Development of Automated Dynamic Bidding Agents for Final Price Prediction in Online Auctions

A Thesis Submitted for the Degree of

Doctor of Philosophy

By

Preetinder Kaur

February 2014

UNIVERSITY OF TECHNOLOGY

SCHOOL OF SOFTWARE

The undersigned hereby certify that this thesis entitled "**Development of Automated Dynamic Bidding Agents for Final Price Prediction in Online Auctions**" by **Preetinder Kaur** has been read and is fully adequate, in scope and in quality, as a thesis for the degree of **Doctor of Philosophy**.

Research Supervisor: ________________________

Dr. Madhu Goyal

Dated: February 2014

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

____

Signature of Author

Dated: February 2014

Acknowledgments

I wish to express my sincere gratitude to my principal PhD supervisor, Dr. Madhu Goyal for offering me the opportunity to begin my research study. Thank you for your encouragement, precious guidance, patience and continuous support over my whole PhD period. This thesis would never have been completed without your meticulous examination, critical comments and suggestions. Your profound knowledge, generosity and sincere approach influenced me deeply, and will be sure to benefit me not only in my future research work but in my personal life. I would also like to express my appreciation to my co-supervisor Prof. Jie Lu for her valuable suggestions on my research and kind support during my PhD study.

Thanks to the University of Technology (UTS), Sydney, which provided me with not only a wonderful education, but also the UTS Doctoral Scholarship (UTSD) that made this work possible. This support was deeply welcome. I am grateful to the Centre for Quantum Computation and Intelligent Systems (QCIS) and the Faculty of Engineering and Information Technology, (School of Software) for providing me with essential resources during my PhD research at UTS. I would also like to thank Ms. Debra Shulkes for her help with proofreading and for suggesting improvements to correct grammatical and presentation problems in my thesis.

I could not have finished this thesis without the support of my family. Most of all, I would like to express my deepest gratitude to my beloved husband, Yashpal Singh. Thank you for supporting me and always being there for me throughout all the ups and downs of my PhD research. Pursuing a PhD was a long-term challenge, and it would not have been possible for me to complete my research study without your continuous encouragement and help. I would like to express the greatest appreciation to my daughter, Khyati Singh whose unlimited love always motivated me during my research. A special thanks to my sister-in-law, Daljit Kaur without whom this PhD research might not have been initiated. I would like to express my heartfelt appreciation to my parents and parents-in-law for the endless ways that they have inspired me throughout my life.

Finally, I would like to extend my gratitude to all the members of the Decision Systems and e-Service Intelligence (DeSI) Laboratory for their support and encouragement during my studies. Thanks for the unforgettable memories of the DeSI Lab workshops.

Table of Contents

List of Figures

List of Tables

Abstract

Online auctions have emerged as a well-recognised paradigm of item exchange over the past few years. In these environments, software agents are being used increasingly and promisingly to bid on or trade goods. This thesis presents an automated dynamic bidding agent framework that makes use of machine learning techniques to forecast bid amounts in simultaneous auctions of the same or similar items. The availability of numerous auctions of similar items complicates the situation of bidders who wish to choose the auction where their participation will give maximum surplus. These bidders also face a perpetual dilemma about how to predict an item's bargain price. Further, the diverse price dynamics of auctions for the same or similar items affect both the choice of auction and the valuation of the auctioned items. There is, thus, a critical need to characterise auctions based on their price dynamics before selecting one to compete in and assessing the true value of the auctioned items.

The main contributions of this thesis are its development of: (i) an automated dynamic bidding agent framework, (ii) an initial price estimation methodology for choosing an auction and assessing the value of auctioned goods, (iii) a final price prediction methodology that designs bidding strategies for buyers with different bidding behaviours and (iv) a simulated electronic marketplace for implementing and evaluating the performance of bidding agents.

The automated dynamic bidding agent (ADBA) framework selects an auction to participate in and predicts its final price in two phases: the first gives an initial estimation and the second phase delivers a final price prediction. The methodology for initial price estimation finds an auction to compete in and assesses the value of the auctioned item using data mining techniques. It handles the problem of diverse price dynamics in auctions for the same or similar items, using a clustering-based bid mapping and selection approach to locate the auction where participation would give maximum surplus. The value of the item is assessed with parametric and non-parametric machine learning approaches to predict the auction's closing price. The proposed approach is validated using real online auction datasets. These results

demonstrate that this clustering-based price prediction approach outperforms existing methodologies in terms of prediction accuracy.

This thesis also introduces a methodology for final price estimation which designs bidding strategies to address buyers' different bidding behaviours. This draws on two approaches: negotiation decision functions and fuzzy reasoning techniques. The bidding strategies are designed based on the bidder's own attitude to win the auction and the behaviour of rival bidders. A simulated electronic marketplace is implemented and developed using Java Agent DEvelopment Framework (JADE). The marketplace is also used to demonstrate the performance of the bidding strategies. The outcomes for heterogeneous and homogeneous bidders are measured separately in a wide variety of test environments subject to different auction settings and bidding restrictions. The results show that ADBA agents who follow this study's bidding strategies outperform other existing agents in most settings in terms of their success rate and expected utility.

Chapter 1

Introduction

Section 1.1 sets out the background and motivation for this research; it outlines the problem that this thesis aims to solve. Section 1.2 explains the research objective and aims. The contributions made by this study are presented in Section 1.3. Section 1.4 describes the structure of this thesis, and Section 1.5 highlights publications related to this study.

1.1 Background and Motivation

The advent of electronic commerce has dramatically advanced traditional trading mechanisms, and online auction settings like eBay and Amazon have been emrged as a powerful tool for allocating goods and resources. Discovery of the new markets and the possibilities opened by online trading has heightened the sellers' and buyers' interest. In recent years, online auctions have become a widely recognised paradigm of item exchange, offering traders greater flexibility in terms of both time and geography. In online auction commerce, traders barter over products, applying specific trading rules over the Internet which support different auction formats. Common online formats are English, Dutch, First-price sealed-bid and Second-price sealed-bid auctions (Haruvy & Popkowski Leszczyc 2010; Ockenfels et al. 2006).

In English auctions, also known as ascending-bid format auctions, bidders bid incrementally for the auction's duration. In this format, each recent bid exceeds the previous highest bid. The bidder who has the highest bid at the end of the auction is the winner, and he pays his own bid. Dutch auctions start with a high price, which is decreased until a bidder accepts the current price and he also pays his own bid (Ockenfels et al. 2006). In First-price sealed-bid auctions, bidders place single bids secretly; the bidder with the highest offer wins and he pays his own bid. Second-price sealed-bid auctions are similar to First-price sealed-bid auctions except that in Second-price sealed-bid auctions, the winning bid is the second highest rather than the highest bid. The English auction is the most common auction type, and is used by online auctioneers at eBay, Amazon, etc.

Bidders in this marketplace often feel challenged when looking for the best bidding strategies to win the auction. Moreover, there are commonly many auctions selling the desired item at any one time. Deciding which auction to participate in, whether to bid early or late, and how much to bid are very complicated issues for bidders (Jank & Zhang 2011; Park & Bradlow 2005). The difficult and time-consuming processes of analysing, selecting and making bids and monitoring developments need to be automated in order to assist buyers with their bidding.

The emergence of software agent technology has created an innovative framework for developing online auction mechanisms. Because of their extraordinary adaptive capabilities and trainability, software agents have become an integral component of online trading systems for buying and selling goods. These software agents represent expert bidders or sellers to fulfil their requirements and pursue their beliefs, and are consequently trained to achieve these aims. Software agents can perform various tasks like analysing the current market to predict future trends, deciding bid amounts at a particular moment in time, evaluating different auction parameters and monitoring auction progress, as well as many more. These negotiating agents outperform their human counterparts because of the systematic approach they take to managing complex decision-making situations effectively (Byde et al. 2002). This creates more opportunities for expert bidders and sellers to maximise satisfaction and profit. Software agents make decisions on behalf of the consumer and seek to guarantee that items are delivered to the buyer's preferences. To function well, these agents must have prior knowledge of the auction's features, whether these are certain or uncertain.

eBay is one of the major global online marketplaces and currently the biggest consumer-to-consumer online auction site. Founded in 1995, eBay Inc. has attracted over 112 million active users and gained a net revenue of $14.1 billion for 2012[1]. eBay does not, however, actually sell any goods that it owns; it only makes the process of displaying and selling goods easier by facilitating the bidding and payment processes. In virtual terms, eBay provides a marketplace where buyers and sellers meet and transact. eBay is a great source of high quality data as it keeps detailed records of completed auctions, and this data has been used extensively by researchers to solve research issues involving online auctions (Bajari & Hortacsu 2003b; Bapna et al. 2003; Jank & Shmueli 2005; Raykhel & Ventura 2009; Roth & Ockenfels 2002; Van Heijst et al. 2008; Wilcox 2000).

eBay-style auctions adopt the English auction format, except with regard to the payment of the winning bid (Haruvy & Popkowski Leszczyc 2010). In eBay auctions, the winner is the bidder with the highest value bid, but instead of paying his own bid, he pays the second-highest bid plus the amount of one bid increment. Bidders in these auctions do not, however, bid their maximum valuation of the item offered. This is either because they do not grasp that they should do so or they simply have trouble figuring out what their maximum valuation is. These bidders are typically afraid of winning the auction at a price above the true value of the item, a phenomenon sometimes known as 'the-winner's curse'. This problem occurs because the bidder lacks information about the true value of the item. In this respect, closing price prediction can assist bidders to establish the true value of the item on auction and thus finalise the maximum amount that they are willing to pay. This helps them to develop a bidding strategy to win the auction if the price is appropriate to an item's value, and it also allows experienced bidders to win auctions at a lower cost (Sun 2005). By presenting

[1] eBay Annual Report 2012

consistent information, closing price prediction supports buyers to make more informed bidding decisions (Raykhel & Ventura 2008). This also solves some of the information asymmetry problem for buyers, cutting down transaction time and cost. At the same time, sellers can use predictions to identify when the market favours selling their products and assess the value of their inventory better. They can also optimise auction attributes and the selling price for their wares (Xuefeng et al. 2006).

Predicting the end-price of an online auction is challenging because it depends on auction's attributes which are dynamic in nature (Lucking Reiley et al. 2007; Van Heijst et al. 2008; Xuefeng et al. 2006). The amount of the winning bid can be forecasted effectively by analysing the data produced as an auction progresses (historical data). Analysis of the plethora of data produced in online auction environments can be done using data mining techniques to predict the end-price of an online auction (Jank & Shmueli 2005; Kehagias & Mitkas 2007; Nikolaidou & Mitkas 2009). Data from a series of identical or similar closed auctions has been used in the past to forecast the winning bid by exploiting regression, classification and regression trees, multi-class classification and multiple binary classification tasks (Ghani & Simmons 2004; Van Heijst et al. 2008). The history of an ongoing auction also contains significant information; this can be exploited for short-term forecasting of the next bid by using support vector machines and functional k-nearest neighbor approaches (Zhang et al. 2010), clustering (Kehagias & Mitkas 2007) and regression and classification techniques (Xuefeng et al. 2006).

Furthermore, bidders repeatedly adjust their bids towards their maximum valuation of an item based on the time left in the auction and the bids placed by other participants. This triggers different bidding behaviours. Analysis of these bidding behaviours suggests that agents can be categorised as evaluators, participators, opportunists, snipers, unmaskers or shill bidders (Bapna et al. 2004; Trevathan & Read 2009). Evaluators have a clear idea of their valuation of the item and place a single, significantly high bid during the early phase of the auction. Participators make a low initial bid and then place ascending bids as the auction progresses. Opportunists place the minimum required bid just before the auction closes. Snipers bid in the closing seconds of the auction. Unmaskers make multiple bids, bidding continuously over a short span of time, without any intermediate bids while the auction is progressing. Shill bidders do not intend to win the auction and place fake bids to increase the end-price of the item. These different types of bidders all perform continuous, early or late bidding based on their bidding behaviours. Late bidding especially has drawn considerable attention from professionals and researchers of eBay-style auctions, which apply hard closing rules with fixed end-times (Du et al. 2010; Ockenfels et al. 2006) . The decision of bidders in these environments to postpone their bids until the auction's last moments is indeed a rational and effective winning strategy (Vergano 2006). This may be the best response to a variety of incremental bidding strategies because late bidders deprive incremental bidders of ample response time; they perform intelligent bidding, drawing on the information they have gathered from the earlier bids of these incremental bidders. Late bidders can also protect their private information about the value of an item, avoiding bidding wars with incremental bidders who compete in these auctions; this leads to higher payoffs for the winners (Ockenfels & Roth 2002). There is, thus, a clear need to design a mechanism which determines the amount to bid at a particular moment, taking into account these different bidding behaviours.

Against this background, the research reported in this thesis addresses the problem of how to develop successful bidding strategies for the different bidding behaviours of the buyers who take part in eBay auctions. When designing bidding strategies for the eBay environment, there are a number of common challenges that have to be dealt with. Of all of these, predicting the closing prices of ongoing auctions and allowing for the behaviour of different bidders are the most critical concerns for those trying to find optimal bidding strategies for bidding agents. Based on this analysis, the research in this thesis takes as its object a theory and methodology for predicting the closing price of an auction, choosing an auction to participate in and designing bidding strategies for bidding agents.

1.2 Research Objective and Aims

The objective of this thesis is to develop an automated and dynamic bidding agent framework that handles simultaneous online auctions of the same or similar items with hard closing rules. This bidding agent assists bidders to make decisions by designing bidding strategies tailored to their different bidding behaviours.

In order to achieve this objective, this thesis tackles five aims:

1. The first aim is to develop a methodology for selecting the auction where participation will bring maximum surplus. The availability of many auctions of the same or similar items is misleading for bidders, who are faced with choosing the appropriate one to compete in. These auctions have different attributes, which play a vital role in the selection process. This aim is achieved by developing an algorithm for auction selection using a clustering-based bid mapping and selection approach by exploiting different auction attributes.

2. The second aim is to assess the value of an item in the target auction so as to help bidders in finalising the maximum price they are willing to pay for it. Bidders resist bidding their maximum valuation of an item during auctions. This is due to two main reasons; they may be afraid that they will end up taking the item away at a higher price, or they have trouble in figuring out what their maximum valuation is. The value of the item also depends on the path that the price takes during an auction, a trajectory also known as the price dynamics of the auction. This thesis therefore develops a methodology to assess the true value of an item based on the price dynamics of its auction.

3. The third aim is to use the diverse price dynamics of auctions of the same or similar items when selecting an auction and valuing the item(s) on offer. Many existing methods of auction selection and value assessment are static: they only make use of information available at the beginning of an auction (Ghani &

Simmons 2004; Van Heijst et al. 2008). These models cannot incorporate the price dynamics of auctions for similar items, a concern that only a few studies take into account (Dass et al. 2011; Kehagias et al. 2005; Wang et al. 2008). However, the price dynamics of auctions are different, even when they deal with similar items. These dynamics have a strong influence on the selection of an auction and value assessment of the auctioned item (Jank & Shmueli 2005). This thesis, thus, develops a methodology which characterises auctions of the same or similar items based on their price dynamics before selecting an auction for participation and assessing the true value of the item on offer.

4. The fourth aim is to design bidding strategies for bidders to win the auction based on their bidding behaviours. It is difficult to make bidding decisions for bidders—even when an auction that gives maximum surplus has been chosen and we know the true value of the item being auctioned—because the bid which a bidder places at a particular point in time will also depend on his bidding behaviour and the bids of other participants. This aim is achieved by developing a methodology for devising bidding strategies for bidders using their own behaviour and the behaviour of competing participants.

5. The fifth aim is to design a simulated marketplace where we can evaluate the performance of the bidding strategies that have been designed for buyers.

1.3 Contributions

The work described in this thesis makes a number of important contributions to the state of the art in the design of bidding agents that can participate in simultaneous online auctions of the same or similar items. Specifically:

1. It develops an automated dynamic bidding agent framework for simultaneous online auctions of identical or similar items. This framework has two phases: phase 1 selects an auction to participate in and assesses the value of the auctioned item based on clustering and a bid mapping and selection approach. Phase 2, on the other hand, designs bidding strategies for buyers with different

bidding behaviour using the output of phase 1 and exploiting negotiation decision functions and fuzzy reasoning techniques.

2. It develops a closing price prediction methodology that an agent can use to assess the value of the auctioned item. The price prediction method employs a bid mapper and selector algorithm (BMS) based on the diverse price dynamics of auctions of the same or similar items. The effectiveness of the methodology is empirically demonstrated using eBay auctions, and the proposed methodology outperforms other related closing price prediction methodologies in the literature.
3. It designs bidding strategies that agents can pair with their different bidding behaviours when participating in online auctions. The effectiveness of the bidding strategies is demonstrated through empirical benchmarking against other strategies proposed in the literature. The evaluation shows the designed bidding strategies outperform alternatives in a wide range of situations.
4. It develops an electronic marketplace and uses automated dynamic bidding agents (ADBA) to demonstrate the bidding strategies in action. The ADBA framework is implemented using Java Agent DEvelopment Framework Environment (JADE).

1.4 Organisation of this Thesis

This thesis consists of seven chapters:

Chapter 1 presents an overview of this research including research issues, the research objective and research contributions.

Chapter 2 surveys and analyses the general state of the art related to the topics of online auctions, software agents for online auctions, price prediction and bidding strategies for online auctions.

Chapter 3 introduces a bidding agent framework for simultaneous online auctions of the same or similar items.

Chapter 4 designs a comprehensive method for initial price estimation which selects an auction to participate in from a number of simultaneous auctions of the same or similar items and assesses the value of the auctioned item. The price is estimated using a clustering approach and a bid mapping and selection technique that characterises similar auctions based on their price dynamics for decision making. Empirical evaluation shows that the proposed methodology outperforms other similar methodologies in the literature.

Chapter 5 designs bidding strategies to forecast bid amounts at a particular moment in time based on different bidding behaviours of bidders. The ADBA agent uses time- and behaviour- dependent negotiation decision functions and fuzzy reasoning techniques to design these bidding strategies.

Chapter 6 develops a simulated electronic marketplace to demonstrate performance of the designed bidding strategies by implementing an automated dynamic bidding agent (ADBA). The bidding agent is developed using a Java Agent DEvelopment Framework (JADE) environment fully implemented in JAVA language; the aim here is to develop software agents that comply with Foundation for Intelligent Physical Agents (FIPA) specifications. Empirical evaluations show that the bidding strategies designed perform more effectively than the bidding strategies proposed in other auction literature in a wide range of scenarios.

Chapter 7 concludes the thesis and outlines future research directions.

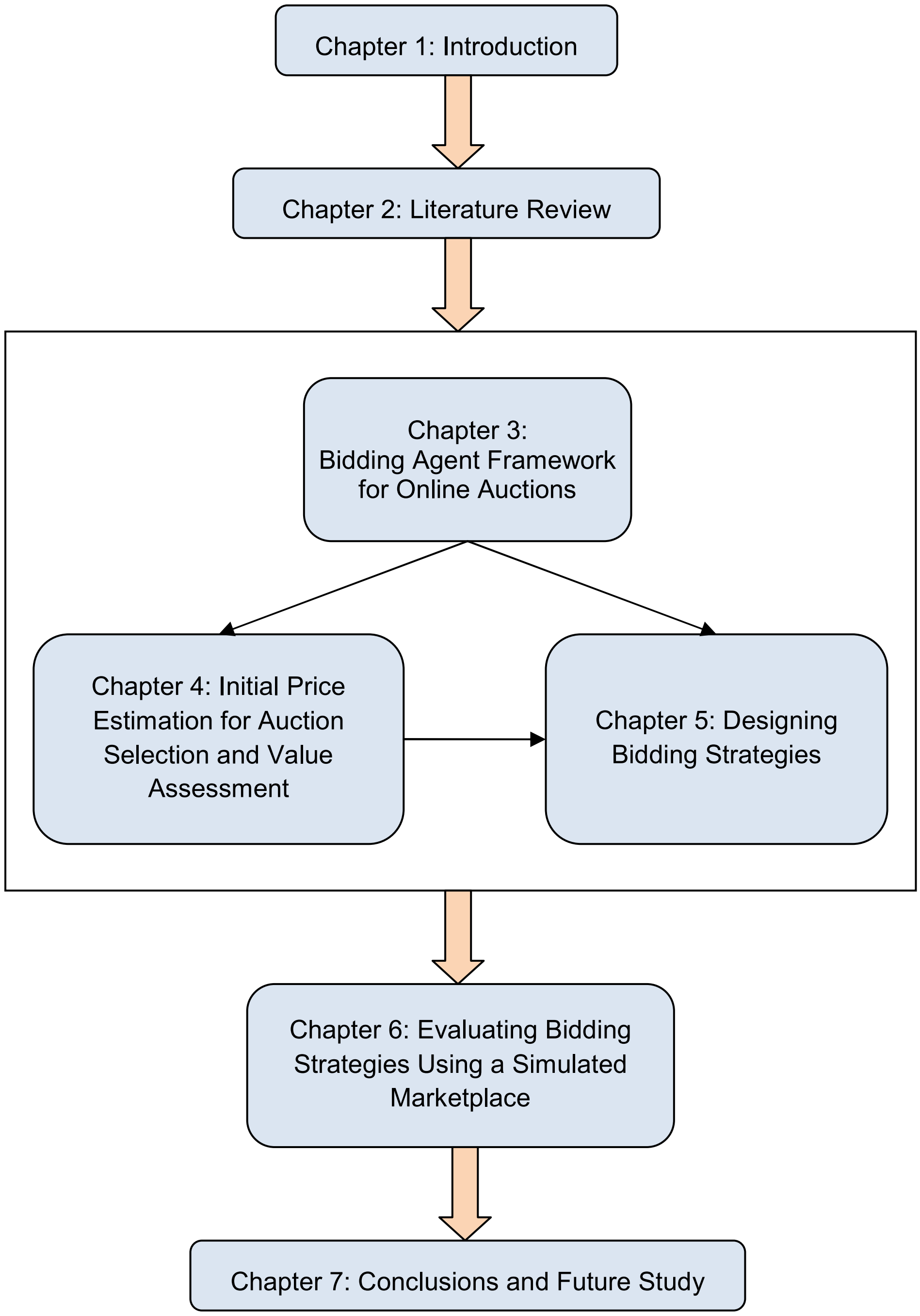


Figure 1.1 Thesis structure

1.5 Publications Related to this Thesis

Below is a list of papers (co-)authored by me which were co-published or else submitted and accepted for publication during this PhD study. I also provide details of my relevant work in progress.

