (1999). Self-organization and learning in multi-agent based brokerage services. In et al., H. Z., editor, *Intelligence in Services & Networks*, volume 1597 of *Lecture Notes in Computer Science*, pages 41–54.

[Gmytrasiewicz and Durfee, 1995] Gmytrasiewicz, P. and Durfee, E. (1995). A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 125–132.

[Gode and Sunder, 1993a] Gode, D. and Sunder, S. (1993a). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137.

[Gode and Sunder, 1993b] Gode, D. and Sunder, S. (1993b). Lower bounds for efficiency of surplus extraction in double auctions. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 199–219. Addison-Wesley.

[Goldman et al., 1995] Goldman, S., Nagel, R., and Preiss, K. (1995). *Agile competitors and virtual organizations: strategies for enriching the customer*. Van Nostrand Reinhold, New York.

[Greenwald, 2002] Greenwald, A. (2002). The 2002 trading agent competition: an overview of agent designs. Technical report, Brown University.

[Greenwald and Boyan, 2001] Greenwald, A. and Boyan, J. (2001). Bidding algorithms for simultaneous auctions: a case study. In *Proceedings of the third ACM conference on Electronic Commerce*, pages 115–124.

[Guttman and Maes, 1998] Guttman, R. and Maes, P. (1998). Agent-mediated integrative negotiation for retail electronic commerce. In Noriega, P. and Sierra, C., editors, *Agent Mediated Electronic Commerce*, volume 1571 of *Lecture Notes in Artificial Intelligence*, pages 70–90.

[Guttman et al., 1998] Guttman, R., Moukas, A., and Maes, P. (1998). Agent-mediated electronic commerce: A survey. *The Knowledge Engineering Review*, 13(2):147–159.

[Ha and Park, 2001] Ha, S. and Park, S. (2001). Matching buyers and suppliers: an intelligent dynamic-exchange model. *Intelligent e-business*, pages 28–40.

[Hake, 1999] Hake, R., editor (1999). *E-commerce, A World of Opportunity*. The British Computer Society, Swindon, Wiltshire, UK.

[He and Jennings, 2002] He, M. and Jennings, N. R. (2002). SouthamptonTAC: designing a successful trading agent. In van Harmelen, F., editor, *Proceedings of the 15th European conference on Artificial Intelligence*, pages 8–12, Amsterdam. IOS Press.

[He and Jennings, 2003] He, M. and Jennings, N. R. (2003). SouthamptonTAC: an adaptive autonomous trading agent. *ACM Transactions on Internet Technology*, 3(3):218–235.

[He and Jennings, 2004] He, M. and Jennings, N. R. (2004). Designing a successful trading agent: A fuzzy set approach. *IEEE Transactions on Fuzzy Systems*, 12(3).

[He et al., 2004] He, M., Jennings, N. R., and A.Prügel-Bennett (2004). A neuro-fuzzy bidding strategy for buying multiple goods in multiple english auctions. *IEEE Transactions on Fuzzy Systems*. submitted.

[He et al., 2003] He, M., Jennings, N. R., and Leung, H. F. (2003). On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003.

[Huhns et al., 2002] Huhns, M., Stephens, L., and Ivezic, N. (2002). Automating supply-chain management. In *Proceedings of the first International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1017–1024, Italy.

[Jang, 1993] Jang, J. (1993). ANFIS: adaptive-network-based fuzzy inference systems. *IEEE Transactions on System, Man and Cybernetics*, 23(3):665–685.

[Jennings, 2000] Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296.

[Jennings, 2001] Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of ACM*, 44(4):35–41.

[Jennings et al., 2001] Jennings, N. R., Faratin, P., Lomuscio, A., et al. (2001). Automated negotiation: Prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 10(2):199–215.

[Jennings et al., 2000a] Jennings, N. R., Norman, T., Faratin, P., et al. (2000a). Autonomous agents for business process management. *Applied Artificial Intelligence*, 14(2):145–189.

[Jennings et al., 2000b] Jennings, N. R., Woghiren, K., and Osborn, S. (2000b). Interacting agents — the way forward for agent-mediated electronic commerce. Technical report. Lost Wax white paper, available at http://www.lostwax.com/lostwax1/opinions/opinions1.html.

[Jennings and Wooldridge, 1998] Jennings, N. R. and Wooldridge, M. (1998). Applying agent technology. In Jennings, N. R. and Wooldridge, M., editors, *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag, Berlin, Germany.

[Johansson et al., 1990] Johansson, E., Dowla, F., and Goodman, D. (1990). *Back-propagation learning for multi-layer feed forward neural networks using the conjugate gradient method*. Lawrence Livermore National Laboratory, CA.

[Jonker and Volgenant, 1987] Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340.

[Jung and Jo, 2000] Jung, J. and Jo, G. (2000). Brokerage between buyer and seller agents using constraint satisfaction problem models. *Decision Support Systems*, 28:293–304.

[Kagel and Vogt, 1991] Kagel, J. and Vogt, W. (1991). Buyer's bid double auctions: Preliminary experimental results. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 285–305. Addison-Wesley.

[Kalakota and Whinston, 1996] Kalakota, R. and Whinston, A., editors (1996). *Frontiers of Electronic Commerce*, chapter 16, pages 595–628. Addison-Wesley.

[Karp, 1972] Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. MIT Press, Cambridge.

[Kephart, 2002] Kephart, J. (2002). Software agents and the route to the information economy. *Proceedings of the National Academy of Sciences*, 99:7207–7213.

[Keskinocak et al., 2001] Keskinocak, P., Goodwin, R., et al. (2001). Decision support for managing an electronic supply chain. *Electronic Commerce Research*, 1(1-2):15–31.

[Khosla and Kitjongthawonkul, 2001] Khosla, R. and Kitjongthawonkul, S. (2001). A human-centered agent-based architecture for electronic brokerage. *Soft computing*, 6:405–411.

[Kowalczyk, 2000] Kowalczyk, R. (2000). On negotiation as a distributed fuzzy constraint satisfaction problem. In *Proceedings of DEXA e-Negotiation Workshop*, pages 631–637.

[Kowalczyk and Bui, 2000] Kowalczyk, R. and Bui, V. (2000). On fuzzy e-negotiation agents: Autonomous negotiation with incomplete and imprecise information. In *Proceedings of DEXA e-Negotiation Workshop*.

[Kraus, 2001] Kraus, S. (2001). *Strategic negotiation in multi-agent environments*. MIT Press, Cambridge, USA.

[Kreps, 1990] Kreps, D. (1990). *Game Theory and Economic Modeling*. Oxford University Press.

[Krulwich, 1996] Krulwich, B. (1996). The bargainfinder agent: comparison price shopping on the Internet. In Williams, J., editor, *Bots, and other Internet Beasties*, pages 257–263. Macmillan Computer Publishing.

[Kumar, 2001] Kumar, K. (2001). Technology for supporting supply-chain management. *Communications of the ACM*, 44(6):58–61.

[Lee and Billington, 1995] Lee, H. and Billington, C. (1995). The evolution of supply-chain-management models and practice at hewlett-packard. *Interface*, 25(5):42–63.

[Liang and Huang, 2000] Liang, T. and Huang, J. (2000). A framework for applying intelligent agents to support electronic trading. *Decision Support Systems*, 28:305–317.

[Lin, 1994] Lin, C. (1994). *Neural Fuzzy Control Systems with Structure and Parameter Learning*. World Scientific.

[Liu and Ye, 2001] Liu, J. and Ye, Y. (2001). Introduction to e-commerce agents: Marketplace solutions, security issues, and supply and demand. In Liu, J. and Ye, Y., editors, *E-Commerce Agents: Marketplace solutions, security issues, and supply and demand*, volume 2033 of *LNAI*, pages 1–6. Springer.

[Luck et al., 2003] Luck, M., McBurney, P., and Preist, C. (2003). Agent technology: Enabling next generation computing. *AgentLink*. ISBN 0854 327886.

[Luo et al., 2003a] Luo, X., Jennings, N. R., and Shadbolt, N. (2003a). Acquiring tradeoff preferences for automated negotiations: A case study. In *Proceedings of the 5th International Workshop on Agent-Mediated E-Commerce*, pages 41–48, Melbourne, Australia.

[Luo et al., 2003b] Luo, X., Jennings, N. R., and Shadbolt, N. (2003b). Knowledge-based acquisition of tradeoff preferences for negotiating agents. In *Proceedings of the 5th International Conference on Electronic Commerce*, pages 138–144, Pittsburgh, USA.

[Luo et al., 2003c] Luo, X., Jennings, N. R., Shadbolt, N., Leung, H., and Lee, J. (2003c). A fuzzy constraint based model for bilateral, multi-issue negotiation in semi-competitive environments. *Artificial Intelligence*, 148(1-2):53–102.