Publications in Refereed Journals

- Kaur, P., Goyal, M. & Lu, J. 2012. 'An Integrated Model for a Price Forecasting Agent in Online Auctions', *Journal of Internet Commerce*, vol. 11, no. 3, pp. 208-225.
- Kaur, P., Goyal, M. & Lu, J. 2014. 'A Clustering based Approach to End Price Prediction in Online Auctions', *International Journal of Computaional Intelligence and Applicatioins,* (Under Review)

Publications in Refereed Conference Proceedings

- Kaur, P., Goyal, M. & Lu, J. 2012. 'Price Forecasting using Dynamic Assessment of Market Conditions and Agent's Bidding Behavior', in *Proceedings of the 19th International Conference on Neural Information Processing, ICONIP2012*, Vol. I. Springer-Verlag Berlin Heidelberg , pp.100-108.
- Kaur, P., Goyal, M. & Lu, J. 2013. 'A Proficient and Dynamic Bidding Agent for Online Auctions', in *Proceedings of the 8th International Workshop on Agents and Data Mining Interaction (ADMI-12) joint with the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS2012*), Springer. Berlin Heidelberg, pp. 178-190.
- Kaur, P., Goyal. M. and Lu, J. 2011. 'Data Mining Driven Agents For Predicting Online Auction's End Price', in ed B. Bouchon-Meunier (ed), *IEEE Symposium on Computational Intelligence and Data Mining in IEEE Symposium Series on Computational Intelligence 2011 (SSCI 2011)*, IEEE, USA, pp. 141-147.

- Kaur, P., Goyal, M. & Lu, J. 2011. 'Pricing Analysis in Online Auctions using Clustering and Regression Tree Approach' in L.Cao et al. (eds.) *Proceedings of the 7th International Workshop on Agents and Data Mining Interaction (ADMI-11) joint with the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS2011)*. Springer-Verlag Berlin / Heidelberg, pp. 248-257.
- Kaur, J., Goyal, M. & Lu, J. 2014. 'A Price Prediction Model for Online Auctions using Fuzzy Reasoning Techniques', in *Special Session on Handling Uncertainties in Big Data by Fuzzy Systems in IEEE International Conference on Fuzzy Systems, WCCI 2014*. IEEE, (In press)

Publications in Progress

- 'Price Prediction by Designing Bidding Strategies for Different Bidder Behaviours.' (To be submitted to *IEEE Transactions on Knowledge and Data Engineering*.)

Chapter 2

Literature Review

This chapter briefly reviews current research progress in the fields related to the objective and aims identified in Chapter 1. Section 2.1 introduces the role of software agents in automated online auctions by presenting a bidder interaction model. Section 2.2 highlights common frameworks used to design bidding agents for online auctions. Section 2.3 offers an overview of approaches in the literature to solving the price prediction problem. Section 2.4 outlines different bidding behaviours of buyers. Section 2.5 describes the methodologies for designing bidding strategies for online auctions. Finally section 2.6 briefly summarises the main areas covered in this chapter.

2.1 Role of Software Agents in Online Auctions

The emergence of software agent technology has created an innovative framework for online auction systems. Software agents have become an integral component of online goods trading (purchase and sale) due to their extraordinarily adaptive capabilities and trainability. These software components can operate autonomously, communicate with other agents or the user and effectively monitor the state of the operating environment on behalf of traders (Anthony & Jennings 2002; Byde et al. 2002; Chang & Chen 1996; Greenwald & Stone 2001). These bidding agent properties can be realized using a bidder interaction model.

2.1.1 Bidder Interaction Model

A model of bidder interactions in online auctions is essential for understanding the role of bidding agents and the tasks that require automation in the trading process

(Indrawan 2001). Using this model, the complete bidder's interactions are modularized into different subsystems: prepurchase interactions, purchase interactions and postpurchase interactions (Figure 2.1).

The prepurchase interaction phase includes product discovery, auction search, selection of an auction for participation and the bidding process. In the discovery phase, the bidder gets to know various products and their specifications which he can explore according to his needs. In the search phase, the bidder combs the large information space to find a set of auctions to participate in that meet his requirements. Each of these auctions may have different attributes, and in the selection phase, the attributes of these auctions are compared. Selection criteria are developed so that he can choose the auction where participation will bring the most surplus. The surplus is the return that bidders enjoy by winning an auction at a price lower than its predicted closing price (Dass et al. 2011). To automate the tasks in this phase, the bidding agent needs to roam autonomously across different websites and collect information about auction attributes, then evaluate all of the presented alternatives and make a decision. This means that the bidding agent must exhibit autonomous, mobile and intelligent features. The last phase of the pre-purchase interaction is the bidding process. Once the bidder identifies his auction of choice, he can start bidding there. During this phase, the bidder competes with other bidders according to his bidding preferences and tries to win the auction with maximum gain. Automation of this phase requires a bidding agent that can communicate with other agents, react in line with the bids placed by other bidders, be goal-oriented and make decisions based on the bidder's bidding preferences (Ashri et al. 2003). To do all of this, the bidding agent needs to possess social, reactive, pro-active and adaptive abilities.

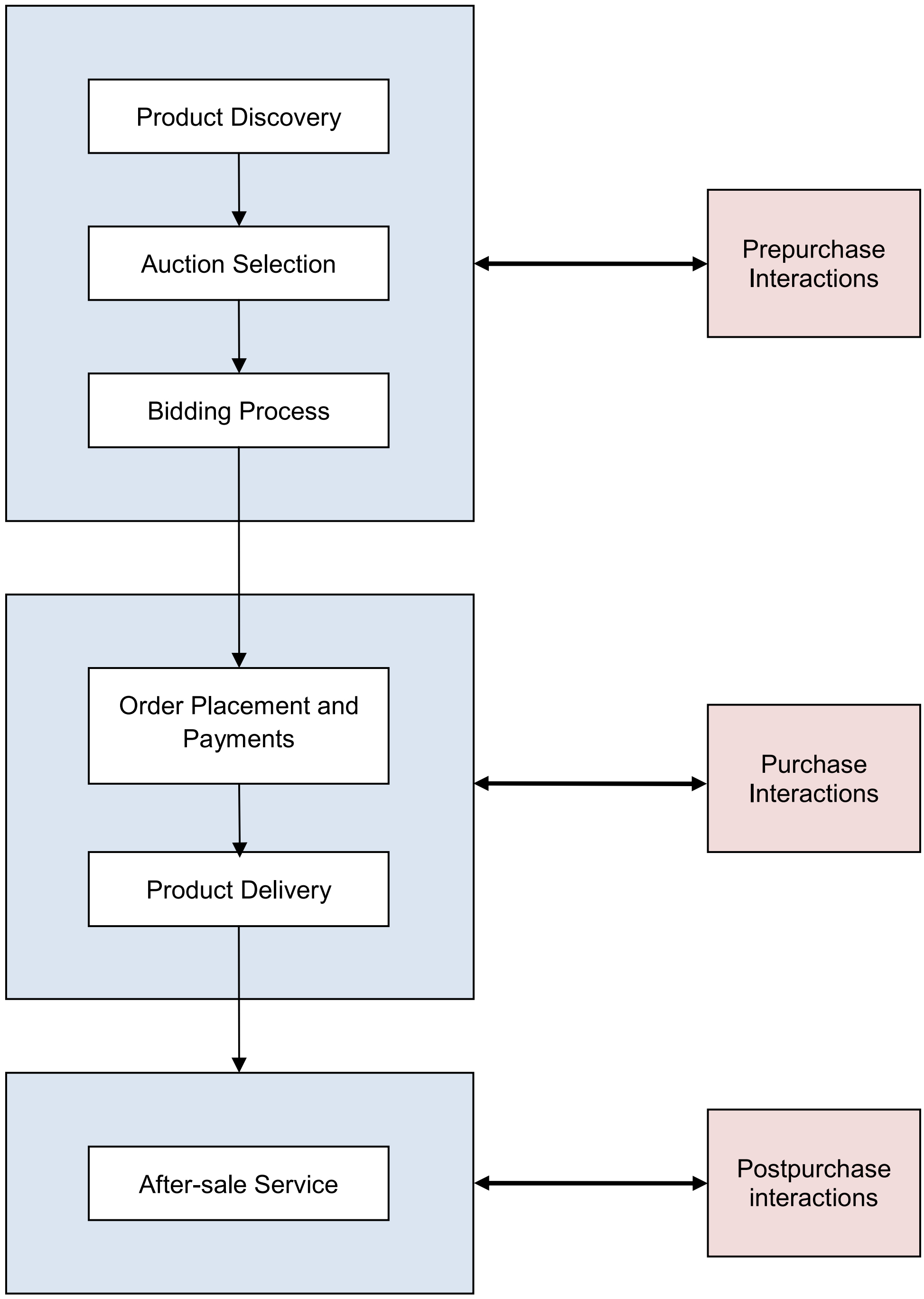


Figure 2.1 Bidder interactions based on consumer interaction model (Indrawan 2001)

The purchase interaction phase involves the placement of an order, the payment process and product delivery. In contrast to the prepurchase phase, as outlined above, the automation of this purchase stage requires a high level of security measures.

Automation of the postpurchase phase is more important from the auctioneer's perspective than that of the bidder. In this phase, agents can be used to collect the bidder's profiles, as required for dispensing personalised postpurchase services (Indrawan 2001). Automation must allow agents to serve as mediators between the bidder and the auctioneer. As mediators, they should be able to perform tasks related to the filtering and retrieval of information as well as personalised evaluation, complex coordination and time-based interactions.Therefore, agents in this postpurchase phase should be personalised, autonomous, mobile and social.

This thesis focuses on designing a dynamic bidding agent that automates two key aspects of the pre-processing phase of the bidder interaction model: the auction selection and bidding process stages.

2.1.2 Software Agent Properties

As we have seen, to automate different phases of the trading process, software agents need different properties. The following list explores these properties in detail:

- *Autonomous*: An autonomous agent interacts with its environment without the direct intervention of other agents and has control over its own actions and internal state (Jennings & Wooldridge 1996). Greater autonomy in an agent, however, reduces predictability. Absolute autonomy is undesirable given the complete unpredictability that would result; agents must serve some purpose which inadvertently constrains them. It is clear, therefore, that some control over the agent's behaviour should be built into the design objectives or other preferences. A bidding agent that interacts in an environment with other agents should have restricted autonomy since it has to comply with social norms.

Nevertheless, a sociable and responsible bidding agent can remain autonomous when performing specific tasks. It should be able to perform the majority of its problem-solving tasks without the direct intervention of humans or other agents. Having control over its own actions will mean the bidding agent can make decisions about which auction to join without consulting the user frequently.

- *Proactive*: An intelligent agent exhibiting proactive behaviour is goal-oriented. These goals should be built into the system in terms of the design objectives (Wooldridge 2008). When a procedure is designed with preconditions that should be satisfied, the effects of carrying out that procedure, or 'post-conditions' should be described. If the preconditions are met and the procedure executed correctly, then the specified post-conditions will be true. A bidding agent with its own built-in goals in terms of its objectives and desired preferences, will attempt to accomplish these goals by planning a course of action and having alternatives to various situations. It should be opportunistic and take initiative where appropriate.
- *Reactive*: Only undertaking goal-oriented behaviour is a necessary but not sufficient feature of the bidding agent if it is designed to act smartly. This goal-oriented behaviour assumes that when the agent is working, the preconditions remain valid and the environment does not change. This behaviour also assumes that the goal and the conditions for pursuing this goal remain valid at least until the agent's actions end. These assumptions of goal-oriented behaviour are not realistic in dynamic and uncertain environments in which other agents act. The bidding agent, therefore, must not only try to achieve its own goals but respond according to the perceived environment and the changes that occur in it (Figure 2.2) (Brooks 1991; Fasli 2007; Maamar 2002). Changes in the environment will affect the agent's own goals and limit

the available options and courses of action that it can take in order to satisfy them. The bidding agent should seek to satisfy its own goals, while at the same time reacting to the environment by absorbing new information and modifying its actions. The bidding agent needs to perceive its environment and respond in a timely fashion to dynamic and unpredictable changes there.

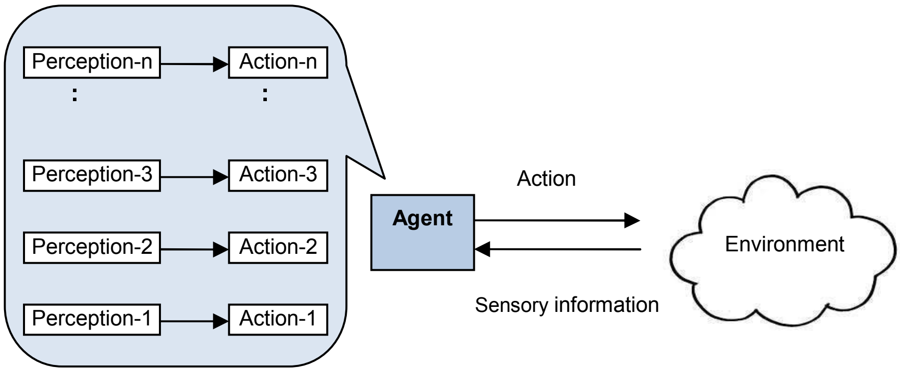


Figure 2.2 Reactive agent: mapping of perceptions to actions (Fasli 2007)

- *Social*: The bidding agent should interact with other agents and humans to complete its own problem solving and also help others to solve their problems where appropriate (Maamar 2002). It should place bids on behalf of the bidder according to the bids made by other bidders and the messages sent by the seller of the item. The seller and the bidding agent should, thus, communicate with each other, meaning that the agent must possess social ability.
- *Adaptive*: The bidding agent should be able to change its behaviour on the basis of previous experience in terms of its previous bids or the bids placed by other bidders.

- *Intelligent*: The bidding agent should be intelligent enough to evaluate all the ongoing auctions of the same item and to choose to join the one which gives maximum surplus (Adomavicius et al. 2009).

2.2 Software Agent Frameworks

A software agent is a computing paradigm that can automate different tasks for bidders operating in an online auction environment. These agents can perform variety of tasks such as analysing current markets to predict future trends, deciding the amount to bid at a particular moment in time, evaluating different auction parameters and monitoring auction progress, along with many more. These negotiating agents outperform their human counterparts because of the systematic approach they take to the effective handling of complex decision making situations. This creates more opportunities for expert bidders and sellers to maximise their profit and contentment. Software agents make decisions on behalf of consumers and seek to ensure items are delivered to buyer preferences. For this to work well, these agents must have prior knowledge of both the certain and uncertain characteristics of the auction. Software agents can represent expert bidders or sellers (auctioneers) to fulfil their requirements and beliefs, and consequently are trained to achieve these aims. As agents in the auction, they can accept bids from a large number of bidders, sort the bids out, allocate goods dynamically among bidders according to pre-defined priority rules (e.g. about price, bid quantity and time of first arrival on a website) and post results instantaneously on the Web.

Much research and many studies have been conducted regarding the design of bidding agents for online auctions. The most commonly used bidding agent architectures are based on Belief-Desire-Intention (BDI) (Georgeff & Ingrand 1989; Rao & Georgeff 1991), decision-theoretic (Byde 2002; Preist et al. 2001), game-theoretic (Mackie-Mason et al. 2004; Vorobeychik 2011) and data mining frameworks (Symeonidis & Mitkas 2005; Dimou et al. 2007).

2.2.1 BDI Frameworks

BDI models are well-recognised in AI literature for the designing of rational agents (Georgeff & Ingrand 1989; Rao & Georgeff 1991). BDI models use the beliefs (B) desire (D) and intention (I) concepts of agents to design bidding agents that solve a particular problem. These agents need to respond rationally to unpredictable events in a dynamic market. BDI agent architecture is used in online trading to judge the reputations of participating traders (Carbo et al. 2001; Pinyol et al. 2010; Pinyol et al. 2008).

Along these lines, Chalmers et al. (2002) have used BDI agents to explore possible virtual organisations or markets. Their agents were capable of choosing the best virtual organisation to meet customer requirements. BDI frameworks have also been employed to design rational bidding agents for online trading. In an attempt to design goal-driven bidding architecture, Jong Yih and Lee (2004) presented a BDI-based formal framework to model the mental structure of agents. This agent has been formed from three models: a goal model of possible goals and events in response to these goals; a belief model about the external environment, the internal agent's state and possible bidding strategies; and a plan model describing a series of tactics for designing bidding strategies.

This agent's bidding decisions were based on the user's reservation price, the time remaining to acquire an item, the current offer at each individual auction and a set of tactics and strategies. However, bidding decisions also depend on the number of bidders in an auction and that auction's bid rate. Sidnal and Manvi (2012) have considered these factors, proposing a BDI agent framework that incorporates human intelligence and a human (cognitive) perspective for use by bidding agents in concurrent English auctions within mobile e-commerce (Figure 2.3). This BDI agent computes amounts and places bids against risk averse and risk neutral bidders. It also learns through simulation to bid, sleep or withdraw. The authors considered the bidder's desires when designing its belief sets which include relevant auction-related information such as its current maximum bid value, the number of bidders and the start time. The agent's intention module is responsible for opting withdrawal conditions, computing payoffs and creating clones. It contains an inference engine which identifies appropriate intentions that should be executed.

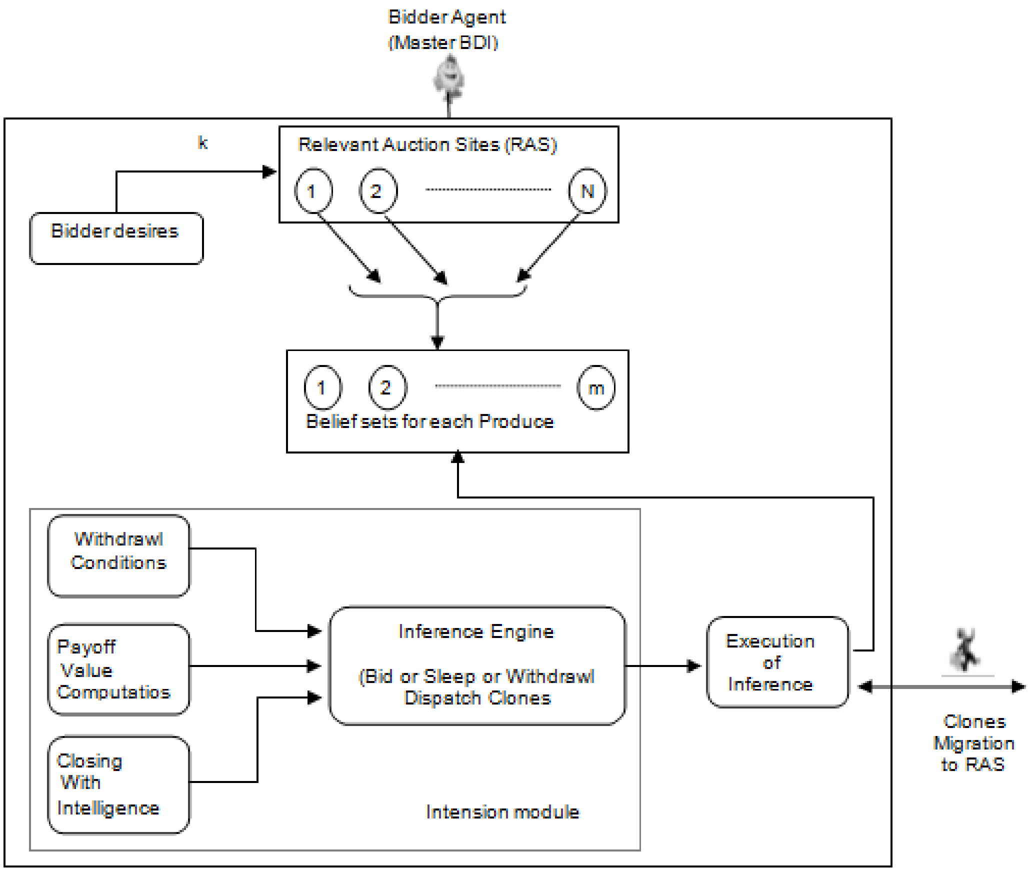


Figure 2.3 BDI based bidder agent (Sidnal & Manvi 2012)

The basic architectural requirements for an intelligent negotiating agent are given by Fasli (2007). His agent consists of three main components that cover communication, decision-making and the knowledge base (Figure 2.4). Different agents communicate with each other by exchanging messages, i.e. communicative acts or 'performatives'. The communication component receives and interprets the incoming messages and extracts the proposals. A proposal may be an offer, acceptance or rejection of a previous proposal, or a termination message. The communication component also generates messages to express the agent's proposals to other agents.