[Luo et al., 2003d] Luo, X., Lee, J., Leung, H., and Jennings, N. R. (2003d). Prioritised fuzzy constraint satisfaction problems: axioms, instantiation and validation. *Fuzzy sets and systems*, 136:151–188.

[Luo et al., 1994] Luo, X., Qiu, Y., and Cai, J. (1994). PILCF: A hybrid uncertain reasoning model. In *Proceedings of the Eighth Chinese Conference on Artificial Intelligence*, pages 70–79.

[Luo et al., 2001] Luo, X., Zhang, C., and Leung, H. (2001). Information sharing between heterogeneous uncertain reasoning models in a multi-agent environment: a case study. *International Journal of Approximate Reasoning*, 27(1):27–59.

[Ma, 1999] Ma, M. (1999). Agents in e-commerce. *Communications of the ACM*, 42(3):79–80.

[Mamdani, 1974] Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plants. In *Proceedings of IEE*, volume 121 of *12*, pages 1585–1588.

[Martinez et al., 2001] Martinez, M., Fouletier, P., Park, K., and Favrel, J. (2001). Virtual enterprise — organisation, evolution and control. *Int. J. Production Economics*, 74:225–238.

[Massotte, 1993] Massotte, P. (1993). New concepts and approaches for the management of complex production systems. In *Proceedings of the International conference on industrial engineering*, Belgium.

[Matos and Sierra, 1998] Matos, N. and Sierra, C. (1998). Evolutionary computing and negotiating agents. In *Agent mediated Electronic Commerce*, volume 1571 of *Lecture Notes in Artificial Intelligence*, pages 126–150.

[Matos et al., 1998] Matos, N., Sierra, C., and Jennings, N. R. (1998). Determining successful negotiation strategies: An evolutionary approach. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 182–189.

[McAfee and McMillan, 1987] McAfee, R. and McMillan, J. (1987). Auctions and bidding. *Journal of Economic Literature*, 25:699–738.

[McDermott, 2000] McDermott, P. (2000). Building trust into online business. *Network seurity*, 10:10–12.

[Meade et al., 1996] Meade, L., Sarkis, J., and Liles, D. (1996). Justifying strategic alliances: A prerequisite for virtual enterprising. *OMEGA: the International Journal of Management Science*, 25(1):29–42.

[Milgrom, 1989] Milgrom, P. (1989). Auctions and bidding: A primer. *Journal of Economic Perspectives*, pages 3–22.

[Milgrom and Robert, 1982] Milgrom, P. and Robert, J. (1982). A theory of auctions and competitive bidding. *Econometrica*, pages 1089–1122.

[Miller, 1999] Miller, M. (1999). *Complete Idiot's Guide to Online Auctions*, chapter 22. Alpha Books.

[Mohammadi et al., 2000] Mohammadi, S., Hassanzadeh, I., Mathur, R., and Patil, K. (2000). A new fuzzy decision-making procedure applied to emergency electric power distribution scheduling. *Engineering Applications of Artificial Intelligence*, 13(6):731–740.

[Murch and Johnson, 1999] Murch, R. and Johnson, T., editors (1999). *Intelligent Software Agents*, chapter 14, pages 131–137. Prentice Hall PTR, Upper Saddle River, New Jersey, USA.

[Norman et al., 2004] Norman, T. J., Preece, A., Chalmers, S., Jennings, N. R., Luck, M., Dang, V. D., Nguyen, T. D., Deora, V., Shao, J., Gray, A., and Fiddian, N. (2004). Agent-based formation of virtual organisations. *Int. J. Knowledge Based Systems*. to appear.

[O'Leary et al., 1997] O'Leary, D., Kuokka, D., and Plant, R. (1997). Artificial intelligence and virtual organizations. *Comminications of the ACM*, 40(1):52–59.

[Park et al., 1999] Park, S., Durfee, E., and Birmingham, W. (1999). An adaptive agent bidding strategy based on stochastic modeling. In *Proceedings of the Third International Conference on Autonomous Agents*, pages 147–153.

[Parunak, 1998] Parunak, H. (1998). Industrial and practical applications of DAI. In Weib, G., editor, *Multi-Agent Systems*. MIT Press, Cambridge, MA.

[Pedrycz, 1990] Pedrycz, W. (1990). Fuzzy sets in pattern recognition methodology and methods. *Pattern recognition*, 23(1):121–146.