The internal knowledge-base component is the information that an agent has about itself and its environment. This knowledge base has three models: the environment model, the self model and the opponent model. Once a message has been processed and bids extracted, this information is stored in the negotiation history and then also passed on to the decision-making component. The decision-making component is responsible for evaluating current proposals, and it generates decisions accordingly.

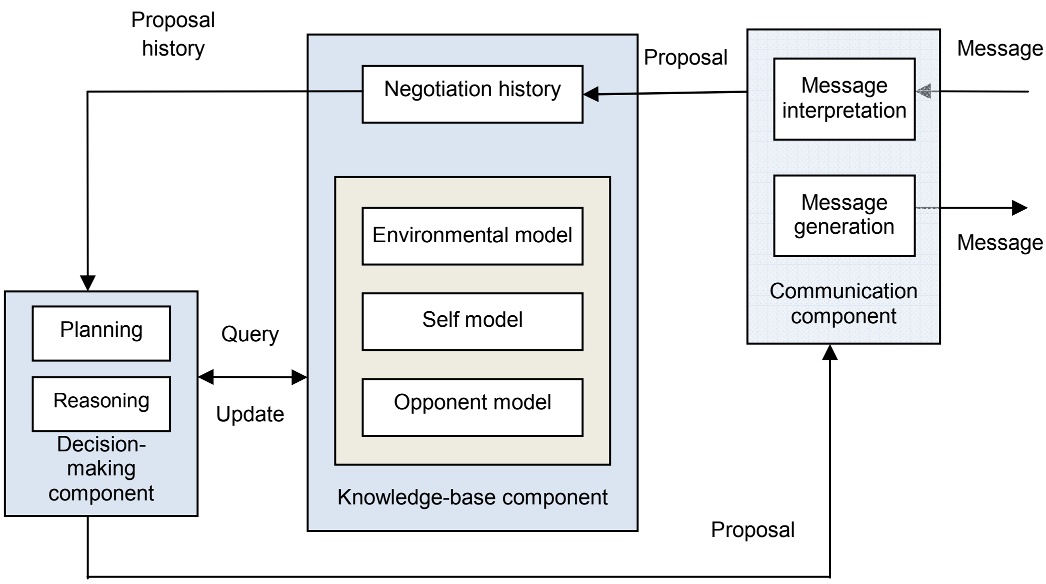


Figure 2.4 Abstract architecture for negotiating agents (Fasli 2007)

To support the development of intelligent agents based on the BDI model, a reasoning engine called Jadex has been developed (Braubach & Pokahr 2009).

Researchers can design their agents in XML and Java for different middleware such as JADE using this software system.

2.2.2 Decision-theoretic Frameworks

Decision-theoretic frameworks are the most commonly used frameworks in the literature on designing bidding agents for e-markets. Byde (2002) has explored three different algorithms—GREEDY, HISTORIAN and Dynamic Programming (DP) —for agents participating in a sequence of overlapping English auctions. The agents' success was measured in a variety of situations such as in the presence of a small or large number of opponents who valued the auctioned items differently and could randomly choose any of these three algorithms. This study demonstrated the robustness of the DP approach, which outperformed its GREEDY counterpart and has reasoning component built on probability distributions that can only be known approximately.

Dynamic programming has since been combined fruitfully with belief-based algorithms. Preist et al. (2001) used this combination to develop a coordination algorithm for effective bidding in ongoing English auctions that all close at the same time. This approach also addressed the question of whether it was appropriate to make a purchase in an auction which was about to close. In contrast to the above-mentioned work, which only considers English auctions, Schmid and Paulussen (2005) have developed a flexible and dynamic programming-based decision-making framework using Markov Decision Processes (MDPs), to support agents to participate in multiple auctions. This framework can handle different auction protocols (English, Dutch, First-price sealed-bid and Vickrey) with sequential and overlapping auctions.

A dynamic programming-based proxy agent—Proxy Agent for Combinatorial Auctions (PRACA)—has also been proposed as a way to bid on a bidder's behalf in online combinatorial auctions (Chakraborty et al. 2009). This agent uses a distributed scheme in its main process and coordinates the activities of several proxy agents through signals. Human bidders do not need to monitor bids continuously because the proxy agents update them.

Further, many researchers have drawn on the observed selling price of items in past auctions to estimate an auction's closing price and design effective bidding agents. To cite a sample of this work: Stone et al. (2002) developed an ATTac-2001 agent—a top-scoring agent in the second Trading Agent Competition (TAC-01)— which uses the boosting technique of price dynamics learned from past auction data to calculate optimal bids. Another study assumed the values of goods from the selling price distributions of past auctions (Greenwald & Boyan 2004). The authors used their findings to design bidding agents for three auction environments: sequential auctions, simultaneous auctions, and the Trading Agent Competition (TAC), a hybrid of sequential and simultaneous auctions. These agents have been developed to compete in the TAC competetion platform, an environment to develop autonomous agents for competing against each other in multiple and simultaneous auctions for different interrelated items. However, this envirnment restricts the development of agents for a set of specific types of auction formats such as Continuous one-sided auctions, English multi-unit auctions and Continuous double auctions. Therefore, the agents for other auction formats such as Standard English auctions, Sealed bid auctions, Dutch auction, are developed to compete in live auction envirnments or in the simulated environments. In this thesis we have designed a bidding agent for Standard English auctions using a simulated environment.

Anthony et al. (2001) have presented an approach similar to the one in this thesis; they used a polynomial function to make complex bidding decisions about monitoring multiple auction houses, picking which auction to join, and making the right bid to ensure an item is bought under conditions matching preferences. The autonomous agent designed in their study can participate in auctions with multiple protocols. It can make decisions based on multiple tactics that each deal with a single facet of its reasoning. These tactics are then combined to give the agent's overall view at a given moment in time.

The history of end-prices has also been exploited to design a probabilistic bidding agent that can compete in several ongoing single-unit auctions (Dumas et al. 2005). The authors developed a prediction method to estimate the probability of winning an auction with a given bid. It is coupled with a planning algorithm that determines where and how much to bid. The success of their approach is borne out in increasing user payoffs and the welfare of the market.

Contrasting with the above-mentioned bidding agents, which support a set of simple, predefined bidding strategies, an agent-based online auction system has been set out by Ford et al. (2010). The system consists of an auction house and various auction (bidding) agents, each of whom represents an auction. This system supports the specification of complex bidding strategies to these autonomous bidding agents. The directory facilitator (DF) assists bidders (users) to search for an auction agent to bid according to their preferences (Figure 2.5). The authors use a model-based approach to specify layered bidding strategies for the autonomous bidding agents. This layered architecture assists bidders to mix and match various strategies to create their own complex variants. It also supports the reuse of simple and more sophisticated bidding strategies.

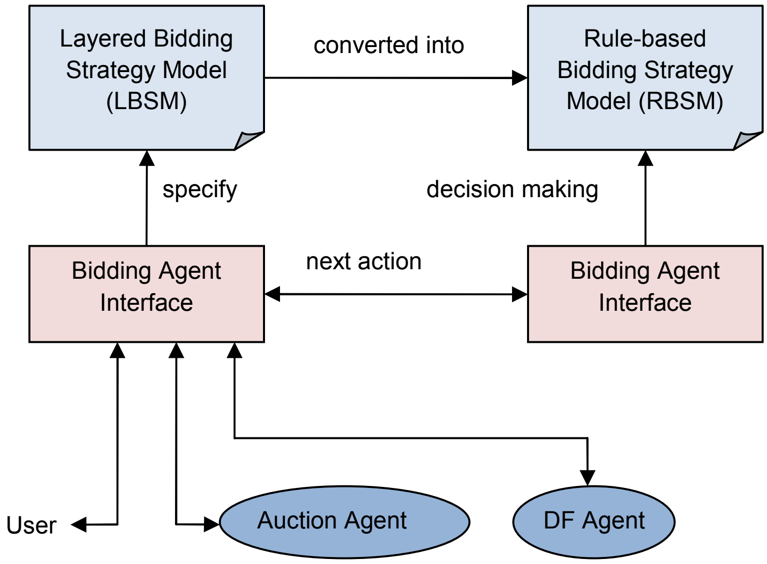


Figure 2.5: Layered bidding agent architecture (Ford et al. 2010)

Furthermore, the presence of intelligent agents affects the marketplace in terms of closing prices and chances of winning. Sow et al. (2010) explored the economic consequences of populating a simulated English auction market by intelligent agents; they were joined by three groups of standard human bidders with different risk preferences. The authors found that the software agents won auctions at a lower price and achieved higher average utility than the standard human bidders. They emerged as the dominant players in the marketplace since they won more auctions.

2.2.3 Game-theoretic Frameworks

Many economics researchers follow a game-theoretic approach, which assumes that all agents share common knowledge of their rationality. In the realm of auctions, this approach computes the Bayes-Nash equilibrium of the auction game where bidding agents play an equilibrium bidding strategy. Mackie-Mason et al.

(2004), for instance, used an empirical game-theoretic approach to reduce the risk that bidding agents would only succeed in meeting some of their requirements when bidding in many separately allocated time slots. To do this, their study relied on different price prediction strategies. The same study found that using final price predictions to design bidding agents significantly improves bidders' performance in a market-based scheduling environment.

In another valuable insight, Vorobeychik (2011) made use of simulation-based game theory to design a winning bidding agent in a TAC/AA ad auction game. (TAC/AA ad auction game is a research forum for the developers of bidding agents for keyword auctions.) Working in a restricted bidding strategy space of discretised linear strategies, the author approximated equilibria, proposed equilibrium selection techniques and evaluated the robustness of various possible strategies. He concluded that in spite of their approximate nature, the equilibria gave very valuable predictions about the actual ad auction tournament bidding

2.2.4 Data Mining-based Frameworks

Bidders in online auctions face the difficult task of deciding the amount to bid to purchase a desired item in line with their preferences. This bid amount can, however, be forecasted effectively by analysing the data produced as an auction progresses (historical data) using data mining techniques (Jank & Shmueli 2005; Kehagias & Mitkas 2007; Min & Qiwan 2009; Nikolaidou & Mitkas 2009). A similar approach has been used in this thesis for bid forecasting to assess the value of an item in the target auction. This forecast bid can also be exploited by bidding agents to improve their behaviours. This improved bidding mechanism of bidding agents results in a reduced number of negotiations between traders i.e. m instead of n, $m<n$ (Figure 2.6) (Symeonidis & Mitkas 2005).

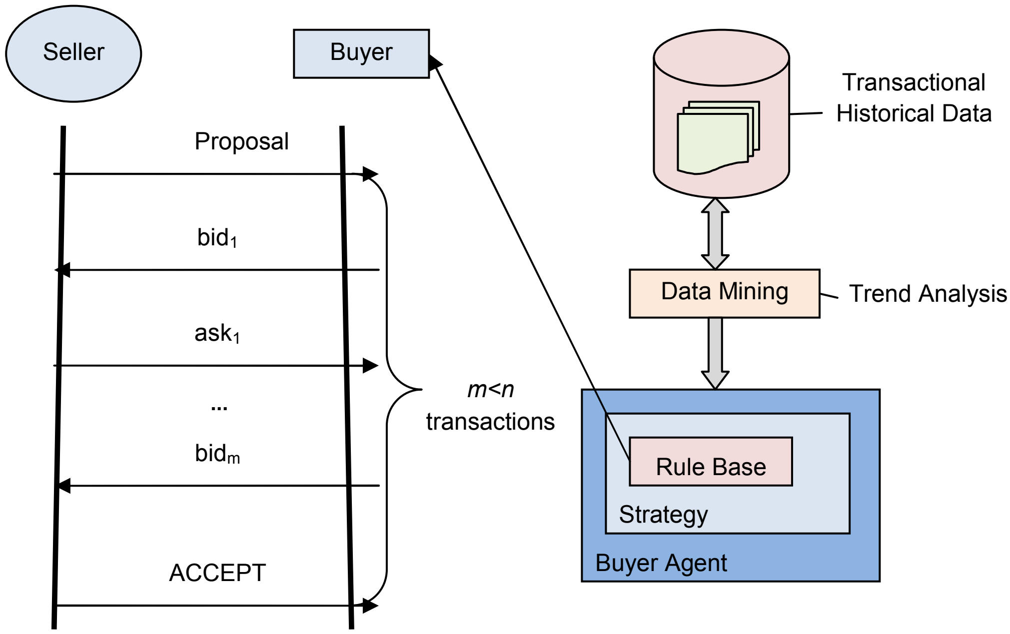


Figure 2.6. Improving the behaviour of bidding agents (Symeonidis & Mitkas 2005)

Supporting these advances, the Agent Academy (AA) has been developed to diffuse knowledge extracted from data mining (DM) techniques to agents and so enhance their intelligence (Dimou et al. 2007). AA is an open-source integrated framework for constructing multi-agent applications as well as supplying DM-extracted knowledge. It is a multi-agent system based on the GAIA methodology and has been implemented on the JAVA and WEKA APIs (Dimou et al. 2007). Knowledge can be diffused into the MAS at three different levels of its architecture depending upon the type of knowledge extracted; these levels are at the application, behavioural and social-architectural stages of the MAS system (Figure 2.7) (Symeonidis & Mitkas 2005). At the application level, data mining is performed on the available application data in order to discover useful rules and patterns. The extracted knowledge relates to the scope of the end-user application rather than to its internal architecture. At the behavioural level, data mining is done on log files containing agent behaviour data in order to predict future actions and eliminate redundant agent activity. The extracted knowledge results in more efficient agent actions. At the evolutionary stage, data mining is performed on agent communities to study agents' social issues. Knowledge is extracted from the interactions between different agents.

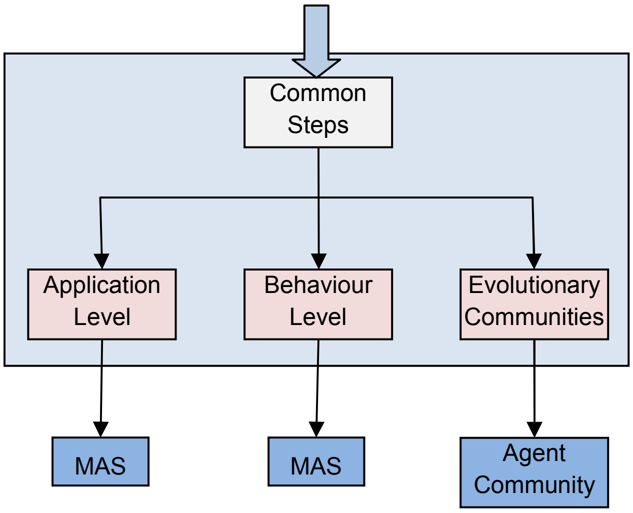


Figure 2.7. The unified MAS methodology (Symeonidis & Mitkas 2005)

2.3 Price Prediction Approaches

Predicting the end-price of an auction before its close has become an increasingly important area of research since the availability of this information can greatly benefit buyers and sellers (Raykhel & Ventura 2008; Zhang et al. 2010). Ongoing advances in artificial intelligence and machine learning techniques have led researchers to employ these methods in online auction price prediction. The methods which have been applied to price forecasting are categorised and illustrated in the sections below:

2.3.1 Neural Networks

Neural networks (NN) are based on a biological processing mechanism which is a simulation of the human neuron system. This simulation performs nonlinear mapping from a set of inputs to a set of outputs using artificial neurons (Beltratti et al. 1996). These neurons are interconnected to form a network which establishes a nonlinear and complex relationship among the input and output variables, and this is used to classify and predict the input data. The use of NN to build forecasting models has been popular since the late 1980s. Some of the earliest research to predict and decode nonlinear regularities in stock price movements using neural networks dates back to 1988 (White 1988). Since that time, many learning algorithms and prediction methods in stock market literature have used NN in their design (Cao et al. 2005; Grudnitski & Osburn 1993; Kimoto et al. 1990; Kohzadi et al. 1996; Lin et al. 2009; Pai & Lin 2005; Tsang et al. 2007; Wang et al. 2011; Yoon & Swales 1991).

Artificial neural network techniques have also been used by researchers to predict the closing price of online auctions in particular. To this end, Kim and Baek (2004) have proposed a model for forecasting winning bid prices to solve an information asymmetry problem in online auctions. The results demonstrate that neural network techniques outperform traditional statistical techniques. Elsewhere, Ghani and Simmons (2004) has used an NN-based multiclass classification and binary classification to build a price-prediction model with greater accuracy than decision tree-based classifications (Ghani & Simmons 2004). Other studies have employed back propagation neural network (BPN) techniques along with traditional statistical approaches to forecast the final prices of auctioned goods (Xuefeng et al. 2006). Chan (2008) applied BPN to predict the final price of eBay auctions. His study developed its own intelligent spider to retrieve online information before actually predicting the final price of auctions. Their customer segmentation approach helped users to choose a bidding policy conducive to placing winning bids. To enhance prediction practices further, Jenamani et al. (2011) combined the neural network method of price forecasting with other methods; these included techniques for removing outliers' support vector data descriptions, selecting the right features using mutual information and clustering datasets. More recent work by Lin et al. (2013) has proposed a price prediction model for English auctions aiming to capture the complicated relationships among key auction features. The authors used regression, neural networks and neuro-fuzzy approaches to construct their price prediction model. The empirical results suggest that the neuro-fuzzy system is most sensitive in predicting the nonlinear relationship among these auction variables.

2.3.2 Fuzzy Logic

Fuzzy systems have recently been introduced in this field to handle the imprecise nature of price forecasting. Fuzzy set theory incorporates a simple IF-THEN-rules approach, which is used to tackle uncertain auction parameters and solve forecasting problems. He and Leung (2001) have presented a fuzzy rules-based bidding strategy (FL-strategy) for the forecasting of optimal bids/asks for bidding agents in continuous double auctions. Their study compared this strategy with ZI-, FM-, CP- and Gjerstad-Dickhaut strategies, concluding that the fuzzy rules and fuzzy reasoning-based FL-strategy outperformed other bidding strategies. In another successful application of fuzzy set theory to agent-mediated e-commerce, He and Jennings (2004) undertook the design of a Southampton TAC bidding agent. Their agent successfully participated in trading agent competitions in 2001 and 2002 employing a range of fuzzy rules. It used fuzzy pattern recognition to design an adaptive bidding strategy that changed according to the type of environment where it was situated. Fuzzy reasoning techniques were also invoked to predict hotel closing prices given the prevailing market conditions.

Goyal et al. (2005) proposed a model of an attitude-based bidding agent which could participate in single auction using fuzzy theory to quantify bidders' attitudes. Fuzzy logic was incorporated in the bidding agent's decision-making process in the form of attitude-based characteristics, and this made it more competent. Along the same lines, He et al. (2006) developed a bidding agent that could participate in multiple overlapping English auctions. Their study used neuro fuzzy techniques to predict the closing prices of auctions (expected closing price); at the same time, they introduced an Earliest Closest First (algorithm) to adapt the agent's bidding strategy to the type of environment where it was situated. The authors benchmarked the ECF algorithm against other much-favoured methods in the literature and concluded that their bidding agent performed best. Yu and Liu (2008) designed a similar bidding agent to participate in multiple overlapped auctions, however in a significant turn, this agent was also capable of making trade-offs in its bidding behavior between different attributes that characterise the desired good with the aim of maximising user satisfaction. This study further demonstrated this bidding agent's effectiveness across in a wide range of settings.

Along with attributes of the good, Ma and Goyal (2010) also assessed the bidder's attitude (eagerness) in their design of a fuzzy attitude-based bidding strategy (FA-Bid). Their approach used overall assessment to set a price range based on a current bid, which was then used to select the best possible price to bid. They found that agents with an attitude-based strategy behave more flexibly and efficiently than those who only consider 'good' attributes and ignore the bidder's attitude.

User-generated feedback reviews also affect the winning price of an auction. Most of the literature adopts a hedonic regression model to examine this research area (Ba & Pavlou 2002; Houser & Wooders 2006; Lucking Reiley et al. 2007; Resnick et al. 2006). Recently, Zhang et al. (2012) have used fuzzy logic along with a hedonic model to thoroughly examine how buyers mentally interpret a seller's reviews and then adjust their bids accordingly. The authors took a hedonic regression approach to select key variables, which were exploited with a fuzzy logic expert system (FLES). They found relationships among variables such as the item characteristics, auction characteristics, review scores and the final winning bid. The authors held that FLES was successful in presenting nonlinear relationships interactions and individual predictor behaviour, and gave a chance to explore what drives the relationships to be interdependent and what causes them .

2.3.3 Grey System Theory

Grey system theory was introduced in 1982 by Ju-Long as a means to work efficiently with systems where there is incomplete information. The theory does this through the effective description and monitoring of these system's operating behaviour (Ju-Long 1982). Since its introduction, grey system theory has been successfully applied to analyse uncertainty in systems across the fields of management, education, environmental sciences and the social sciences (Liu & Lin 2005). More recently Lim et al. (Lim et al. 2007) have explored the potential application of the theory to predict the closing price of an online auction. The authors found that the prediction accuracy exceeded 90% when grey theory was used for price forecasting.

This study also compared the prediction accuracy of grey theory methodology and has been evaluated against time series methods for forecasting closing price of an auction. The grey theory methodology outperformed other approaches in situations when data was less readily available. Similar results arose from a subsequent comparison of grey theory prediction agent with an artificial neural network (ANN) agent (Lim et al. 2008). The ANN agent was successful, however when a lot of information about auction parameters was available.

2.3.4 Functional Data Analysis

Functional data analysis (FDA) is a statistical method which analyses models and predicts time series data. It has been applied successfully by researchers in the fields of medicine, biomedicine, biology, environmental science, ecology, engineering and economics since 1995 (Ullah & Finch 2013). Nevertheless, it received little attention from online auctions researchers until the mid-2000s.

Jank and Shmueli (2005) used FDA as an innovative way to model and analyse the price dynamics of auction data. Their study detected differences in the price dynamics of auctions even when they concerned comparable goods. Based on these dynamics, they identified three types of auctions: experienced seller/buyer auctions where there are moderately high opening prices and low competition, yet high closing prices are achieved; 'greedy-seller' auctions, which have the highest average opening prices and the lowest closing prices; and bazaar auctions which have the lowest opening prices, the highest level of competition and the highest closing prices. These price dynamics were first used by Wang et al. (2008) to design a dynamic forecasting model for prices in online auctions. Their forecasting model was capable of effectively representing unevenly spaced bids, estimating price dynamics and integrating them into the price forecaster, while also incorporating both static and time-varying information about the auction into the forecasting system. The forecasting model outperformed standard forecasting methods, such as double exponential smoothing which do not take account of dramatic changes in auction dynamics. Dass et al. (2010) extended this work, presenting an innovative and dynamic forecasting model for simultaneous online auctions (SOA) where high-end art items are sold for prices ranging from a few thousand to a few million dollars. Their research expanded the usefulness of the model beyond eBay auctions and showed the strength and flexibility of this dynamic price forecaster. The authors highlighted the capacity of this auction price dynamics-based model to capture bidder competition within and across auctions. A subsequent study (Dass et al. 2011) developed a similar intelligent bidding system called SOABER (SOA BiddER system) that uses dynamic forecasting to assist SOA bidders to maximise their surplus and optimise their bidding strategies.

In an effort to combine functional and non-functional distances, Zhang et al. (2010) proposed a functional K-nearest neighbor (fKNN) forecaster for the real-time forecasting of online auctions. This forecaster uses static and time-varying features, as well as information about the auction's price dynamics. The price dynamics information is captured via a functional representation of the auction's price path. The authors noted the proficiency of the fKNN for forecasting heterogeneous auction populations.

2.3.5 Case-based Reasoning

Case-based reasoning (CBR) is a problem-solving paradigm which uses previous experiences to solve new problems by integrating problem solving, learning and understanding with memory processes (Kolodner 1992). CBR has been used successfully in industrial, healthcare, finance and environmental surveillance applications (Montani & Jain 2010). There are, however, few works on online auctions that make use of this technique. Kaushik and Tagra (2005) have proposed a bidding agent which acquires learning capability when it participates in online auctions. The bidding agent stores past histories of similar auctions along with their solutions, which helps it to learn from past experience. Their findings showed that the CBR-agent performed better than other participating agents in terms of success percentage and average winning price.

Brzostowski and Kowalczyk (2005) presented a possibilistic CBR method to select partners for multi-attribute negotiation on the basis of their behaviour in previous negotiations. A possibility distribution was used to calculate the chances of successful agreement with a given partner in a form of expected utility. However, this method of calculating expected utility based on discretization was found to grow exponentially in computational complexity with the number of attributes; it threatened to become infeasible when a high number of these attributes is involved. To overcome this problem, the authors proposed an improved algorithm that exhibited linear computational complexity to explore the shapes of α-cuts of the possibility distribution obtained through case-based reasoning (Brzostowski & Kowalczyk 2012). This algorithm can make decisions about potential offers during negotiations and discover the highest number of prospective partners for multi-attribute negotiations. In another move to simplify the multi-agent frameworks used in e-commerce, Jain and Dahiya (2012) also proposed a multi-agent knowledge management system based on case-based reasoning methodology. This system aims to help agents with their decision-making during negotiations.

2.3.6 Classification and Regression

Of all the intelligence techniques, classification and regression have been the most widely used for price forecasting in online auctions since 2000. Regression techniques assist researchers with specific outputs while classification methods create discrete categories of results in terms of range outputs. Ghani and Simmons

(2004) have predicted the closing price of eBay English auctions using both regression and classification machine learning techniques. The authors employed linear regression, polynomial regression with degree 2 and 3, along with a classification and regression tree (CART) approach to continuous price prediction. The discrete price range was forecasted through multi-class classification and multiple binary classification tasks in which decision trees and neural networks were exploited to construct classifiers. The authors found that the multiple binary classification technique was best-suited to price prediction.

Kehagias et al. (2005) have also developed a short-term dynamic price forecasting model using an AR approach which employs on outstanding price offers from past English auctions. This forecasting model can be used to improve the bidding behaviour of autonomous agents when incorporated within their reasoning mechanism. Notably, all these prediction models have used linear models without adaptivity. In an effort to improve these methods, Kehagias and Mitkas (2005) applied AR and autoregressive-moving average (ARMA) models as linear adaptive filters to estimate the price histories revealed in open outcry auctions such as English and Dutch auctions. Their study exploited a genetic algorithm to deal with the optimization problem when calculating the parameters of these two filters; the performance of these models was measured using different metrics. According to the empirical results, the performance of the ARMA model improved as it approached the last sample in a sequence of values.

In a series of similar work, Xuefeng et al. (2006) used multivariate regression (MR) to predict continuous price value and applied multiclass classification to the discrete price prediction of eBay English auctions. The multiclass classification was implemented using logistic regression and neural networks. In contrast with previous prediction models, however, these authors also sought out a clustering effect in the prediction results. Further, they found that prediction accuracy was improved for neural network models if data clustering was used. A clustering-based price prediction framework was introduced by Kehagias and Mitkas (2007) in a study that aimed to select the eBay auction where a traded item was sold at the lowest price. The authors used a *k-means* clustering technique to classify auctions into discrete clusters. This technique operates dynamically with multiple auctions since bid prices change in ongoing auctions. They applied a probability-based decision-making process to select the auction with the lowest price. The empirical results demonstrated the robustness of the designed system for dealing with multiple on-line auctions about which there is little or no available information.

Seeking similarly to advance eBay end-price prediction, Van Heijst et al. (2008) applied a boosting method for the first time in this domain. In boosting, multiple models are combined to sharpen performance by reducing errors from unseen data. The authors presented a decision support system which used closed auctions for an item and then combined decision trees to predict its end-price in current auctions. Machine learning algorithms were also combined by Nikolaidou and Mitkas (2009) who set out to forecast the bidding strategy of trading agents in the context of the Reverse TAC (or CAT) game of the Trading Agent Competition. Their prediction was based on the time series of traders' past bids using a similar approach to the one in the van Heijst et al. (2008) study. The authors demonstrated that the boosting methodology, by combining Bagging and J48 decision tree techniques, outperforms other native prediction methods. Hirose et al. (2012) took a similar approach for improving prediction accuracy by combining *k-NN* regression and multiple linear regression methods. These authors put forward a method NNRMLR to predict prices at used car auctions; according to this approach, distances of *k-NN* are evaluated using the multiple linear regression technique. They found dramatic improvements over traditional approaches. Lastly, Grossman (2013) applied a combination of machine learning approaches to predict the final sales price of a sports autograph in eBay auctions. He deployed *k-NN* to eliminate outliers in the CART algorithm when predicting the final sale price.

2.4 Bidding Strategies in Online Auctions

Traders need to make difficult decisions about how they will act as bidders when they compete over an item in an auction. A bidding strategy consists of a set of bidding activities and conditions (Ford et al. 2010). When certain conditions become true during an auction, appropriate bidding activities can be performed. For buyers, the bidding activities may be about deciding which auction to participate in, the appropriate price to bid and when to take part in bidding. For sellers, on the other hand, bidding activities may concern when to put their goods up for sale.

Both buyers and sellers try to make decisions to achieve the maximum surplus (Jiang & Leyton-Brown 2007). This thesis focuses, however, on the bidding strategies that bidders follow when engaged in online auctions. When a buyer looks for an item to bid for online, a simple search may return about 100 hits. Buyers face decisions about when to bid, which auction to bid in and how much to bid. They may opt to bid in one auction not realising that another auction is offering the same item at lower price. Apart from selecting a auction to enter, they need to decide the right amount to bid so that they do not pay too much and fall prey to the winner's curse.

The prediction of end-prices in ongoing auctions helps buyers to make these decisions (Jank & Shmueli 2009; Kehagias et al. 2005; Lim et al. 2011). If the end-price of an online auction is known, then bidders can settle on a bidding strategy to win the auction if the price is appropriate to the item’s value, and experienced bidders can win auctions at lower cost (Raykhel & Ventura 2008; Sun 2005). In this thesis, the predicted closing price of an auction is used to create successful bidding strategies for bidders with different behaviours. It is, thus, essential to understand these bidding behaviours so that we can design strategies that will give these bidders maximum surplus.

2.4.1 Understanding the Bidding Behaviours of Buyers

To design bidding strategies for buyers, it is essential to have some understanding of the factors that drive bidders' behaviour. A key idea here is that different online auction attributes and external market information are important. These attributes may be static or dynamic. Static attributes may include information about the type of auction, the opening bid, the seller's reputation, the item itself and the auction's duration, etc. Dynamic attributes may be about competition on the market, the number of bidders, etc. External market information may include details about participating traders, the availability of the goods and their expected re-sale value, the historical situation vis-à-vis prices and participants' behaviour, etc.

Competition is regarded as an extremely important variable when determining bidding behaviours (Johns & Zaichkowsky 2003). Therefore it would be highly beneficial to both sellers and buyers to see how the number of bidders affects other bidding variables such as the bid range, the number of bids placed, bid increments, bidding strategies and the values of both the average and winning bids. Sellers may be able to manipulate the auction format in order to reap the benefits of increased competition (Haubl & Popkowski-Leszczyc 2000). Bidders, on the other hand, can adjust their bids according to their bidding behaviour.

Brannman et al. (1987) found correlations between the number of bidders present and behaviours related to the average value of bids in various English auctions. These studies were conducted based on auctions of commodities such as oil and rice, and they predominantly involved bidders who placed bids on behalf of a firm, resulting in fewer directly involved consumers. None of these studies looked at similar behaviours around more typical consumer-oriented products such as household furniture, art or antiques. With a different audience, these types of auctions might involve different bidding behaviour patterns, with either more or less dynamic inter-bidder relation. Individual bidders might exhibit higher levels of involvement, as they make purchases on their own behalf rather than for a firm or larger corporation (Johns & Zaichkowsky 2003). The findings about bidding behaviour could also help to bridge the informational gap between the auctioneer and the bidder. They could result in an increase in informed bidders who make more rational bidding decisions.

Bidders also value the information they glean from observing the bids of others when deciding on their own bidding behaviour (Schindler & Schindler 2011). In English auctions, ascending bids are open to all potential bidders, and the number, size and timing of the early bids provide a considerable amount of information that is potentially useful to later bidders. Furthermore, the bidding behaviour of bidders also depends on their own attitude to get an item (Anthony & Jennings 2003). They may be desperate to have the item or they may be intent on acquiring it at a bargain price. A desperate bidder bids aggressively as soon as auction starts in order to maximise his chances of securing the item. On the other hand, a bidder who is seeking a bargain starts bidding in the auction at a very low price and increases his bids by the minimum possible increments. This type of bidder bids the maximum amount that he is willing to pay when the auction is about to close.

In this thesis, the bidding behaviour of bidders is understood to depend on the auction's attributes, other bids in the auction and the bidder's own attitude for getting the item.

2.4.2 Types of Bidding Behaviour of Buyers

Researchers often rely on common assumptions when analysing the bidding behaviours by buyers. For example, Shang et al. (2006) suppose that each bidder is homogeneous and can be rational and risk-neutraland he clearly knows the value that the auctioned item has to him and do not know other's valuations of it. Bidders tend to bid strategically to achieve their own greatest benefit. Nevertheless, the homogeneity assumption cannot explain all complex behaviours. The low cost of attending an online auction draws a greater variety of participants. At the same time, the homogeneity assumption is sometimes necessary for those wishing to compare different bidding behaviours (Sow et al. 2012).

Bidding behaviours depend upon a variety of inter-behavioural dynamics, which may vary according to the number of bidders involved. Significant relationships have been found between the number of bidders and bid range, starting bid values and winning bids (Johns & Zaichkowsky 2003). Several studies have tried to identify heterogeneity among bidders (Bapna et al. 2000; Bapna et al. 2004; Shah et al. 2003; Sow et al. 2012). Three different types of bidders i.e. evaluators, participators and opportunists were identified during an empirical investigation of multi-unit auctions (Bapna et al. 2000). Evaluators are early bidders who have a clear idea of their own valuation of the goods. They execute a single bid during the early phases of the auctions. Evaluators' bids are significantly higher than the minimum required bid at that time. Participators make a low initial bid equal to the minimum required bid; they progressively monitor the auction and make ascending bids. Opportunists look out for bargains and buy when they see one. They place the minimum bids required just before the auction closes.

In a similar study, the heterogeneity of bidding behaviour was evaluated by analysing the data collected from multi-unit Yankee auctions in three directions; time of entry, time of exit and number of bids (Bapna et al. 2004). The authors identified five types of bidders based on their behaviours: early evaluators, middle evaluators, opportunists, sip-and-dippers and participators. Early evaluators are bidders who place just one bid during the early stages of the auction, possibly reflecting the maximum amount they are willing to pay. Middle evaluators are bidders that place a single maximum bid in the middle of the auction. This may be due to their late arrival or because they simply observe the auction's initial progress and then place their bid based on the bids of others. Opportunists are late bidders; they differ from snipers (who bid in the closing seconds of the auction) because of the existence of a going-going-gone period in Yankee auctions. Sip-and-dippers usually place two bids. One is lodged early on in the auction to establish their time priority and perhaps to assess the competition. They revise their bids only towards the end of the auction. Participators bid throughout the auction and are characterised by early entrances and late exits, representing high levels of involvement. In their data mining analysis, the authors assumed that these types of bidders remain unchanged in auctions of different lengths. They found that opportunists and sip-and-dippers won auctions in significantly higher proportions than participators and evaluators.

The authors of another study, Sow et al. (2012), introduced three distinct groups of standard bidders: risk-averse, risk-neutral and risk-seeking. They also outlined three types of intelligent agents (Greedy agents, Heuristic agents and Sniping agents) to characterise bidding agents based on their attributes. Risk-averse bidders are ready to compromise their profit to reduce the risk or uncertainty of losing an auction. Faced with the same outlook, a risk-seeking bidder prefers to accept the risk and not surrender any profit. A risk-neutral bidder, in contrast, is unaffected by both the risks that come from the uncertainty and the maximising of profit. A Greedy agent always takes the auction with the lowest current bid as its target auction. Sniping agents hold their bids until the final time stage of the auction. Heuristic agents, in contrast, use tactics based on the amount of time remaining, the remaining auctions, the desire to bargain and desperation to obtain goods. These same Heuristic tactics were explored in a study by (Anthony & Jennings 2003). In addition, Sow et al. (2012) found that intelligent agents outperformed standard bidders, delivering more satisfaction to their owners so long as the correct agents were selected.

Shah et al. (2003) also identified common bidding behaviours by analysing data from eBay video game auctions in a hierarchical process. These authors broadly classified bidder types according to the number of their engagements. An engagement is the set of all bids by an individual bidder in an individual auction. Three variables (time from the end, excess increment and total number of bids) were used to identify an individual bidder's behaviour. Four types of bidders were found from the collected data: evaluators, skeptics, snipers and unmasking bidders. Evaluators bid only once, doing so early and at high value. Skeptical bidders always bid the minimum increment over the current ask price and submit multiple bids. Snipers bid in the closing seconds of the auction. Bidders exhibiting unmasking behaviour conform to four conditions: first, they place multiple bids over a short span of time; second, there are no bids by other bidders between their bids; third, at least one bid in their engagement has an excess increment greater than zero; and finally, at least two non-winning bids are submitted as part of the sequence.

From the research and studies discussed and reviewed in this section, it can be seen that all the bidding behaviors are classified according to two main factors: the number of the bidder's engagements (one or many) and the timing of their bids. In this thesis, the bidding strategies are designed to match bidders' behaviour based on these two factors. These strategies are assessed separately for heterogeneous and homogeneous bidders.

2.5 Methodologies for Designing Bidding Strategies

Several methodologies have been proposed for the design of bidding strategies for online auctions in recent years. These vary from decision-making models to learning methods that support bidding strategies based on a variety of commonly used techniques including game theory, polynomial functions, dynamic programming and computational intelligence-based learning approaches.

2.5.1 Game Theory

Game theoretic technique is one of the earliest approaches taken to the automation of bidding activities during the negotiation process in e-auctions (Preist et al. 2001). Hu and Guan (2005) proposed an agent to automate auction-based negotiations based on a game theoretic approach. They found that this negotiation agent improved the efficiency of a multi-agent automated negotiation system, made buyer agents bid more rationally and determined the price more accurately. Jiang and Leyton-Brown (2007) used a similar approach to designing bidding strategies, also invoking a decision-theoretic model based on the maximum-likelihood estimations of an item's learned valuation distribution. To do this, the authors adopted an Expectation Maximization algorithm. The proposed bidding strategies estimated bidder valuation distributions and the distribution over the true number of bidders more accurately than density estimation techniques. In another study, Pavanje (2012) used game theory to predict optimal bidding strategies for bidders' participation in an auction which gives maximum surplus in the context of simultaneous auctions of the same or similar items.

The game-theoretic approach has been highly popular and is the dominant paradigm in microeconomics. However, this approach is not always useful for solving realistic problems because of its restrictive assumptions when the existence of other bidders affects the bidding outcome of an auction (Faratin et al. 1998; Jiang & Leyton-Brown 2007; Zeng & Sycara 1997). Another important limitation of game theoretic models is that these models are fundamentally static, focusing primarily on outcomes rather than negotiation processes.

2.5.2 Polynomial Functions

Polynomial or decision functions have been proposed as a way to design negotiation strategies that overcome the limitations of game theory. These functions produce offers based on the time available, the resources remaining or an opponent's behaviour. Faratin et al. (1998) have set out decision or polynomial functions that are approximated with a choice of rational bidding strategies. These functions are based on a number of tactics which an agent can use during negotiations, taking into account that these tactics may change over time to reflect various strategic behaviours of participating agents. The polynomial functions are well-suited to practical agent applications, being based on assumptions guided by practical experiences in developing real-world agent applications in the realm of business process management.

These polynomial functions have been used by Anthony and Jennings (2003) to determine maximum bid values at a given time based on multiple tactics: the remaining time tactic, the remaining auctions tactic, the desire to bargain tactic and the desperateness tactic. The authors determined agents' preferences at a particular moment in time by combining these tactics. Their empirical evaluation of these bidding strategies established that the main factors affecting their performance are private valuations, the number of auctions remaining and the time remaining. In separate research, Hou (2004) used these polynomial functions to determine the behaviour of a negotiating agent based on an opponent's previous offers. The author applied a nonlinear regression approach to predict offers and identify the opponent’s tactic type. The study found improvements in the agent's performance were linked to predicting the opponent’s deadline and reservation value, terminating pointless negotiations and avoiding the breakdown of other negotiations.

Polynomial function-based strategies create a large search space and make it difficult to look for an optimal solution. This problem was solved by Gan Kim et al. (2009) using a self-adaptive genetic algorithm which can perform in a flexible and configurable heuristic decision-making framework in a multiple-auction environment. The authors applied both the non-adaptive and the self-adaptive genetic algorithm to enhance the search for an effective strategy. They found that strategies evolved from the self-adaptive genetic algorithm performed better than those from the non-adaptive genetic algorithm in terms of the success rate and average payoff for bids in an online auction marketplace. Sow et al. (2010) used polynomial approach to measure the effect of the presence of these intelligent bidding agents on the closing price of an auction and the chances of winning. They discovered that intelligent agents perform better than standard bidders. These polynomial functions were also used by Naser (2012) in defining the behaviours of agents (sellers or buyers) participating in a negotiation process. A genetic algorithm was employed to determine the best solutions for trading agents, who then reached mutual agreement using a time-bound negotiation framework (TBNF).

2.5.3 Dynamic Programming

Dynamic programming (DP), which is based on the learning of value functions representing cumulative long-term rewards, has previously been used in literature on online auctions with sequential bidding. In these auctions, the bidding agent's ultimate objective is to maximise the cumulative surplus or profit obtained over the entire trading period. Byde (2001) derived algorithms for agents participating in simultaneous auctions for a single privately-valued item using stochastic dynamic programming. These algorithms were designed based on the probability distributions of auction characteristics which had to be learned. Such distributions may not, however, accurately reflect the important dynamics of the auction in which the agent is participating. He addressed this issue in another study in which bidding agents learned the probability distributions of an auction's characteristics by observing that auction (Byde 2002). As we have seen, the author proposed a design comprised of three bidding algorithms: GREEDY, HISTORIAN and DP (Dynamic Programming) for use in sequences of overlapping English auctions. To determine the quality of bidding algorithms, the study measured various parameters of the agents' success. This demonstrated the robustness of the dynamic programming approach, where beliefs about the opposition the agent is likely to face are built up on-the-fly, when compared with the other approaches. Tesauro and Bredin (2002) also formulated agent bidding strategies using real-time dynamic programming for ongoing double auctions. The resulting strategies were successful in optimising cumulative long-term discounted profitability in contrast with traditional strategies such as Gjerstad-Dickhaut (GD), which merely optimise immediate profits. Bertsimas et al. (2009) designed an algorithm for optimal bidding for a single item in an online auction, as well as in simultaneous or overlapping online auctions. The proposed algorithm uses the value functions of single auctions found by dynamic programming in an integer programming framework. The authors found that this new approach produces high-quality solutions both fast and reliably.