[Perloff, 1998] Perloff, J. (1998). *Microeconomics*, chapter 2, pages 29–32. Addison Wesley.

[Preist, 1999] Preist, C. (1999). Commodity trading using an agent-based iterated double auction. In *Proceedings of the Third International Conference on Autonomous Agents*, pages 131–138.

[Preist et al., 2001] Preist, C., Byde, A., and Bartolini, C. (2001). Economic dynamics of agents in multiple autions. In *Proceedings of the fifth International Conference on Autonomous Agents*, pages 545–551, Montreal Canada.

[Preist and van Tol, 1998] Preist, C. and van Tol, M. (1998). Adaptive agents in a persistent shout double auction. In *Proceedings of the First International Conference on the Information and Computation Economies*, pages 11–18.

[Rabelo et al., 1999] Rabelo, R., Camarinha-Matos, L., and Afsarmanesh, H. (1999). Multi-agent-based agile scheduling. *Robotics and Autonomous Systems*, 27:15–28.

[Rassenti et al., 1982] Rassenti, S., Smith, V., and Bulfin, R. (1982). A combinatorial auction mechanism for airport time slot allocation. *Bell J. of Economics*, 13(2):402–417.

[Ribeiro, 1996] Ribeiro, R. (1996). Fuzzy multiple attribute decision making: a review and new preference elicitation techniques. *Fuzzy sets and systems*, 78:155–181.

[Ross, 1989] Ross, S. (1989). *Introduction to Probability Models*, chapter 5, pages 201–247. Academic Press.

[Rumelhart et al., 1986] Rumelhart, D., Hinton, G., et al. (1986). Learning internal represenations by error propagation. *Parallel distributed processing*, 1:318–362.

[Rust et al., 1991] Rust, J., Miller, J., and Palmer, R. (1991). Behavior of trading automata in a computerized double auction market. In Friedman, D. and Rust, J., editors, *The Double Auction Market: Institutions, Theories, and Evidence*, pages 155–198. Addison-Wesley.

[Sadrieh, 1998] Sadrieh, A. (1998). *The Alternating Double Auction Market: A Game Theoretic and Experimental Investigation*. Springer.

[Sandholm, 1993] Sandholm, T. (1993). An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the First International Conference on Artificial Intelligence*, pages 256–262.

[Sandholm, 1999a] Sandholm, T. (1999a). Automated negotiation. *Communications of the ACM*, 42(3):84–85.

[Sandholm, 1999b] Sandholm, T. (1999b). Distributed rational decision making. In Weiss, G., editor, *Multiagent Systems*, pages 201–258. The MIT Press.

[Sandholm, 2002] Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54.

[Sarkis and Sundarraj, 2002] Sarkis, J. and Sundarraj, R. (2002). Evolution of brokering; paradigms in e-commerce enabled manufacturing. *Int. J. Production Economics*, 75:21–31.

[Sashi and O'Leary, 2002] Sashi, C. and O'Leary, B. (2002). The role of Internet auctions in the expansion of B2B markets. *Industrial Marketing Management*, 31:103–110.

[Schmid and Lindemann, 1998] Schmid, B. and Lindemann, M. (1998). Elements of a reference model for electronic markets. In *Proceedings of the 31st HICSS Conference*, Hawaii.

[Schotter, 1994] Schotter, A. (1994). *Microeconomics: a modern approach*, chapter 14, pages 442–445. Harper Collins.

[Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating 'Word of Mouth'. In *Proceedings of the Computer Factors in Computing Systems*, pages 210–217, Denver, CO.

[Sharma, 2002] Sharma, A. (2002). Trends in Internet-based business-to-business marketing. *Industrial marketing management*, 31:77–84.

[Shaw, 2000] Shaw, M. (2000). Electronic commerce: State of art. In Shaw, M., Blanning, R., Strader, T., and Whinston, A., editors, *Handbook on Electronic Commerce*, chapter 1, pages 3–24. Springer.

[Shehory and Kraus, 1998] Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101:165–200.

[Sierra and Dignum, 2001] Sierra, C. and Dignum, F. (2001). Agent-mediated electronic commerce: Scientific and technological roadmap. In Dignum, F. and Sierra, C., editors, *Agent Mediated Electronic Commerce*, volume 1991 of *Lecture Notes in Computer Science*, pages 1–18. Springer.