Dynamic programming has also been used frequently in keyword auction literature. Fruchter and Dou (2005) employed it, for example, to find a dynamic strategy for allocating a limited budget between a generic web portal and a specialised web portal using keyword activated banner ads. These authors demonstrated empirically that in the long run, an advertiser must always spend more ad money at a specialised portal. Brogs et al. (2007) used a similar approach to solve a problem involving online keyword advertising auctions among multiple bidders with limited budgets. Perlich at al. (2012) presented a bid-optimisation approach to bidding on online advertising opportunities at an appropriate price using supervised learning algorithms and second price auction theory. The problem of how to find an optimal dynamic bidding policy for placing online search ads with Google was solved recently by Dayanik and Parlar (2013) using a dynamic programming approach.

2.5.4 Computational Intelligence or Soft Computing

Computational intelligence is an offshoot of artificial intelligence that emphasises heuristic algorithms such as fuzzy systems, neural networks and evolutionary computation. The use of these techniques to design bidding strategies for online auction agents has been popular since the early 2000s. Of all these algorithms, fuzzy logic especially has drawn attention from online auction researchers seeking to cope with the uncertainty of the multi-auction context in a timely fashion and to handle the ambiguities of the current environment.

He and Leung (2001) developed a fuzzy reasoning mechanism FL-strategy, which consists of heuristic fuzzy rules and fuzzy reasoning on the rule base and is used to capture human behaviour to decide a bid in a given state of marketplace Their study demonstrated that FL-strategy outperform other classic strategies (ZI-strategy, FM-strategy, CP-strategy and Gjerstad-Dickhaut strategy) in a CDA (Continuous Double Auctions) market in a range of situations. At the same time, the agents with the FL-strategy were unable to adapt their behaviour to deal with changing environmental conditions. He et al. (2003) therefore extended the FL-strategy and introduced A-FL-agents (adaptive agents) that were able to tailor their strategy to the supply (demand) of the market. Similarly, Minghua et al. (2004) set out to improve the parameter adaption capability of the fuzzy rules in these adaptive A-FL-agents. The FNN-strategy which they developed exploits neuro-fuzzy techniques that can do fuzzy reasoning; it adjusts the parameters of the fuzzy terms through learning while delivering outputs as the auction progress. Their FNN-strategy increased user satisfaction by making trade-offs between the different attributes that characterise the auctioned item. In a series of similar studies, He et al. (2006) arrived at an Earliest Closest First (ECF) algorithm using a comparable neuro-fuzzy technique that guides an agent's bidding behaviour in overlapping English auctions for multiple items with multiple attributes.

Moving away from the A-FL-strategy, FNN-strategy and ECF algorithm, which all focus on short-term adaptability, Ma and Leung (2008) have proposed an adaptive attitude strategy (AA-strategy) which uses both long-term and short-term adaptability. The AA-strategy employs heuristic rules and a reasoning mechanism based on this two-level adaptive bid-determination method. The authors demonstrated the effectiveness of using an eagerness formulation based on two-level adaptive attitudes and heuristic rules when compared with alternatives in the literature.

The above-discussed bidding strategies offer bidders better chances of winning an auction at maximum surplus. Nevertheless, these strategies remain unfriendly to native users as these cannot be modified to meet their personal preferences or needs. In contrast, Ford et al. (2010) have defined a layered bidding strategy model (LBSM) which explicitly gives users the means to adapt an existing bidding strategy. They can also design their own strategies and develop the ones available into more complex versions. This mechanism allows bidding agents to place bids on behalf of a human user in line with that user's bidding requirements. LBSM's layered approach also supports the reuse of bidding strategies, whether these are simple or complex.

An attitude based agent's bidding strategy is presented by Goyal et al. (2008). It uses fuzzy techniques to make decisions about an auction's outcome and alter the agent's bidding strategy in response to different criteria and market conditions. The authors demonstrate that these attitude-based agents outperform non-attitudinal agents in a highly dynamic bidding environment because of their flexible behaviour. At the same time, their research does not consider the role of market competition for the auctioned items, an issue which plays an important role in the decision-making processes of many bidders. This problem has subsequently been resolved by Goyal et al. (2010), who put forward a fuzzy competition and attitude-based bidding strategy (FCA-Bid) for bidding agents in continuous double auctions. Their agents estimated the best transaction price by invoking their attitude and competition on the market for the goods based on a fuzzy reasoning technique.

2.6 Summary

This chapter has reviewed the role of software agents in online auctions along with the different frameworks behind these agent systems. This chapter has also surveyed a range of approaches used to model the end-price of an auction and methodologies for designing various bidding strategies for different bidders. As we have seen, many researchers have applied machine-learning algorithms to designing automated software agents for use in online auctions. Nevertheless, designing bidding strategies for bidders that are tailored to their bidding behaviour remains an active area of research. Moreover, there is a clear need to take account of the auction's diverse price dynamics in order to accurately predict its closing price in an environment of simultaneous auctions for the same or similar items.

Chapter 3

Automated Dynamic Bidding Agent Framework for Online Auctions

This chapter presents a bidding agent framework for simultaneous online auctions of the same or similar items. The remainder of the chapter is organised as follows: Section 3.1 describes the role of agent technology in automating the bidding process in online auctions. Section 3.2 outlines the framework of the automated dynamic bidding agent at a general level before detailing each of its phases. Finally, Section 3.3 summarises the main issues covered in this chapter.

3.1 Bidding Issues in Simultaneous Online Auctions

The process of automating online bidding is crucial in the context of numerous and simultaneous online auctions of the same or similar items, and software agent technology has a very important role to play in this process (Anthony et al. 2001; Dumas et al. 2005). The dynamic and distributed nature of both data and applications in online auctions requires a bidding agent that does more than respond to information requests; it needs to anticipate and adapt intelligently and actively seek ways to support users. Such a bidding agent must assist in coordinating tasks among humans while also managing cooperation among the dispersed players of the online auction marketplace.

Bidders participating in this marketplace strive to choose the most favourable bidding strategies to win an auction based on their bidding behaviour. The existence of many auctions of similar items complicates their task of selecting an auction to enter. These auctions have different attributes, which prove vital in the auction selection process. Criteria need to be developed for selecting the auction which will deliver maximum surplus to the bidder.

Bidders rarely bid their maximum valuation of an item when participating in its auction. This may be because they have trouble understanding that they should bid this maximum value or they may simply struggle to figure out what their ceiling price is. Having the ability to accurately assess the value of the auctioned item would benefit both bidders and sellers. It would help to solve some of the information asymmetry problem and so reduce transaction time and costs. It would also prevent the overpricing of items by setting a reference price for bidders. An item's value strongly depends on the path that its price takes during its auction, or what is known as the price dynamics of the auction. Therefore, we require a methodology that assesses the true value of items based on the price dynamics of their auctions.

Currently the most widely used methods for auction selection and value assessment item are static; they use information available only at the beginning of an auction (Ghani & Simmons 2004; Van Heijst et al. 2008). These models cannot incorporate the price dynamics of auctions for similar items, a concern that only a few studies (Dass et al. 2011; Kehagias et al. 2005; Wang et al. 2008) have pursued. Further, the price dynamics of auctions are different, even when they deal with similar items, and this has a profound influence on the choice of an auction and valuation of the item being auctioned (Jank & Shmueli 2005). We need, thus, to characterise these auctions based on their price dynamics before we can select one to participate in and assess the true value of the goods on offer.

The focus in this thesis is on auctions like eBay, which have fixed deadlines. Bidders in these types of auctions are reluctant to reveal the maximum amount which they are willing to pay until the closing moments of an auction. They are late bidders who use information from previous bids for intelligent bidding (Ockenfels & Roth 2002). They avoid price wars and may be rewarded with higher payoffs (Bajari & Hortacsu 2003a). Late bidding is also justified as a way to confront shill bidding, which tries to raise the auction price using fake bids. However, these closing-moment bids may also get lost in a congested network or never even be placed. Accordingly, there is a need to find methods of auction selection and valuation so that bidders can place single or multiple bids towards closing time rather than only in the very final moments of an auction.

Formulating bidding decisions for bidders remains difficult even after a suitable auction has been selected and we know the true value of the auctioned item. This is because the amount of a bid at any moment in time also depends on other bidding features like the auction's attributes, the bidder's own attitude and other bidders's attitude. Combined with the growth of online trading systems, the complexities of designing bidding strategies for buyers have given rise to the development of a framework for the automated dynamic bidding agent.

3.2 Automated Dynamic Bidding Agent Framework

Based on the characteristics of auctions and bidding behaviours of bidders, this thesis presents the novel framework of an Automated Dynamic Bidding Agent (ADBA) for price forecasting at a particular moment in time. This framework comprises two phases: phase 1 for initial price estimation and phase 2 for final price prediction (Figure 3.1).

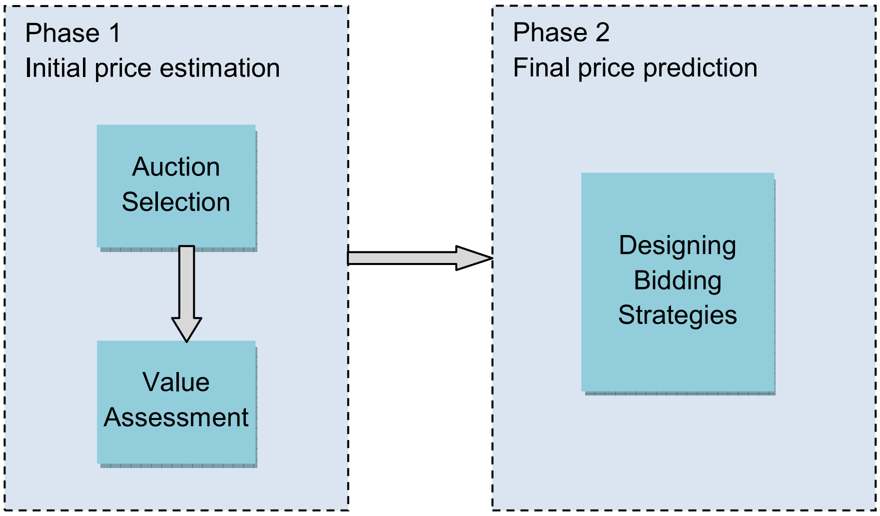


Figure 3.1 Automated Dynamic Bidding Agent architecture

Phase 1 selects an auction to participate in and assesses the value of the auctioned item based on a clustering approach combined with a bid mapping and selection technique. The purpose of this phase is to use the price dynamics of different auctions both to choose a target auction to compete in and to assess the value of the auctioned item. This assessed value serves as the initial price during the second phase of the bidding agent framework.

Phase 2 predicts the final price of the auction by designing bidding strategies that exploit negotiation decision functions and fuzzy reasoning techniques. The purpose of this phase is to use the output of phase 1 and bidding-specific information to forecast the bid amount at a particular moment in time in the target auction based on different bidding strategies of the bidders.

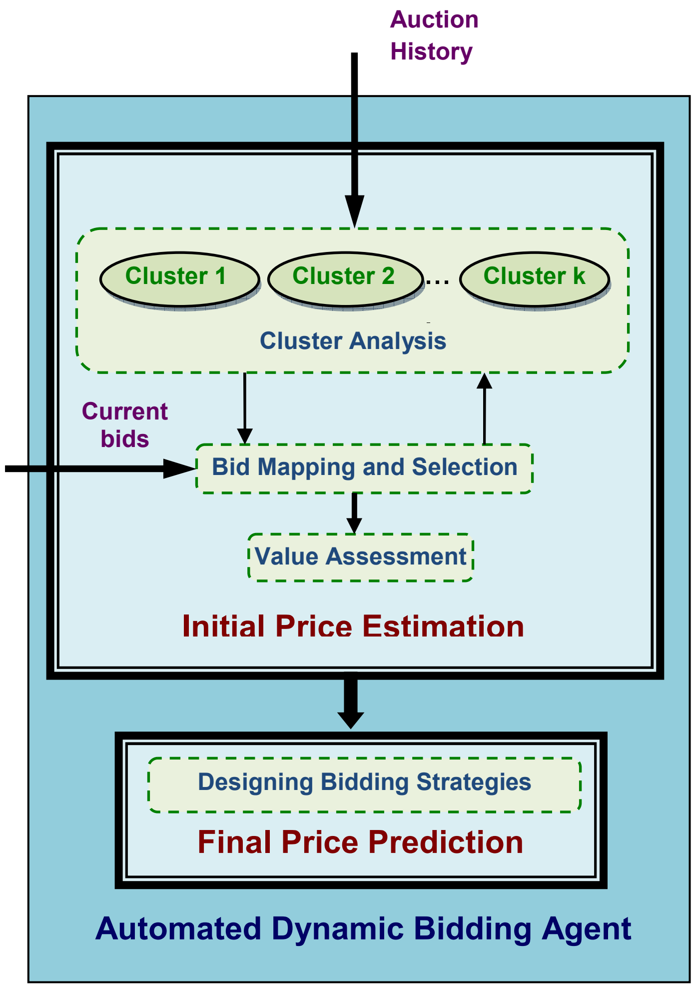


Figure 3.2 Automated Dynamic Bidding Agent framework

3.2.1 Phase 1: Initial Price Estimation

The first phase of the ADBA framework concerns the initial price estimation of an auction. It consists formally of three steps: cluster analysis, bid mapping and selection, and value assessment (see Figure 3.2). The first step aims to cluster similar auctions together in k groups based on their price dynamics. In the second step, the bid mapping and selection component nominates the cluster for each ongoing auction. A target auction is selected for participation based on the transformed data after clustering and the characteristics of current auctions. The last step applies machine learning algorithms to assess the value of the item in the selected auction.

3.2.1.1 Cluster Analysis

The price dynamics of online auctions are different, even when those auctions are for similar items (Jank & Shmueli 2005). The value of an item assessed by the ADBA framework therefore depends on the path that its price follows during an auction. These varying price dynamics recommend partitioning of the input auction space into groups of similar auctions before we actually predict bid amounts—if we want to improve prediction outcomes. The ADBA framework, thus, divides the auction space into a set of auctions with similar price dynamics before proceeding to forecast the closing price.

Furthermore, ADBA predicts the amount of a bid in an ongoing auction at a particular moment during the auction and at the end of the auction. This contrasts with static prediction models that solely predict the closing price of the auction based on information which is available only at the start of the auction (Ghani & Simmons 2004; Van Heijst et al. 2008). The static models may use auction attributes such as the opening bid, the auction type, the length of the auction, the seller's rating and the product description. These attributes are decided at the beginning of an auction and remain fixed for its active period. At the same time, however, these models do not incorporate the dynamic attributes which are generated using information that arrives during the active period of the auction such as the bid rate of the auction, the number of auction participants and the amount of a bid at a particular time and moment of competition. ADBA therefore extracts these essential auction features from the auction history in order to assess the value of the item based on dynamic auction attributes.

3.2.1.2 Bid Mapping and Selection

The Bid Mapper and Selector (BMS) algorithm is applied to select the target auction to participate in. Once similar auctions are grouped together based on their price dynamics using cluster analysis, the next step is to map current ongoing auctions to match these clusters with similar price dynamics (see Figure 3.3). There are numerous active auctions for very popular items, and it is very difficult to choose one to participate in. The rating of auctions allows bidders to select a subset of auctions with the lowest prices, which in turn saves their energy and time (Wang et al. 2008). After the mapping of ongoing auctions to a given cluster, the auctions can be rated based on their lowest price to assist bidders in choosing a subset of auctions. The target auction can then be selected where participation will deliver bidders the maximum surplus.

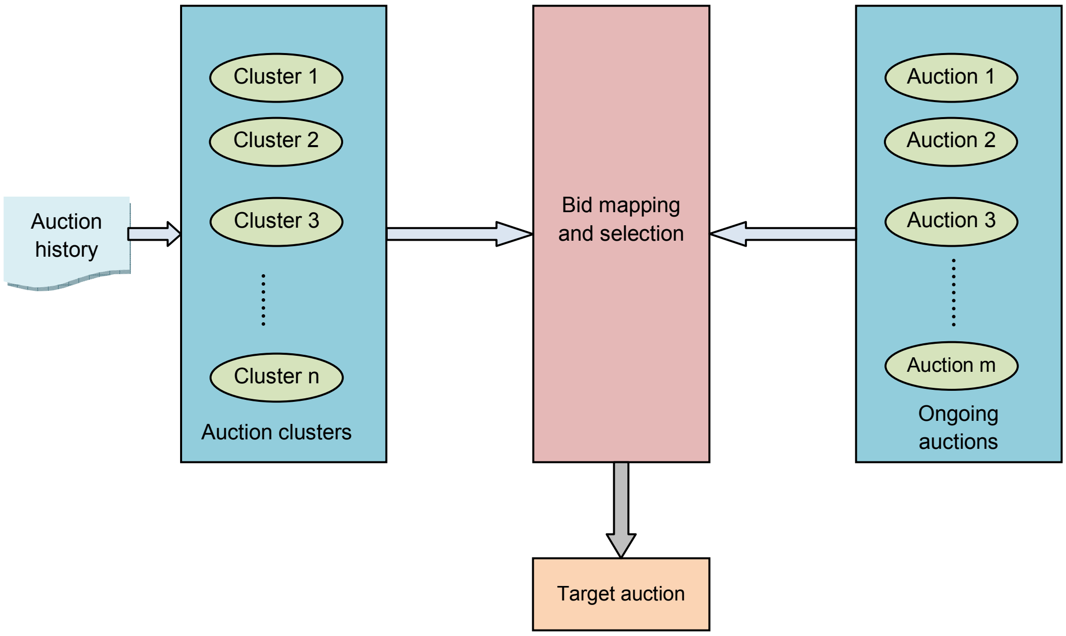


Figure 3.3 Bid mapping and selection technique in the ADBA framework

3.2.1.3 Value Assessment

Once an auction is selected for participation, the next step is to assess the value of the item in that target auction. Having the ability to accurately assess the value of an auctioned item would assist both bidders and sellers. Knowing an item's actual value would help to solve some of the information asymmetry problems that draw out transactions and inflate costs (Raykhel & Ventura 2008; Sun 2005). It would also prevent the overpricing of items by establishing a reference price for bidders which might benefit auction houses and sellers. These auction houses could use the true value of the item to achieve faster clearance of customers—the sellers who use the auction houses to re-sell their items—by paying them for their products before auctions close. This would again save customers' time. Awareness of the true value of items would improve auction houses' abilities when it comes to optimising both the selling price of items and auction features. Furthermore, it would allow them to offer insurance to sellers guaranteeing a minimum sale price for an item. A dynamic price forecasting model can value an item at a particular moment in time during the active period of its auction; depending on how the auction is going, this adds to seller satisfaction. The dynamic nature of this model could allow this framework to insure sellers dynamically in contrast with static models (Wang et al. 2008).

The value of the item is assessed by predicting the closing price of its auction. Knowing that closing price helps bidders to establish the true value of the item and decide on their maximum valuation. The task of closing price prediction is handled here using machine learning algorithms. Two approaches—one parametric and the other non-parametric—are considered when predicting the continuous price value of the item. A detailed description of all three of these steps of phase 1 of the ADBA framework follows in Chapter 4. This assessed value becomes the initial price during phase 2 of the framework.

3.2.2 Phase 2: Final Price Prediction

The second phase of the framework predicts the final price of the auction by creating bidding strategies for bidders. Based on the output of the first phase, these bidding strategies are designed to forecast the amount of a bid at a particular moment in time in the target auction for bidders with different bidding behaviours. Recognising the bidding behaviours of bidders is fundamental to the design of an efficient electronic marketplace. In addition, calculating bids in online auctions is not an easy task even when bidders know the true value (initial price from ADBA phase 1) of the item in the auction. Efficient bidding involves skilful decision making. These decisions include determining the amount and timing of a bid.

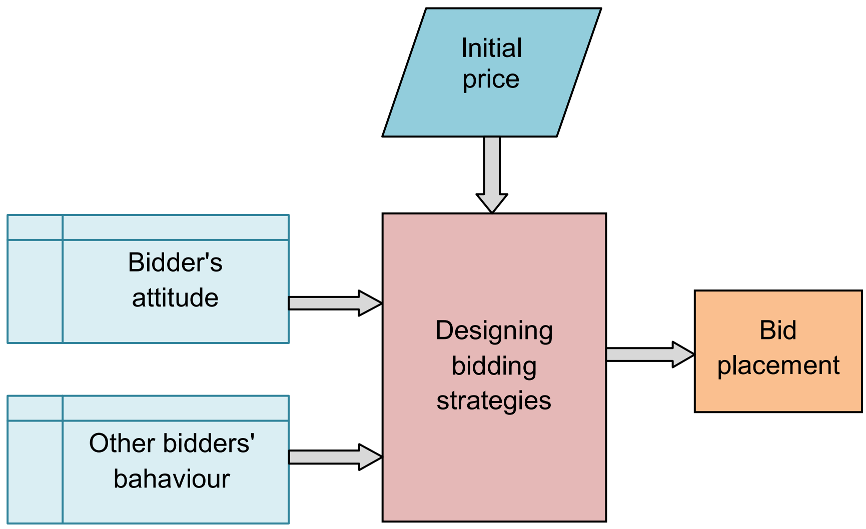


Figure 3.4 Final price prediction in the ADBA framework

When making these decisions, bidders evaluate their chance of obtaining the item in the auction taking into account bidding characteristics such as the auction's attributes, their own attitude to bidding in the auction to get the product and other bidders' behaviour (see Figure 3.4). The role of the auction's features and their use is explained in Section 3.2.1 covering the design of phase 1 of the ADBA framework for initial price estimation.

The bidders' own attitude to bidding in the auction to obtain the item has a huge impact on their bidding decisions. The bidder's attitude reflects many bidding characteristics, which may include the number of times they bid in an auction, the timing of their bids and the degree of their desperation or desire to bargain (Anthony & Jennings 2003; Borle et al. 2006). Bidders may place single or multiple bids in an auction based on their bidding behaviour. The amount of a bid depends on the time that it is placed. Bidders are also more likely to bid their true valuation of the item near to the auction's close when less time remains. Around this closing time, bidders have limited time to win the auction and they are under time pressure, so they are more likely to exceed their bidding limits. On the other hand, bidders do not exceed their bidding limits when there is abundant time before the auction ends. Other characteristics that bidders consider when competing for an item are their desperation to win the auction, or wish to bargain. Those desperate to win will bid high as soon as they start bidding so as to maximise their chances. On the other hand, if they wish to bargain, they will begin with lower bids and slowly raise these by increments approaching the maximum they are willing to pay near the closing moments of the auction.

Bidders can judge other bidders' behaviour using those bidders' previous bids. Rational bidders bid strategically using the behaviour of other bidders (the amounts they bid) as a guide to avoid the winner's curse (Ku 2000). These bidders are more likely to bid their real valuation or higher when other bidders lodge higher bids. Participants trying to be the highest bidder find themselves in a competitive space with others bidding against them. When one participant bids high, his opponents also tend to raise their bids. On the other hand, when a bidder offers a lower amount, competing bidders tend not to place high bids since they aim to purchase the item at maximum surplus. The bidding decisions of bidders are, thus, influenced by other bidders' past bids.

In this thesis, time- and behaviour-dependent negotiation decision functions and fuzzy reasoning techniques are exploited to assist bidders to place intelligent bids. The proposed methodology uses the following bidding characteristics to design bidding strategies for bidders: the auction’s attributes, the bidder's own attitude to bidding in the auction to get the product, and other bidders’ behaviour. A detailed description of the design of phase 2 of the ADBA framework is given in Chapter 5.

The ADBA framework forecasts the price at a particular moment in time based on the characteristics of auctions and bidding behaviours of bidders for Standard English auctions with hard closing rules. However, to adapt ADBA framework for other common auction formats such as Dutch, Sealed-bid and Continouse double auctions, the initial price estimation and the final price prediction phases are required to be reconsidered.

3.3 Summary

This chapter has presented an automated dynamic bidding agent framework for simultaneous online auctions of the same or similar items. It has provided an overview of the important methodologies of this research as well as the relationships among them. Although many statistical and soft computing models and methods have been applied to the design of bidding strategies for online auctions, they have been unable to incorporate the diverse price dynamics of auctions of the same or similar items in their valuations of goods in their price prediction models. They similarly have not considered the behaviour of competing participants for developing successful bidding strategies.

In contrast, the proposed ADBA framework in this thesis deals with these restrictions, producing a number of significant realisations. The ADBA framework considers different auction attributes and the price dynamics of online auctions when choosing the auction where participation will mean maximum surplus. The agent also considers the diverse price dynamics of auctions of the same or similar items during the auction selection and value assessment processes. Finally, this framework develops efficient bidding strategies for bidders by attending their own bidding behaviour as well as behavior of competing participants.

Chapter 4

Initial Price Estimation for Auction Selection and Value Assessment

This chapter designs a comprehensive methodology for initial price estimation in order to choose an auction to participate in from simultaneous auctions for the same or similar items and to assess the value of the auctioned item. To achieve these aims, a clustering approach is adopted along with bid mapping and selection technique. The value is assessed by characterising similar auctions based on not only their static attributes but their evolving attributes such as their price dynamics. A search is made for past auctions that have similar dynamics to those of current ongoing auctions in order to assess the value of the items now being auctioned. The rest of the chapter follows this schema: Section 4.1 designs the methodology for initial price estimation in order to select a target auction to participate in and assess the value of its item. Section 4.2 describes the dataset used for these validations. Section 4.3 presents the experiments used to validate the approach and the results achieved. The proposed methodology is compared with other methods in Section 4.4. Lastly, Section 4.5 offers a brief conclusion summarising and bringing together the main areas covered in this chapter.

4.1 Initial Price Estimation for Online Auctions having Hard Closing Rules

This thesis introduces a methodology for initial price estimation which chooses a target auction to compete in from simultaneous auctions for the same or similar items and assesses the value of the item on offer. The focus here is strictly on auctions— like eBay—with hard closing rules, which means that the end time of these auctions is fixed. The methodology for initial price estimation considers past auctions for the same or similar items in order to choose an auction to enter and appraise the value of the auctioned item. As we have discussed in Chapter 3, these auctions often possess different price dynamics even when they are for the same or similar items. The algorithm for auction selection and value assessment starts therefore by characterising different types of auctions based on their price dynamics.

This approach consists of three formal steps. First, sets of similar auctions are clustered together in k groups. Second, based on the transformed post-clustering data and the characteristics of current auctions, a bid mapping and selection component nominates the cluster for each ongoing auction to select the auction to participate in. Finally, the value of the item in that auction is assessed.

4.1.1 Cluster Analysis

A clustering-based method is used to assess the value of items in multiple ongoing auctions for an autonomous agent-based system. In the proposed methodology, historical auction data is extracted for the required dynamic attributes. These form the agent's knowledge base of auctions for cluster analysis. Let ATT_i be the set of attributes collected for the i^{th} auction, so that $ATT\_i = \{a_1, a_2 \ldots\ldots a_j\}$ where j is the total number of attributes. Different types of auctions are categorised based on the attributes of online auctions. These attributes may include the average bid amount, the average bid rate, the number of bids, the item type, the seller's reputation, the opening bid, the closing bid, the quantity of items available in the auction, the type of auction, the duration of the auction, the buyer's reputation and many more. To classify the different types of auctions, this thesis concentrates on a single set of attributes; the opening bid, the closing price, the number of bids, the average bid amount and the average bid rate. Of these, the opening bid and closing price are static attributes, i.e. they do not shift over time, while the number of bids, the average bid amount and the average bid rate are dynamic attributes, which change as the auction progresses.

Now $ATT\_i=\{OpenB_i, CloseP_i, NUM_i, AvgB_i, AvgBR_i\}$

where ATT_i is the set of attributes of the i^{th} auction

$OpenB_i$ is the opening bid or the starting price of the i^{th} auction

$CloseP_i$ is the closing price of the i^{th} auction

NUM_i is the total number of bids placed in the i^{th} auction

$AvgB_i$ is the average bid amount of the i^{th} auction and can be calculated as $Avg(B_1, B_2, ...B_l)$ where B_1 is the 1st bid amount, B_2 is the second bid amount and B_l is the last bid amount for the i^{th} auction.

$AvgBR_i$ is the average bid rate of the i^{th} auction and is calculated as

$$AvgBR_i = \frac{1}{n}\sum_{i=1}^{n}\frac{B_{i+1}-B_i}{t_{i+1}-t_i} \tag{4.1}$$

where B_{i+1} is the amount of $(i+1)^{th}$ bid, B_i the amount of the i^{th} bid, t_{i+1} is the time at which $(i+1)^{th}$ bid is placed and t_i is the time at which the i^{th} bid is placed.

The input auctions are partitioned into groups of similar auctions depending on their different characteristics. This partitioning has been achieved by using the *k-means* clustering algorithm. The main objective of the *k-means* clustering is to define *k* centroids, one for each cluster.

Deciding the value of *k* in the *k-means* algorithm is a recurring problem in clustering and remains a distinct issue during the process of actually solving the clustering problem. The optimal choice of *k* is often ambiguous; increasing the value of *k* always reduces the error and increases the computation speed. The most favourable method of finding *k* is to adopt a strategy which strikes a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. There are several approaches available to decide the value of *k*: rule of thumb (Fred & Jain 2002), the elbow method (Bouguessa et al. 2006), the information criterion approach (Fraley & Raftery 1998), the information theoretic approach (Still & Bialek 2004) and choosing *k* using the Silhouette (Rousseeuw 1987). In this thesis, we explore the Elbow approach using a one-way analysis of variance (ANOVA) which works well in many situations (Milligan & Cooper 1985; Mooi & Sarstedt 2011). This method estimates the value of k so that adding another cluster does not improve the data modelling. A graph has been plotted for percent of variance against the number of clusters (Figure 4.1). We choose the value of k where marginal gain drops in the graph.

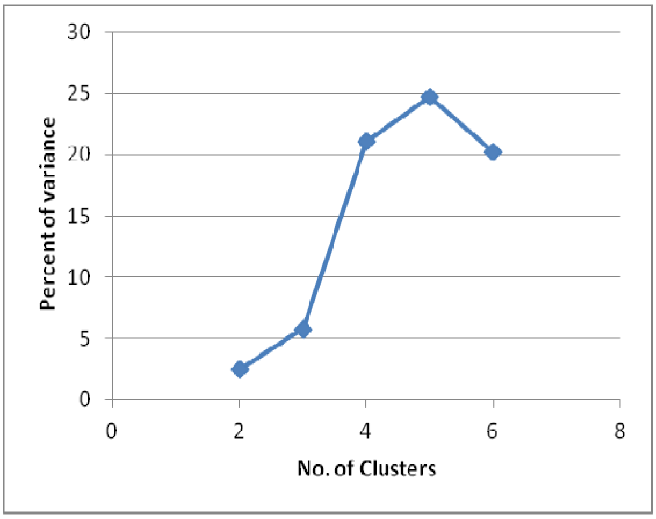


Figure 4.1 Choosing the value of k

Below is the mathematical model explained for the value of k.

$$\text{Percent of variance}=\frac{\sum_{n=1}^{k}\sum_{i=1}^{N}\left(x_{in}-\bar{x}\right)^2-\sum_{n=1}^{k}\sum_{i=1}^{N}\left(x_{in}-\bar{x}_n\right)^2}{\sum_{n=1}^{k}\sum_{i=1}^{N}\left(x_{in}-\bar{x}\right)^2} \tag{4.2}$$

where k is the total number of clusters

N is the number of auctions in the n^{th} cluster

$\bar{x}_n$ is the mean of distances of auctions from the cluster center in the n^{th} cluster

x_{in} is the distance of the i^{th} auction in the n^{th} cluster from its cluster centre

$\overline{x}$ is the mean of the distances of auctions from the whole auction data set centre

Once we decide the value of k, the *k-means* clustering algorithm is used to partition the similar auctions based on their characteristics. Given a set A of N auctions $A=\{a_1,a_2,...a_N\}$ where each auction is a *5-dimensional* real vector ATT_i and $ATT\_i=\{OpenB_i, CloseP_i, NUM_i, AvgB_i, AvgBR_i\}$, *K-means* clustering aims to partition N auctions into k clusters $(k<N)$ so that within-cluster dispersion is minimal. The within-cluster dispersion is the sum of squared Euclidean distances of auctions from their cluster centroid. The steps of the *k-means* clustering algorithm are as follows:

Initialisation step: Initialise k-centroid vectors for k initial clusters as $c_1, c_2, ...c_k$ where $c_n = \{OpenB_n, CloseP_n, NUM_n, AvgB_n, AvgBR_n\}$ and $OpenB_n$ is calculated as $(OpenB_{1n} + OpenB_{2n} + ... + OpenB_{Nn})/N$, where N is the number of auctions in the n^{th} cluster. The values of $CloseP_n$, NUM_n, $AvgB_n$, and $AvgBR_n$ are calculated similarly.

Assignment step: Assign each auction to a cluster with the closest mean by calculating the Euclidean distance of each auction from each of these k centroids as follows:

$$d_{in} = \sqrt{\left(OpenB_i - OpenB_n\right)^2 + \left(CloseP_i - CloseP_n\right)^2 + \left(NUM_i - NUM_n\right)^2 + \left(AvgB_i - AvgB_n\right)^2 + \left(AvgBR_i - AvgBR_n\right)^2} \tag{4.3}$$

where d_{in} is the Euclidean distance of the i^{th} auction from the n^{th} cluster. The auction data is normalised before computing the distance to ensure that the distance measure is not dominated by variables with a large scale.

Update step: Calculate the new means as *k*-centroid vectors for *k* auction clusters which are generated in the assignment step as $C_1, C_2, \ldots C_k$ where $C_n = \left\{OpenB_n, CloseP_n, NUM_n, AvgB_n, AvgBR_n\right\}$ and $OpenB_n$ is calculated as $(OpenB_{1n} + OpenB_{2n} + \ldots\ldots + OpenB_{Nn})/N$, where N is the number of auctions in the n^{th} cluster. The values of $CloseP_n$, NUM_n, $AvgB_n$, and $AvgBR_n$ are calculated similarly.

We repeat the assignment and update steps until moving an auction between clusters increases the within-cluster dispersion.

4.1.2 Bid Mapping and Selection

In order to decide the cluster which current ongoing auctions belong to, the bid mapping and selection component is activated. Based on the transformed post-clustering data and the current auctions' characteristics, the component nominates a cluster for each ongoing auction so that one can be chosen to participate in.

The use of a Bid Mapper and Selector (BMS) algorithm is proposed for the process of finding the target auction to compete in. The clustering technique described in the previous section characterises different auctions following a specific price path based on the distinct range of the average bid rate (*AvgBR*). This thesis, as we have noted, focuses on auctions with hard closing rules like eBay auctions. These hard closing rules persuade many bidders not to bid until the closing moments of the auction, so these bidders do not reveal the maximum they are willing to pay during the early moments of the auction. To neutralise this tendency, eBay uses a proxy bidding system which allows a bidder to submit the maximum amount that he is prepared to pay instead of his current bid. The proxy system keeps this amount private; it bids on the bidder's behalf at just one increment over the next highest bid until it reaches the maximum that the bidder is willing to hand over. This helps to reduce hikes in price paths near the end of the auction and to keep the average bid rate (*AvgBR*) at almost the same level in the closing moments as earlier on. So, the BMS algorithm uses the *AvgBR* value near the closing time when it maps the ongoing auctions to the clusters.

Given a set of ongoing auctions $OA= OA_1 \cup OA_2 \cup \ldots\ldots \cup OA_k$. where $OA_i =\{OA_{i1}, OA_{i2}, \ldots\ldots OA_{in}\}$, OA_i is the set of ongoing auctions belonging to the i^{th} cluster , $i=1,2,\ldots.k$ where k is the total number of clusters and n is the total number of ongoing auctions belonging to the i^{th} cluster, the BMS algorithm selects a subset of OA if $AvgB_{kn} < AvgBC_k$, where $AvgB_{kn}$ is the average bid amount of the n^{th} auction in the k^{th} cluster and $AvgBC_k$ is the average bid amount of the k^{th} cluster.

The target auction *TA* is selected on the basis that it gives the maximum surplus to bidders. As defined in Chapter 2, the surplus is the return that bidders enjoy by winning an auction at a price lower than its predicted closing price (Dass et al. 2011). Auctions with lower average bid amounts are expected to give higher surplus to bidders. The target auction *TA* is, thus, selected for participation on the basis that its average bid amount is lowest in order to maximise bidders' surplus. Figure 4.2 presents the developed algorithm:

```
K ← total number of clusters
N ← total number of auctions
Cnt_k ← counts the number of auctions in the k^th cluster
AvgBROA_i ← average bid rate of the i^th ongoing auction
MinAvgBRC_k ← minimum average bid rate of the auctions in the k^th cluster
MaxAvgBRC_k ← maximum average bid rate of the auctions in the k^th cluster
OA_k ←set of auctions in the k^th cluster
AvgBC_k ← average bid amount of the auctions in the k^th cluster
TA ← target auction for participation
AvgB_TA ← average bid amount of the target auction


AvgB_TA ← 0;
TA← 0;
for  i ← 1 to N {    // Check all the auctions.
       Cnt_k ←  0    //Init variable that counts the auctions in a cluster.
       for k ← 1 to K {       // Loop to check all the clusters.

                 // Check if the value is in range of the cluster.
                 if ( MinAvgBRC_k ≤ AvgBROA_i ≤ MaxAvgBRC_k ) {
                          OA_k  ← OA_k U OA_i;  // Add the auction to the current cluster.
                          Cnt_k = Cnt_k +1;        // Increment the cnt.

                          if (AvgBOA_i ≤  AvgBC_k){      //Compare with the cluster min.
                                   AvgBC_k = AvgBOA_i; //New cluster min.
                          }//end if

                          if(AvgB_TA = 0 OR AvgBC_k < AvgB_TA ) {
                               AvgB_TA = AvgBC_k;
                               TA = OA_i;          // Target auction.
                          }//end if
                 }// end if
       } // for 1 to K
} //for1 to N
```

Figure 4.2 Bid Mapper and Selector algorithm (BMS)

4.1.3 Value Assessment

Once an auction is selected for participation, the next task is to assess the value of the auctioned item. The value of the item is assessed by predicting the closing price of the auction. Knowing the closing price predicted for the auction helps bidders to establish the true value of the item and in turn finalise their own maximum valuation of the auctioned item. The closing price prediction task is handled using machine learning algorithms. Two approaches were considered in predicting the continuous price value of the item: parametric and non-parametric.

4.1.3.1 Parametric Approach to Price Prediction

Multiple linear regression is the most common parametric approach to making predictions (Han & Kamber 2006, Shmueli et al. 2007, Xuefeng et al. 2006). This model has been used to fit a linear relationship between the dependent variable (the closing price) and a set of predictor variables. The multiple linear regression model is employed based on the distinct clusters' characteristics. To predict the closing price of the selected auction, regression coefficients w_j for each attribute are opted such that the sum of squares between the predicted and the actual bid (4.4) over all the training auction data is minimal.

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{m} w_j a_{ij} \right)^2 \tag{4.4}$$

where y_i is the actual bid for the i^{th} auction

m is the total number of attributes

w_j is the regression coefficient for the j^{th} attribute

a_{ij} is the j^{th} attribute for the i^{th} auction.

4.1.3.2 Non-parametric Approach to Price Prediction

A k-Nearest Neighbour approach is used to identify k auctions in the dataset that are similar to the target auction whose price is to be predicted (Han & Kamber 2006, Shmueli et al. 2007, Xuefeng et al. 2006). This is a simple classification method that does not make assumptions about the form of the relationship between the closing price of the auction and the predictor variables. This is a non-parametric method because it does not involve estimation of parameters (coefficients) as a linear regression technique would; instead, this method draws information from similarities between the predictor values of the different auctions in the dataset based on the distance between these auction records.

The *k-NN* is employed based on the distinct clusters' characteristics. Euclidean distance (d_{ij}) is used to measure the distance between the target auction and the auctions in the respective cluster (the one to which it belongs). The auction data is normalised before computing the distance to ensure that the distance measure is not dominated by variables with a large scale.

$$d_{ij} = \sqrt{\left(OpenB_i - OpenB_j\right)^2 + \left(CloseP_i - CloseP_j\right)^2 + \left(NUM_i - NUM_j\right)^2 + \left(AvgB_i - AvgB_j\right)^2 + \left(AvgBR_i - AvgBR_j\right)^2} \tag{4.5}$$

where d_{ij} is the Euclidean distance between the i^{th} and j^{th} auction.

After computing the distance between the auctions, the target auction is assigned to a class of k auctions based on the classes of its auction neighbours. A value of k is chosen which achieves a balance between over-fitting to the predictor information (if k is too low) and ignoring this information completely (if k is too high). To find the optimal choice of k, the auctions in the training dataset are used to classify the auctions in the validation dataset and the error rates for each value of k are calculated. A value of k is chosen which has the best classification performance and so minimises the misclassification rate (of the validation set).

The closing price of the target auction is predicted using the weighted average of the closing price of the k nearest neighbour.

$$ClosePT_{TA} = \frac{\sum_{i=1}^{k} w_i ClosePT_i}{\sum_{i=1}^{k} w_i} \tag{4.6}$$

where k auctions are ordered as per their increasing distances from the selected auction, w_i is the weight assigned to the i^{th} auction and $w_1 > w_2 > w_3 > ......w_k$, $CloseP_{TA}$ is the closing price of the target auction.