[Sierra et al., 1997a] Sierra, C., Faratin, P., and Jennings, N. R. (1997a). A service-oriented negotiation model between autonomous agents. volume 1237 of *Lecture Notes in Artificial Intelligence*, pages 17–35.

[Sierra et al., 1997b] Sierra, C., Jennings, N. R., Noriega, P., and Parsons, S. (1997b). A framework for argumentation-based negotiation. In *Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, pages 177–192.

[Simon, 1997] Simon, H. (1997). *The Sciences of the Artificial*. The MIT Press, Cambridge, MA.

[Singh, 1999] Singh, S. (1999). Web home pages as advertisements. *Communications of the ACM*, 42(8):91–98.

[Smith, 1980] Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12):1104–1113.

[Sosnowski, 2000] Sosnowski, Z. (2000). Comments on fuzzyshell: A large-scale expert systems shell using fuzzy logic for uncertainty reasoning. *IEEE Trans. Fuzzy Systems*, 8(6):817–820. Authors' reply, p.821.

[Standifird, 2001] Standifird, S. (2001). Reputation and e-commerce: eBay auctions and the asymmetrical impact of positive and negative ratings. *Journal of Management*, 27:279–295.

[Stone et al., 2001] Stone, P., Littman, M., Singh, S., and Kearns, M. (2001). ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15:189–206.

[Subramani and Walden, 2000] Subramani, M. and Walden, E. (2000). Economic returns to firms from business-to-business electronic commerce initiatives: an empirical examination. In *Proceedings of the twenty first international conference on information systems*, pages 229–241.

[Sugeno, 1985] Sugeno, M. (1985). An introductory survey of fuzzy control. *Information Sciences*, 36:59–83.

[Sun et al., 1999] Sun, R., Chu, B., Wilhelm, R., and Yao, J. (1999). A CSP-based model for integrated supply chain. In *AAAI-99 Workshop on AI for Electronic Commerce*.

[Tan et al., 2000] Tan, G., Shaw, M., and Fulkerson, W. (2000). Web-based global supply chain management. In Shaw, M., Blanning, R., Strader, T., and Whinston, A., editors, *Handbook on Electronic Commerce*, chapter 22, pages 457–478. Springer.

[Tan and Tang, 2001] Tan, K. and Tang, K. (2001). Vehicle dispatching system based on taguchi-tuned fuzzy rules. *European Journal of Operational Research*, 128(3):545–557.

[Teich et al., 1999] Teich, J., Wallenius, H., and Wallenius, J. (1999). Multiple-issue auction and market algorithms for the world wide web. *Decision Support Systems*, 26:49–66.

[Till, 1998] Till, R. (1998). Transforming the way we do business. In Nash, T., editor, *Electronic Commerce*, pages 9–12. Kogan Page, London, UK.

[Tirole, 1998] Tirole, J. (1998). *The theory of industrial organisation*. The MIT press, Cambridge, MA.

[Tsvetovat and Sycara, 2000] Tsvetovat, M. and Sycara, K. (2000). Customer coalitions in the electronic marketplace. In *Proceedings of the fourth International Conference on Autonomous Agents*, pages 263–264, Spain.

[Tully, 2000] Tully, S. (2000). The b2b tool that really is changing the world. *Fortune*, pages 135–145.

[Tuma, 1998] Tuma, A. (1998). Configuration and coordination of virtual production networks. *Int. J. Production Economics*, 56-57:641–648.

[Turban et al., 1999] Turban, E., Lee, J., King, D., and Chung, H., editors (1999). *Electronic Commerce: A Managerial Perspective*. Prentice Hall.

[Vickrey, 1961] Vickrey, W. (1961). Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37.

[Vidal and Durfee, 1996] Vidal, J. and Durfee, E. (1996). The impact of nested agent models in an information economy. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 377–384.

[Vries and Vohra, 2001] Vries, S. and Vohra, R. (2001). Combinatorial auctions: A survey. Technical report. http://www-m9.mathematik.tu-muenchen.de/dm/homepages/devries/.

[Vulkan and Jennings, 2000] Vulkan, N. and Jennings, N. R. (2000). Efficient mechanisms for the supply of services in multi-agent environments. *International journal of decision support systems*, 28(1-2):5–19.

[Walsh and Wellman, 1999] Walsh, W. and Wellman, M. (1999). Modeling supply chains formation in multiagent systems. In *IJCAI-99 Workshop on Agent-Mediated Electronic Commerce*.

[Wellman et al., 2004] Wellman, M., Reeves, D., Lochner, K., and Vorobeychik, Y. (2004). Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36.