4.1.4 Validation of the Auction Clusters

The resulting clusters are validated using two criteria: cluster separation and cluster interpretability. *Dunn's Index (DI)* is applied as a measure to identify "compact and well separated" (*CWS*) clusters (Halkidi et al. 2001; Rivera-Borroto et al. 2012). *Dunn's index (DI)* is defined for k clusters as below:

$$DI = \min_{1 \le m \le k} \left\{ \min_{1 \le n \le k, m \ne n} \left\{ \frac{d(m,n)}{\max_{1 \le c \le k} d'(c)} \right\} \right\} \tag{4.7}$$

where $d(m,n)$ represents the distance between clusters m and n and is defined as

$$d(m,n) = \min_{x \in m,\, y \in n} d(x,y) \tag{4.8}$$

$d'(c)$ measures the intra-cluster distance of cluster c and is defined as

$$d'(c) = \max_{x,\, y \in c} d(x,y) \tag{4.9}$$

For compact and well-separated clusters in the dataset, the distance between the clusters is expected to be greater and the intra-cluster distance is expected to be small. $DI > 1$ shows that the dataset is being partitioned into compact and well-separated clusters (Rivera-Borroto et al. 2012).

Further, for the interpretation of each cluster: a) its characteristics are explored by obtaining summary statistics from each cluster on each attribute used in the cluster analysis and b) it is categorised based on their characteristics.

4.1.5 Validation of the Value Assessment

The value of the item is assessed by predicting the closing price of its auction using parametric and non-parametric approaches based on machine learning algorithms. Multiple linear regression is used as the parametric approach, and *k*-nearest neighbour is used as the non-parametric approach. These methods are validated separately for accurate results as follows:

The multiple linear regression method for price prediction is validated by dividing the data into two distinct sets: a training dataset and a validation dataset. These sets represent the relationship between the dependent and independent variables. The training dataset is used to estimate the regression coefficients, and the validation dataset is used to validate these estimations. The prediction is made for each case in the validation data using the estimated regression coefficients and then these predictions are compared to the value of the actual dependent variable to validate the outcomes. The square root of the mean of the squares of these errors is used to compare different models and to assess the prediction accuracy of the method used.

To validate the *k*-nearest neighbour method, first, the auctions' dataset is divided into training and validation datasets and then *k*-closest members (based on the minimum distance between them) of the training dataset are located for each auction in the validation dataset. *k* models are built and scoring on the best of these models is performed to choose the value of *k*. The value of *k* is chosen which has the best classification performance while classifying the auctions in the validation data using the auctions in training data. A very low value of *k* classifies data sensitive to the local characteristics of the training data, and a large value of *k* predicts the most frequent recorded type in the dataset. To find the optimal *k*, the mis-classification rate of the validation data is examined for different values of *k* and the one is chosen which minimises the classification rate. The *k*-nearest neighbour method is validated by comparing the square root of the average of the squared residuals of different models to assess the prediction accuracy.

4.2 Dataset for Experiments

The experiments for the initial price estimation were performed on an eBay auctions dataset. This section first discusses the eBay bidding mechanism including its important features and then presents the dataset used for the experiments.

4.2.1 eBay Auctions - Model and Mechanism

eBay offers various features to its participants. This section presents the eBay model and its bidding mechanism, as well as its important features. Auctions of single items in an English auction format are, thus, considered. To start an auction on eBay, a seller needs to register for an auction of that item. The static attributes of the auction are set before the start of the auction (and do not change as the auction progresses). These may include the starting price of the item, the duration of the auction (3, 5, 7 or 10 days), the description of the item and its location, the seller's rating, the reserve price of the item set by the seller, the shipping price, terms and payment methods. The reserve price is the lowest price at which the seller is obliged to sell the item. eBay does not disclose the reserve price to participants; rather a message is displayed on the item indicating whether the reserve price has been met or not.

Potential bidders find auctions of the item they are interested in by browsing the categories listed on eBay. An example of an eBay auction item listing is presented in Figure 4.3. All bidding participants do not become aware of the auction at the same time because there is no advance announcement of an online auction unlike the situation with traditional auctions. All the auctions proceed as per the rules pre-announced by eBay. These auctions use an open, ascending-bid format that is different from traditional auctions in term of its ending rules. eBay auctions have a fixed closing time instead of the 'going-going-gone' ending rule of traditional auctions. This hard closing rule of eBay auctions encourages many bidders to withhold their bids until the final moments of the auction, so that they do not reveal the highest amount they are willing to pay for the item. To stabilise this behaviour, eBay uses a proxy bidding system.

The proxy bidding system of eBay allows the bidder to submit the maximum value he is prepared to pay instead of his current bid. The proxy system keeps this amount private and bids on the bidder's behalf at just one increment over the next highest bid until it reaches the maximum the bidder is willing to pay. If the bid amount by any other bidder is greater than the bidder's ceiling, then the proxy bidding system does not bid for that overtaken bidder. Instead it sends him an email about the need to revise his maximum bid.

Figure 4.3 An example of an eBay auction item listing

For instance, when a bidder submits a proxy bid in an auction, he is asked to enter the maximum amount which he is willing to pay for the auctioned item. Assume that the starting price of the auction is $15, the minimum bid increment is $0.50 and X, the first-arrived bidder submits $30 as a proxy bid. eBay automatically sets the highest bid to $15.50, just enough to make bidder X the highest bidder. Suppose another bidder, Y enters the auction and submits $25 as a proxy bid. The proxy server then raises X's bid to \$25.50 to make bidder X the higher bidder. If any other bidder submits a proxy bid higher than \$30 in the course of the auction, X will no longer be the highest bidder and the eBay server will notify him via email about this development. At that stage, bidder X can revise his proxy bid. A bidder cannot change his bid; he may increase his proxy bid, but he must not decrease the amount he has said he is willing to pay. The whole process is repeated until the auction is concluded. The winner of the auction is notified by email and he pays the second highest bid plus the bid increment of the auction. eBay maintains the auction listings and their results publicly for at least one month after the auction closes.

4.2.2 Dataset used

The dataset includes the complete bidding records for 149 auctions of new Palm M515 PDAs. This dataset was made available at http://www.ModelingOnlineAuctions.com by researchers in 2010 (Jank & Shmueli 2010). All the auctions in this dataset are for the same product: a new Palm M515 handheld device. The market price of the item at that time was \$250. The bidding record of each auction includes the auction ID, opening bid amount, the closing price of the auction, information about the ratings of bidders and the seller, the number of bids placed in the auction, all bids along with their placement and the duration of the auction. Table 4.1 shows statistical information for the dataset. The opening bid is set by the seller, influencing the number of bidders attracted to the auction. As the number of bidders in the auction rises, it becomes more competitive, and the closing price of the item is also higher. The average number of bids is 21.36 and the average bid amount of the auction is \$148.91.

Table 4.1 Description of data

Variable	*Min*	*Max*	*Mean*	*Std. Deviation*
Opening Bid(OpenB)	0.01	240	40.55	69.71
Closing Price(CloseP)	177	280.65	228.24	16.10
# Bids(NUM)	2	51	21.36	10.16
Avg bid amt(AvgB)	77.47	243.75	148.91	33.13
Avg bid rate(AvgBR)	-2196.27	42679.94	11624.09	8143.42

OpenB: Starting price of an auction.
CloseP: End price of an auction.
NUM: Total number of bids placed in an auction.
AvgB: Average bid amount of each auction.
AvgBR: Average bid rate of the auction

Bidders face information overload as they participate in eBay auctions. In particular, each bidder has information about the path of his bid in an ongoing auction, that is, the amount of that bid at a particular moment in time in the auction. Furthermore, there are a number of simultaneous auctions of the same or similar items, and the bid path will be different for each of these auctions. This is known as the price dynamics of the auction. Bidders find it increasingly difficult to process information about auctions with such diverse price dynamics. The model for estimating an auction's closing price in this thesis considers the price dynamics to be one of the important factors. The section below gives step-by-step details of the method for selecting an auction to participate in from simultaneous auctions for the same or similar items and then predicting the closing price. This closing price is called the initial price in this thesis. This is because during the second phase of bidding, it serves as the initial price when it comes to designing bidding strategies for buyers with different bidding behaviours (as described in Chapter 5).

4.3 Experiments

Several experiments were performed using the initial price estimation method in order to demonstrate the validity of the proposed approach, to evaluate the methodology with the eBay auctions dataset and to compare the results achieved to other methods. The initial price estimation method was validated with the dataset explained in Section 4.2. It is important to note that the main purpose of these experiments was to select an auction where participation would give maximum surplus and to assess the true value of the auctioned item. An auction was selected using a clustering-based methodology with a bid mapping and selection approach, and the value of the item was assessed using machine learning techniques. The experiments for auction selection and value assessment are presented below separately.

4.3.1 Experiments for Auction Selection

Auctions of the same or similar items were clustered using the *k-means* algorithm. The value of *k* in *k*-means algorithm was estimated by the Elbow approach using one-way analysis of variance (ANOVA). The percent of variance was calculated after performing clustering for subsequent values of *k* using the *k*-means algorithm to estimate the point where marginal gain drops. In the experiments, this point occurred after five clusters, as shown in and Figure 4.4. The input space was, thus, divided into five clusters, considering a set of attributes of auctions comprising the opening bid, closing price, number of bids, average bid amount and average bid rate for a particular auction. These five clusters respectively contained 19%, 34%, 20%, 21% and 6% of the auctions' data. Based on the transformed data after clustering and the characteristics of the current auctions, the BMS algorithm nominated the cluster for each of the ongoing auction to select the auction in which to participate. The algorithm used *AvgBR* value at the beginning of the last hour to map ongoing auctions to the clusters for auction selection.

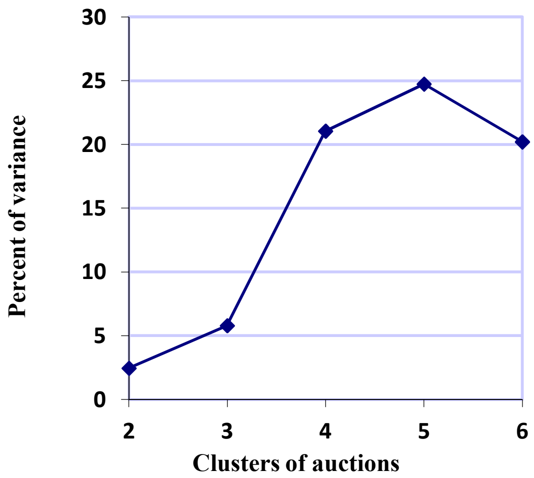


Figure 4.4 Choosing the value of *k* for auction input data

4.3.1.1 Method Validation

The clusters obtained using cluster analysis were validated using two criteria: cluster separation and cluster interpretability, as explained in Section 4.1.4. *Dunn's Index (DI)* was calculated to find *CWS* clusters. In these experiments, *DI > 1* showed that the dataset was being partitioned into the *CWS* clusters. The summary statistics from each cluster for each attribute are given in Table 4.2. It is obvious that the *k-means* clustering algorithm distinguished these five clusters according to the *AvgBR* of the auctions. We interpret these clusters with *very low* (8.30), *low* (21.69), *medium* (35.17), *high* (53.20) and *very high* (86.48) average bid rate (AvgBR) auctions. These clusters were ordered according to the increasing values of their *AvgBR* from *very low* to *very high* for the sake of convenient representation of results. It is also evident from Table 4.2 that the auctions with the highest opening bid attracted the least bidders (cluster1). In addition, the bid rate of these auctions was lowest, which lowered the closing price of the auction. On the other hand, the lowest value opening bid attracted the most bidders, which in turn raised the closing price of these auctions even with a medium bid rate (cluster 3).

It is noteworthy that in 92 of the 149 auctions in this dataset, the winner first appeared in the last hour of the auction; this accounted for 62% of total auctions, a finding consistent with the recognition of the late-bidding attitude of bidders in the online auction literature (Du et al. 2010; Ockenfels & Roth 2006; Rasmusen 2006). As explained, clustering divided the auction data into groups of auctions with a distinct range of average bid rate (*AvgBR*) values. It was observed that the value of *AvgBR* in 78% of the completed auctions belonged to the same cluster as at the beginning of the last hour of the auction. These criteria serve as benchmarks in validating the procedure adopted by the *BMS* algorithm which maps ongoing auctions to the clusters based on their *AvgBR* value at the beginning of the last hour.

Table 4.2 Attributes' statistics for each cluster

Cluster no.	*%age of Auctions' data*	*Normalized AvgBR*	*Avg. NUM*	*Avg. OpenB*	*Avg. CloseP*
1	19%	8.30	11.07	112.14	225.02
2	34%	21.69	22.53	28.31	226.65
3	20%	35.17	27.86	16.09	231.73
4	21%	53.20	21.96	24.91	225.83
5	6%	86.48	23.12	20.37	227.80

4.3.2 Experiments for Value Assessment

The predictive accuracy of the models for value assessment of the auctioned item was compared using *RMSE* (root mean square error) and *MRE* (mean relative error) error measures. For this purpose, *RMSE* was the square root of the average squared errors and *MRE* was defined as how much the prediction departed from the real price on average. The experiment was repeated four times to derive stable evaluation results, and in each of these subsequent experiments, the dataset was randomly split into training and validation sets. The average over these sets is reported as the prediction results. The *RMSE* for each cluster and average *MRE* for all the clusters are reported for evaluation in Table 4.3 and Table 4.4 respectively. The results show that the parametric approach to price prediction is not as promising as the non-parametric approach since the non-parametric approach performs better when the predicted variable can be defined by multiple combinations of predictor values allowing for a wide range of relationships between the predictors and the response. Further, in contrast to parametric linear regression functions, the non-parametric approach is able to determine which of the attributes should be used for modelling the relationship with the target variable. It was observed that the *k*-nearest neighbour technique performed better, yielding minimum *RMSE* and *MRE* on average, so this technique was used for value assessment.

Table 4.3 Root mean square error for price prediction approaches

	Parametric approach	***Non-parametric approach***
Cluster1	15.15	12.76
Cluster2	14.42	13.82
Cluster3	15.65	6.58
Cluster4	14.98	13.84
Cluster5	54.29	4.69

Table 4.4 Average mean relative error for price prediction approaches

	Parametric approach	***Non-parametric approach***
Average MRE	0.09	0.04

4.3.2.1 Method Validation

The price predicted for valuation purposes was validated by comparing the *RMSE* in two scenarios: first, the price was predicted when considering the input auctions as a whole, and second, it was predicted using different auction clusters. The results were evaluated by comparing the root mean square errors (*RMSEs*) in both of these scenarios. The results of the experiment demonstrated an improvement in RMSE by 36.64% on average when the price is predicted using different auction clusters.

4.4 Comparison with Other Algorithms

The previous section described experiments performed on the Palm PDA dataset of eBay auctions for auction selection and value assessment. The value of an item was assessed by predicting the closing price of its auction. Other studies have also used non-parametric techniques to predict the closing price of eBay auctions and their findings can be compared with the clustering-based non-parametric approach to closing price prediction in this thesis.

Van Heijst et al. (2008) validated the price prediction technique using heterogeneous and homogeneous datasets. Their study used datasets from closed Nike and Canon auctions as heterogeneous datasets. The Nike dataset included data for auctions of both used and new models of Nike men's shoes, while the Canon dataset contained data about various camera models as well as accessories like lenses and batteries. Datasets for auctions of H700 Motorola Bluetooth headsets and 30 GB Apple iPod mp3 players were included as homogeneous datasets. In these datasets, the items were technically identical, a trait also true for the dataset for auctions of new Palm M515 PDAs described above. We can, thus, turn to Van Heijst et al's closing price prediction results (using the dataset of 30 GB Apple iPod mp3 players) as a comparison for this study's clustering-based non-parametric approach to closing price prediction. The proposed price prediction methodology is this thesis is also compared with the price prediction results of Raykhel and Ventura (2009), who used laptop auctions as a homogeneous dataset.

The results achieved by these other researchers are compared with this thesis's clustering-based price prediction approaches in Table 4.5. It can be seen that the clustering-based approach is the most effective in predicting the closing price of auctions. The *MRE* recorded using the clustering-based approach (0.04) is lower than the results achieved with the algorithms of Van Heijst et al. (0.1) and Raykhel and Ventura (0.164), suggesting the greater precision of the proposed methodology.

Table 4.5 Error comparison using others' algorithms

	Clustering-based price prediction	***(Raykhel & Ventura 2009)***	***(Van Heijst et al. 2008)***
MRE	0.04	0.164	0.1

4.5 Summary

This chapter has proposed a method for initial price estimation which selects an auction to compete in and assesses the value of the auctioned item. It is unrealistic to assume that the price dynamics are identical in simultaneous auctions of the same or similar item. This thesis has therefore introduced a clustering-based approach with a bid mapping and selection technique to characterise different types of auctions based on their diverse price dynamics. This is as a key step before price prediction.

This chapter has also presented a BMS algorithm for selecting the auction where participation will give maximum surplus. The value of the auctioned item is assessed using parametric and non-parametric machine learning techniques. The proposed approach has been validated using eBay auctions for a new Palm M515 PDAs dataset. The results demonstrate that this clustering-based approach outperforms other methodologies in the literature in terms of prediction accuracy.

Chapter 5

Designing Bidding Strategies for Different Bidding Behaviours

This chapter discusses the bidding strategy design phase of the automated dynamic bidding agent (ADBA) framework given in Chapter 3. These bidding strategies are designed for potential buyers and aim to forecast their bid amounts at a particular moment in time based on their bidding behaviour and their valuation of an auctioned item, as assessed in Chapter 4. The remainder of the chapter proceeds as follows: Section 5.1 profiles the types of bidders that participate in online auctions. Section 5.2 describes the bidding strategies used by ADBA agents. Section 5.3 and Section 5.4 introduce the methodologies used for designing bidding strategies for bidders using negotiation decision functions and fuzzy reasoning techniques respectively. Section 5.5 describes the evaluation technique and performance measures used to validate the methodologies presented. Section 5.6 draws conclusions and summarises the key contributions of this chapter.

5.1 Types of Bidders

Bidders find it difficult to make bidding decisions even after selecting an auction to compete in and finalising their own valuation of the item on offer. This is because these decisions depend on their bidding behaviour. Common bidding patterns in online auctions with hard closing rules include placing one or several bids towards the auction's deadline; making just a single bid in the auction's last seconds; and placing a number of incremental bids during the auction. For bidders in eBay-style auctions, however, late bidding—making only a single bid in the auction's its dying moments— remains the most accepted behaviour. Late bidders engage in intelligent bidding, drawing on the information they have gathered from the earlier bids of other participants (Ockenfels & Roth 2002; Schindler 2003). This bidding strategy avoids price wars, affording buyers higher payoffs (Bajari & Hortacsu 2003a). Late bidding is also an optimal strategy for well-informed bidders who avoid divulging information through their own early bids (Wintr 2008). This information can be used by experienced bidders to discern the true resale value of the auctioned item. Further, late bidding is the ideal response to dishonest sellers who attempt to drive up the price, engaging shill buyers to bid against a proxy bidder.

Nevertheless, there are risks involved with last-second bidding. These late-coming bids may be lost in Internet congestion and extended connection times. As a result, bids may not reach the auction before its closes. One survey found that 86% of participants had experienced this problem at least once (Ockenfels & Roth 2002). In addition, a strict focus on calculated last-moment bidding does not allow for the emotional overbidding that most buyers experience when bidding on eBay. Moreover, if we consider the auction-bidding process as a kind of sport, last-moment bidding shows poor sportsmanship that misleads the auction. Therefore, in this thesis, bidding strategies are designed not only for the types of bidders who interject a single bid at the last second in the auction, but also for those who place one or many bids towards the tail end. Bidders are categorised based on the number of bids they make and the timing of their bid placement. Those who make just a single bid are identified as *Mystical* and *Sturdy* bidders. *Mystical* place their only bid in the closing moments (the last five seconds) of the auction while *Sturdy* bidders lodge just one bid in the five minutes before the end. Bidders who make several bids in the last hour of the auction are identified as *Strategic*. These bidders up their bid amounts strategically based on the bids recorded by other participants.

Generally speaking, bidders tend to exhibit one of two attitudes: they may be desperate to win an item, or they may be willing to bargain for it (Anthony & Jennings 2003). These two attitudes are described here as *Ambitious* and *Sophisticated* respectively.

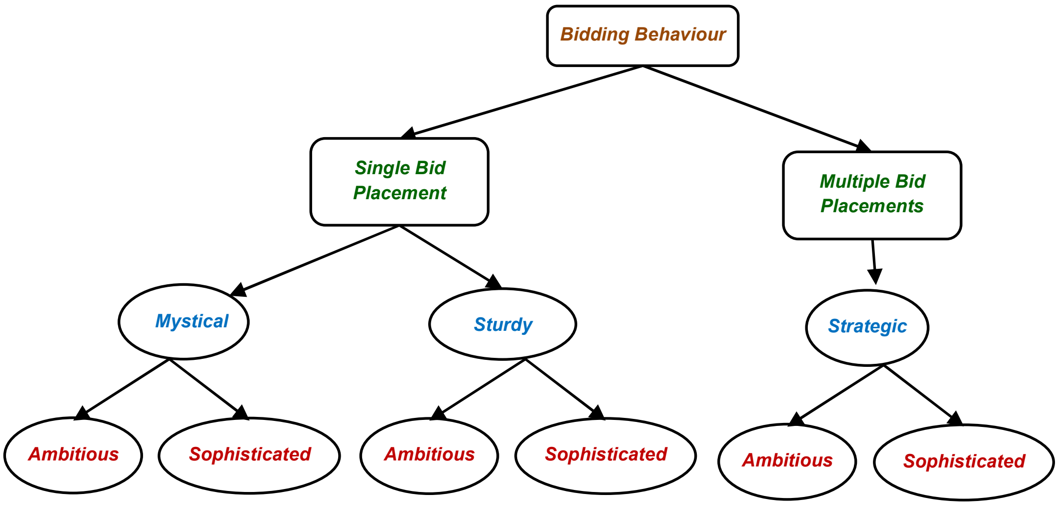


Figure 5.1 Types of bidders

5.2 Bidding Strategies for the ADBA Agent

This thesis introduces a methodology for designing bidding strategies for ADBA agents. This is based on the calculation of a bid amount at a particular moment in time for *Mystical*, *Sturdy* and *Strategic* bidders. A bid equals the maximum value that the agent is willing to pay at that point in time. The agent determines this value based on bidding characteristics such as the auction's attributes, the bidder's own attitude and other bidders' attitudes. The auction's attributes have been considered in predicting the initial price of the auction in Chapter 4. Two types of the bidder's own attitudes are also taken into account in this study. They have been called *Ambitious* and *Sophisticated*.