[Wellman et al., 2002] Wellman, M. P., Greenwald, A., Stone, P., and Wurman, P. R. (2002). The 2001 trading agent competition. In *Proceedings of the Fourteenth Conference on Innovative Applications of Artificial Intelligence*, Edmonton.

[Wolfstetter, 1999] Wolfstetter, E. (1999). *Topics in Microeconomics: Industrial Organization, Auctions and Incentives*, chapter 8, pages 182–242. Cambridge university press.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

[Wurman, 2001] Wurman, P. (2001). Dynamic pricing in the virtual marketplace. *IEEE Internet Computing*, 5:36–42.

[Wurman et al., 1998] Wurman, P., Wellman, M., and Walsh, W. (1998). The Michigan Internet Auctionbot: a configurable auction sever for human and software agents. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 301–308.

[Yager, 1994] Yager, R. (1994). Aggregation operators and fuzzy systems modeling. *Fuzzy Sets and Systems*, 67:129–145.

[Yager and Filev, 1994] Yager, R. and Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*. New York, USA.

[Yager and Rybalov, 1996] Yager, R. and Rybalov, A. (1996). Uninorm aggregation operators. *Fuzzy Sets and Systems*, 80:111–120.

[Yam and Koczy, 2000] Yam, Y. and Koczy, L. (2000). Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation. *IEEE Trans. Fuzzy Systems*, 8(6):761–772.

[Yamamoto and Sycara, 2001] Yamamoto, J. and Sycara, K. (2001). A stable and efficient buyer coalition formation scheme for e-marketplaces. In *Proceedings of the fifth International Conference on Autonomous Agents*, pages 576–583, Canada.

[Yao and Yao, 2001] Yao, J. and Yao, J. (2001). Fuzzy decision making for medical diagnosis based on fuzzy number and compositional rule of inference. *Fuzzy sets and systems*, 120(2):351–366.

[Ye et al., 2001] Ye, Y., Liu, J., and Moukas, A. (2001). Agents in electronic commerce. *Electronic Commerce Research*, 1(1-2):9–14.

[Yen, 1999] Yen, J. (1999). Fuzzy logic — a modern perspective. *IEEE Trans. Knowledge and Data Engineering*, 11(1):153–165.

[Young, 2001] Young, E. (2001). Web marketplaces that really work. Technical report. Fortune/CNET Tech.

[Zacharia et al., 2000] Zacharia, G., Moukas, A., and Maes, P. (2000). Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29:371–388.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

[Zeng, 2001] Zeng, D. (2001). Managing flexibility for inter-organizational electronic commerce. *Electronic Commerce Research*, 1(1-2):33–51.

[Zeng and Sycara, 1998] Zeng, D. and Sycara, K. (1998). Bayesian learning in negotiation. *Int. J. Human-Computer Studies*, 48:125–141.

[Zimmermann, 1996] Zimmermann, H.-J. (1996). *Fuzzy Set Theory and Its Applications*, chapter 11, pages 203–240. Kluwer Academic Publishers.

Appendix A

Allocators' Setup for SouthamptonTAC

This appendix describes the setup for allocators (see Section 4.2.4) of our agent, SouthamptonTAC.

The Allocator-1 deals with the allocation of available and unavailable goods and outputs what flights and hotels to buy during the first 11 minutes of a game. The setup for this allocator is as follows.

- Notation
  - (i) Let the 20 travel packages be described in the following way: ($outdate$ $indate$ $hotel$-$type$), where $outdate \in \{1, \cdots, 4\}$ indicating date out to Tampa; $indate \in \{2, \cdots, 5\}$ indicating the date back to TACtown; $hotel$-$type \in \{0, 1\}$ where 0 stands for S hotel and 1 for T hotel. The 20 valid travel packages can be expressed as: (1 2 1), (1 3 1), (1 4 1), (1 5 1), (2 3 1), (2 4 1), (2 5 1), (3 4 1), (3 5 1), (4 5 1), (1 2 0), (1 3 0), (1 4 0), (1 5 0), (2 3 0), (2 4 0), (2 5 0), (3 4 0), (3 5 0), and (4 5 0).
  - (ii) The entertainment tickets have 3 types for 4 days. Let ($type$ day) denote a ticket, where $type \in \{1, 2, 3\}$ and $day \in \{1, 2, 3, 4\}$. For each customer, various tickets can be expressed as: (1 1), (1 2), (1 3), (1 4), (2 1), (2 2), (2 3), (2 4), (3 1), (3 2), (3 3) and (3 4).
- Variables
  - (i) For customer $i \in \{1, \cdots, 8\}$, there are 20 variables $f_{i,j} \in \{0, 1\}$ where $j \in \{1, \cdots, 20\}$, each representing the jth travel package. Customer i is allocated to package j when $f_{i,j} = 1$. Thus, there are 160 variables for 8 customers.
  - (ii) $BUY[0], \cdots, BUY[3]$ represent the inflight tickets to buy; $BUY[4], \cdots, BUY[7]$ represent the outflight tickets to buy; $BUY[8], \cdots, BUY[11]$ represent the T rooms to bid for; $BUY[12], \cdots, BUY[15]$ represent the S rooms to bid for.
  - (iii) For each customer $i \in \{1, \cdots, 8\}$, there are 12 variables $e_{i,j} \in \{0, 1\}$ where $j \in \{1, \cdots, 12\}$, each representing the jth entertainment ticket. Customer i is allocated ticket j when $e_{i,j} = 1$.

- Constants

Let $OWN[0],\cdots,OWN[3]$ be the inflight tickets owned by the agent; $OWN[4],\cdots,$ $OWN[7]$ be the outflight tickets owned by the agent; $OWN[8],\cdots,OWN[11]$ represent the T rooms owned by the agent; $OWN[12],\cdots,OWN[15]$ represent the S rooms owned by the agent; and $OWN[16],\cdots,OWN[27]$ denote each kind of entertainment ticket owned by the agent.

- Constraints

(i) Each customer $i \in \{1,\cdots,8\}$ has only one valid package (8 constraints),

$$\sum_{j=1}^{20} f_{i,j} \leq 1.$$

(ii) The flights tickets that can be used must be less than the number the agent owns or that it will buy (8 constraints):

$$\sum_{i=1}^{8}(f_{i,1}+f_{i,2}+f_{i,3}+f_{i,4}+f_{i,11}+f_{i,12}+f_{i,13}+f_{i,14}) \leq OWN[0]+BUY[0],$$
$$\sum_{i=1}^{8}(f_{i,5}+f_{i,6}+f_{i,7}+f_{i,15}+f_{i,16}+f_{i,17}) \leq OWN[1]+BUY[1],$$
$$\sum_{i=1}^{8}(f_{i,8}+f_{i,9}+f_{i,18}+f_{i,19}) \leq OWN[2]+BUY[2],$$
$$\sum_{i=1}^{8}(f_{i,10}+f_{i,20}) \leq OWN[3]+BUY[3],$$
$$\sum_{i=1}^{8}(f_{i,1}+f_{i,11}) \leq OWN[4]+BUY[4],$$
$$\sum_{i=1}^{8}(f_{i,2}+f_{i,5}+f_{i,12}+f_{i,15}) \leq OWN[5]+BUY[5],$$
$$\sum_{i=1}^{8}(f_{i,3}+f_{i,6}+f_{i,8}+f_{i,13}+f_{i,16}+f_{i,18}) \leq OWN[6]+BUY[6],$$
$$\sum_{i=1}^{8}(f_{i,4}+f_{i,7}+f_{i,9}+f_{i,10}+f_{i,14}+f_{i,17}+f_{i,19}+f_{i,20}) \leq OWN[7]+BUY[7].$$

(iii) For customer $i \in \{1, \cdots, 8\}$ entertainment tickets must be used between the days in Tampa (32 constraints):

$$f_{i,1} + f_{i,2} + f_{i,3} + f_{i,4} + f_{i,11} + f_{i,12} + f_{i,13} + f_{i,14} \geq e_{i,1} + e_{i,5} + e_{i,9},$$
$$f_{i,2} + \cdots + f_{i,7} + f_{i,12} + \cdots + f_{i,17} \geq e_{i,2} + e_{i,6} + e_{i,10},$$
$$f_{i,3} + f_{i,4} + f_{i,6} + \cdots + f_{i,9} + f_{i,13} + f_{i,14} + f_{i,16} + \cdots + f_{i,19} \geq e_{i,3} + e_{i,7} + e_{i,11},$$
$$f_{i,4} + f_{i,7} + f_{i,9} + f_{i,10} + f_{i,14} + f_{i,17} + f_{i,19} + f_{i,20} \geq e_{i,4} + e_{i,8} + e_{i,12}.$$

(iv) The number of hotel rooms must be less than the number of rooms the agent owns and that it will bid for (8 constraints):

$$\sum_{i=1}^{8}(f_{i,1} + f_{i,2} + f_{i,3} + f_{i,4}) \leq OWN[8] + BUY[8],$$
$$\sum_{i=1}^{8}(f_{i,2} + f_{i,3} + f_{i,4} + f_{i,5} + f_{i,6} + f_{i,7}) \leq OWN[9] + BUY[9],$$
$$\sum_{i=1}^{8}(f_{i,3} + f_{i,4} + f_{i,6} + f_{i,7} + f_{i,8} + f_{i,9}) \leq OWN[10] + BUY[10],$$
$$\sum_{i=1}^{8}(f_{i,4} + f_{i,7} + f_{i,9} + f_{i,10}) \leq OWN[11] + BUY[11],$$
$$\sum_{i=1}^{8}(f_{i,11} + f_{i,12} + f_{i,13} + f_{i,14}) \leq OWN[12] + BUY[12],$$
$$\sum_{i=1}^{8}(f_{i,12} + f_{i,13} + f_{i,14} + f_{i,15} + f_{i,16} + f_{i,17}) \leq OWN[13] + BUY[13],$$
$$\sum_{i=1}^{8}(f_{i,13} + f_{i,14} + f_{i,16} + f_{i,17} + f_{i,18} + f_{i,19}) \leq OWN[14] + BUY[14],$$
$$\sum_{i=1}^{8}(f_{i,14} + f_{i,17} + f_{i,19} + f_{i,20}) \leq OWN[15] + BUY[15].$$

(v) For entertainment ticket $j \in \{1, \cdots, 12\}$ the number of tickets used must be less than the number the agent owns (12 constraints):

$$\sum_{i=1}^{8} e_{i,j} \leq OWN[j+15],$$

(vi) For customer $i \in \{1, \cdots, 8\}$ each type of entertainment ticket can only be used once (24 constraints):

$$\sum_{j=1}^{4} e_{i,j} \leq 1, \qquad \sum_{j=5}^{8} e_{i,j} \leq 1, \qquad \sum_{j=9}^{12} e_{i,j} \leq 1.$$

- Objective function

  The objective function is to maximise the following formula:

$$\sum_{i=1}^{8}\sum_{j=1}^{20}(f_{i,j} \times u_{i,j}) + \sum_{i=1}^{8}\sum_{j=1}^{12}(e_{i,j} \times E_{i,j}) - \sum_{p=0}^{15}(BUY[i] \times Price[i]),$$

  where $u_{i,j}$ is the utility if customer i chooses package j; $E_{i,j}$ is the preference value if customer i enjoys entertainment ticket j; and $Price[i]$ is the updated ask price of the corresponding auctions.

The Allocator-2 deals with the allocation of available and unavailable goods and outputs what flights and entertainment tickets to buy for the final minute of a game when the hotel rooms are finialised. The setup for this allocator is as follows.

- Variables (276 variables). The difference with *allocator-1* is that there are no variables BUY[8],$\cdots$,BUY[15], since all the hotel auctions have closed. However, BUY[16],$\cdots$,BUY[27], indicating the number of entertainment tickets to buy for each of the 12 tickets, are added.
- Constraints (92 constraints). All but the following are the same as those of *allocator-1*: the constraints (v) above are changed to

$$\sum_{i=1}^{8} e_{i,j} \leq OWN[j+15] + BUY[j+15],$$

  where $j \in \{1, \cdots, 12\}$. This means the allocator can also calculate which entertainment tickets to bid for in order to get the optimal utility. All the other constraints are the same as *allocator-1*.
- Objective function. The objective function is to maximise the following formula:

$$\sum_{i=1}^{8}\sum_{j=1}^{20}(f_{i,j} \times u_{i,j}) + \sum_{i=1}^{8}\sum_{j=1}^{12}(e_{i,j} \times E_{i,j}) - \sum_{p=0}^{7}(BUY[i] \times Price[i]) - \sum_{p=16}^{27}(BUY[i] \times Price[i]),$$

  where $u_{i,j}$ is the utility if customer i chooses package j; $E_{i,j}$ is the preference value if customer i enjoys entertainment ticket j; and $Price[i]$ ($0 \leq i \leq 7$) is the updated ask price of the flight auctions and $Price[i]$ ($16 \leq i \leq 27$) is the entertainment ticket price in the corresponding auctions.