Other bidders’ attitudes are used to gauge competition in an auction; their previous bids (competing bids) are noted and exploited. Bidders update their bids at a particular moment in time based on others' bids (Ariely & Simonson 2003). When the earlier offers of other bidders (competing bids) are higher and the rate of bid change accelerates, a bidder will make higher bids in more frequent increments to win the auction. These are indications of heightened competition (Figure 5.2). Further, the time left until the auction closes is an important factor that also affects competition. Bidders decide fast towards the end of an auction due to the time pressure. This pressure increases arousal, and bidders bid beyond their limits as the deadline approaches since there is so little time left (Ku et al. 2005). This exacerbates the sense of competition among auction participants. In other words, competition rises as the remaining duration of the auction wanes (Figure 5.3).

The bidding strategies in this thesis are designed to model the bid amount at a particular moment in time for each bidding behaviour in Figure 5.1 using negotiation decision functions (Section 5.3) and fuzzy reasoning (Section 5.4).

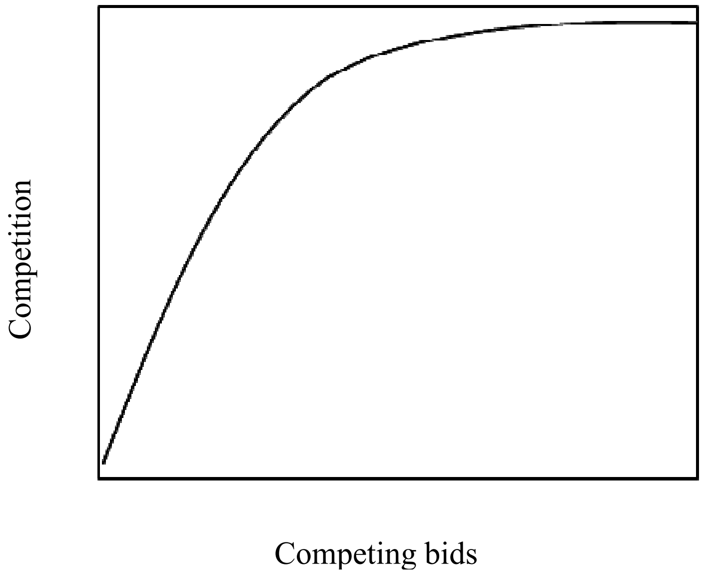


Figure 5.2 Competition versus competing bids

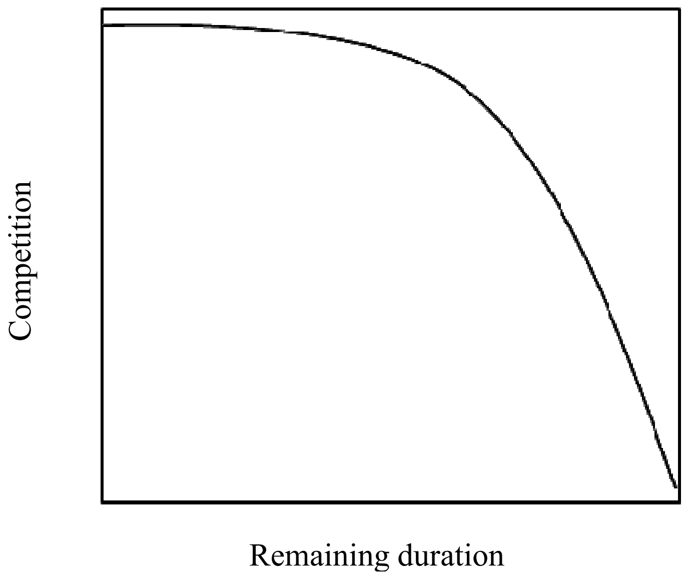


Figure 5.3 Competition versus remaining duration

5.3 Bidding Strategies Using Negotiation Decision Functions

The bidding strategies are designed by calculating the bid amount at a particular moment in time based on the initial price (p_i) (i.e. the predicted closing price of the selected auction, as described in Chapter 4) and using the negotiation decision functions (NDFs) given by Faratin et al. (1998). First of all, a bid value is recommended at a particular moment in time based on the competition in the auction. Second, this value is updated to determine the bid amount for buyers with different bidding behaviours.

5.3.1 Competition and Bid Determination

The bid amount is calculated using negotiation decision functions (NDFs) in two phases. In the first phase, the amount is determined based on the competition in an auction. NDFs are applied for the remaining duration of the auction and competing bids. In the second phase, the bid is updated in order to design bidding strategies for each type of bidding behaviour in Figure 5.1 according to the listed bidding attitudes.

Assume that $F(t)$ is the function that determines the bid amount based on the remaining duration and $F_c(t)$ is the function that determines the bid amount based on competing bids. Assume also that the agent's bids occur at time $0 \leq t \leq t_{max}$ and the agent's bidding limit is $[min_b, max_b]$. The bidding agent defines a constant k, which when multiplied by the size of the interval, gives the value of the starting bid amount. $F(t)$ with $0 \leq t \leq t_{max}$ is represented using a function $\alpha(t)$ as follows:

$$F(t) = min_b + \alpha(t)(max_b - min_b) \tag{5.1}$$

where $$\alpha(t) = k + (1-k)\left(min(t, t_{max})/t_{max}\right)^{1/\beta}$$

A wide range of time-dependent functions can be calculated by varying the value of $\alpha(t)$, where $0 \leq \alpha(0) \leq 1$, $\alpha(0)=k$ and $\alpha(t_{maxt})=1$,which ensures that the bid amount remains within the value range. Initially this represents the starting bid and at the end of the auction at $t = t_{max}$, the bid amount reaches the reservation price of bidder, i.e. p_r (the maximum value of the auctioned item set by the bidder).

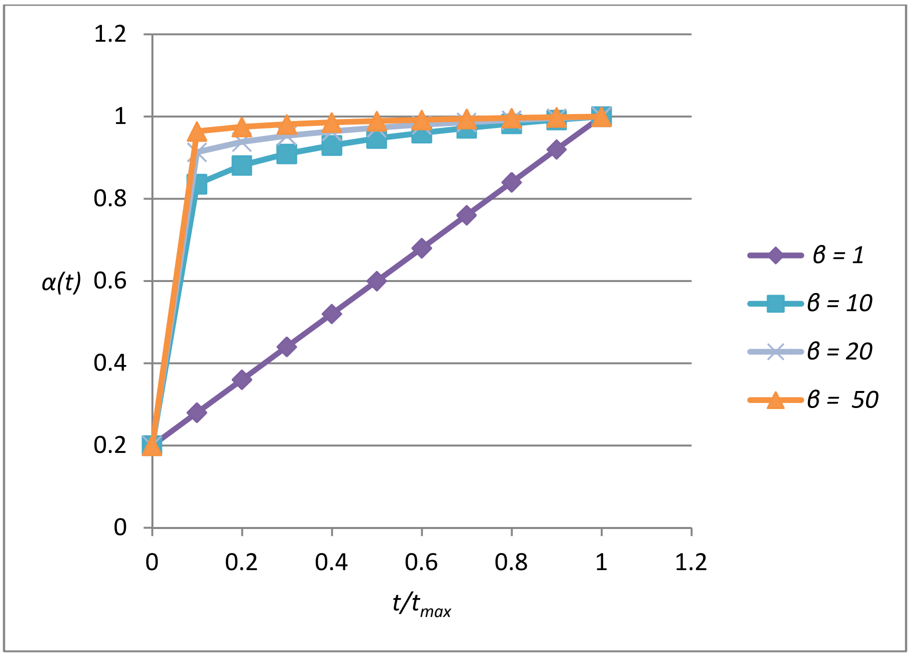
β is a constant which belongs to R^+. A number of possible bidding regulations can be obtained by varying the value of β for different bidder-specific issues. For *Ambitious* bidders who are desperate to have the item, $\beta > 1$ and the agent quickly goes to its reservation price p_r. where $p_r = p_i = max_b$. The mathematical model for this behaviour is as follows:

$$\lim_{\beta \to +\infty} k + (1-k)\left(min(t, t_{max})/t_{max}\right)^{1/\beta} = 1 \qquad (5.2)$$

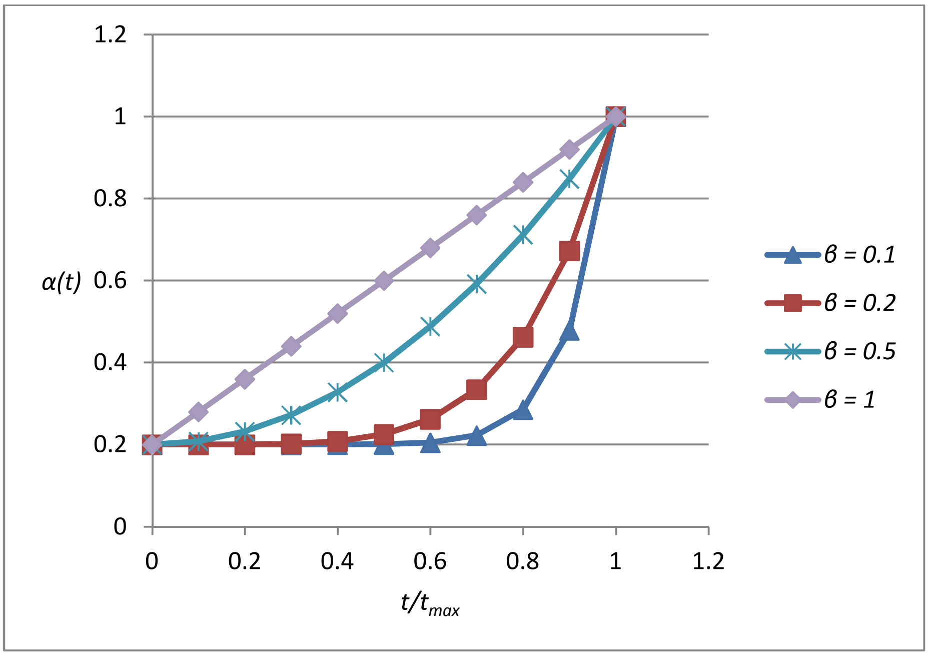
For *Sophisticated* bidders, who are willing to bargain for an item, $\beta < 1$ and the minimum bid amount is maintained until t_{max} is almost reached. The mathematical model for this behaviour is as follows:

$$\lim_{\beta \to 0^+} k + (1-k)\left(min(t, t_{max})/t_{max}\right)^{1/\beta} = k \qquad (5.3)$$

The computation of $\alpha(t)$ with respect to time (presented here as relative to t_{max}) for $\beta \geq 1$ and $\beta \leq 1$ is presented graphically in Figure 5.4.



(a)



(b)

Figure 5.4 Computation of $\alpha(t)$ for $\beta \geq 1$ (a) $\beta \leq 1$ (b)

$F_c(t)$ computes the bid amount at time t based on the previous bids placed by other participants. To calculate $F_c(t)$ at a particular moment of time t, the agent reproduces the behaviour of the other participants in earlier steps for $\delta \geq 1$ where $n > 2\delta$.

$$F_c\left(t_{n+1}\right) = min\left(max\left(\frac{F'(t_{n-2\delta+2})}{F'(t_{n-2\delta})} F(t_{n-1}), min_b \right), max_b \right) \tag{5.4}$$

and where $F'(t)$ is the bid amount placed by the other participants at time t.

At a given time, the bidding agent may consider any combination of these issues based on its current situation, As such, bid amounts can be computed to reflect the level of competition in the auction.

In the second phase, the bid amount is updated to design bidding strategies for bidders with different bidding behaviours, as detailed below.

5.3.1.1 Mystical Bidding Strategy

An agent with this strategy places a single bid in the closing moments of the auction. This bid amount depends upon the remaining duration of the auction, as well as on competing bids.

The bid amount at time t for *Mystical* behaviour will be calculated as the average of $F(t)$ as in (5.1) and $F_c(t)$ in (5.4). Here, min_b is the lower bound of the bid value at the start of the last five seconds of the auction. The values for k and β are set according to the bidding attitude of the bidders. For *Ambitious Mystical* bidders, the value of k will be high and $\beta > 1$, since this type of bidder bids at a price near to the reservation price p_r. On the other hand, for *Sophisticated Mystical* bidders, the value of k will be low and $\beta < 1$.

5.3.1.2 Sturdy Bidding Strategy

This strategy is similar to *Mystical* bidding behaviour with the exception of the time at which a bid is placed. A bidder with *Sturdy* behaviour will place a single bid during the last five minutes of the auction based on the time remaining and the competing bids.

$F(t)$ and $F_c(t)$ functions are similar to those in *Mystical* behaviour where they are used to compute the bid amount. Here, however, min_b is the lower bound of the bid amount at the beginning of the last five minutes of the auction. The values for k and β for *Ambitious Sturdy* and *Sophisticated Sturdy* bidders follow the same conventions as when *Mystical* bidders have these attitudes.

5.3.1.3 Strategic Bidding Strategy

Strategic bidders place multiple bids during the last hour of an auction. In this strategy, each and every bid is strategically placed based on the bids of competing bidders in the auction. These bidders continue bidding until the bid amount reaches their reservation price p_r.

Each bid will be calculated as the average of $F(t)$ and $F_c(t)$. The value of min_b is the lower bound of the bid amount at the beginning of the last hour of the auction. An *Ambitious Strategic* bidder starts bidding at a value close to his valuation for the item; the values for k are high for this bidding strategy. Here $\beta > 1$, since bidders with an *Ambitious* attitude tend to quickly reach p_r before the deadline is reached by placing multiple bids. However, *Sophisticated Strategic* bidders do not start bidding at an amount close to p_r; rather, they increase their bid amount slowly based on the other bids in the auction. As such, the values for k are low and $\beta < 1$ for *Sophisticated Strategic* behaviour.

Assume *AMST* and *SMST* represent the *Ambitious Mystical* and *Sophisticated Mystical* bidding strategies respectively, and that *ASTD* and *SSTD* represent the *Ambitious Sturdy* and *Sophisticated Sturdy* bidding strategies respectively. *ASTG* and S*STG* represent the *Ambitious Strategic* and *Sophisticated Strategic* bidding strategies respectively. The values of k and β for these behaviours are shown in Table 5.1.

Table 5.1 Choice of *k* and *β* for different bidding strategies

Bidding Strategy	***k***	***β***
AMST	$0.6\leq k\leq1$	$\beta > 1$
SMST	$0\leq k\leq0.3$	$\beta < 1$
ASTD	$0.6\leq k\leq1$	$\beta > 1$
SSTD	$0\leq k\leq0.3$	$\beta < 1$
ASTG	$0.6\leq k\leq1$	$\beta > 1$
SSTG	$0\leq k\leq0.6$	$\beta < 1$

5.4 Bidding Strategies Using Fuzzy Reasoning

In this section, Mamdani's Method for fuzzy relations and the compositional rule of inference are used to design bidding strategies for buyers (Tanaka & Niimura 1997). The bid increment for the auction is calculated based on the competition in that auction, and the bidding attitude of buyers with different bidding behaviour (where bid increment (ΔP) is the amount by which the bidder raises the current bid (initial bid p_i)). First, the level of competition in the auction is assessed, and then the bid increment is calculated for different bidding behaviours of bidders.

5.4.1 Competition Assessment

The degree of competition is assessed using the remaining duration of the auction and the previous offers made by competing participants. Assume C is competition, having a fuzzy set of values as $c_1, c_2, \ldots\ldots c_n$, D is the remaining duration, having a fuzzy set of values as $d_1, d_2, \ldots\ldots d_n$ and B is competing bids, having a fuzzy set of values $b_1, b_2, \ldots\ldots b_n$. According to Mamdani's Direct Method, the adaptability n no. of rules $w_1, w_2 \ldots\ldots w_n$ are found as follows:

$w_1 = \mu d_1(D) \ \vee \mu b_1(B)$

$w_2 = \mu d_2(D) \ \vee \mu b_2(B)$

………………………..

$w_n=\mu d_n(D) \lor \mu b_n(B)$

Competition is then assessed for each rule as follows:

$\mu c'_1 (C) = w_1 \lor \mu c_1$

$\mu c'_2 (C) = w_2 \lor \mu c_2$

………………………

$\mu c'_n (C) = w_n \lor \mu c_n$

These rules are aggregated for the final competition evaluation:

$$\mu c(C) = \mu c'_1(C) \wedge \mu c'_2(C) \wedge \ldots\ldots\ldots\ldots\ldots\ldots \wedge \mu c'_n(C) \qquad (5.5)$$

A definite value of competition is found by applying centre of gravity of the fuzzy set in equation 5.5 as follows;

$$C = \frac{\int \mu c(C) C dC}{\int \mu c(C) dC} \qquad (5.6)$$

5.4.2 Bid Determination

The bid increment ΔP for the auction is calculated based on attitudes and competition by applying Mamdani's Method for fuzzy relations and the compositional rule of inference. Let ΔP have the fuzzy set of values $p_1, p_2, \ldots\ldots p_n$, E is attitudes with a fuzzy set of values as $e_1, e_2 \ldots\ldots\ldots e_n$ and C is competition with a fuzzy set of values as $c_1, c_2, \ldots\ldots c_n$.

Here, *premise 1:* *IF c is C and e is E THEN p is* ΔP

premise2: *c is C' and e is E'*

consequence: *p is ΔP'*

where *C, C', E, E', ΔP* and *ΔP'* are fuzzy sets. As per the mechanism of fuzzy reasoning, we infer " *p is ΔP'*" when the condition " *c is C' and e is E'* " is given for the rule " *IF c is C and e is E THEN p is ΔP*".

Using a fuzzy relations approach, we first convert the *IF-THEN* rule in *premise 1* into the fuzzy relation $R_{C\ and\ E \rightarrow \Delta P}$. Then, by applying compositional operation, we infer conclusion *ΔP'* from the fuzzy relations $R_{C\ and\ E \rightarrow \Delta P}$ and the condition "*c is C' and e is E'*" of *premise 2* (Figure 5.5).

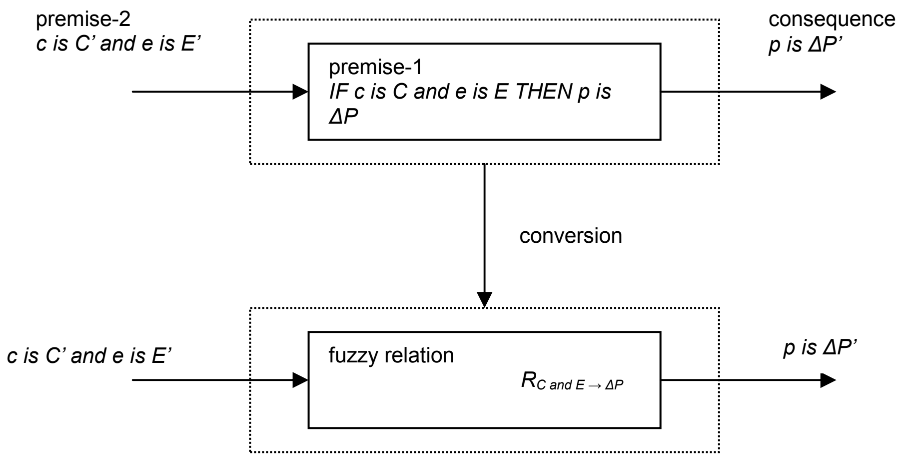


Figure 5.5 Fuzzy reasoning by using fuzzy relations and the compositional rule of inference

According to Mamdani's Method for fuzzy relations and the compositional rule of inference, the rule e_i *and* $c_j \rightarrow p_k$ is described by:

$$R = \int_{E \times C \times \Delta P} \frac{\left(\mu e_i(E) \wedge \mu c_j(C) \wedge \mu p_k(\Delta P)\right)}{(E, C, \Delta P)} \tag{5.7}$$

The conversion in Eq. (5.7) is based on a Cartesian product such as:

$$e_i and c_j \rightarrow p_k = e_i \wedge c_j \wedge p_k \tag{5.8}$$

The conversion by using the membership value form is given as follows:

$$\mu R(E, C, \Delta P) = \mu e_i(E) \wedge \mu c_j(C) \wedge \mu p_k(\Delta P) \tag{5.9}$$

For *n* number of rules, the compiled fuzzy relation *R* is given as:

$$R = R_1 \cup R_2 \cup .... \cup R_n = \bigcup_{i=1}^{n} R_i \tag{5.10}$$

For the input of fuzzy sets *E'* on *E* and *C'* on *C*, the output fuzzy set *ΔP'* on *ΔP* can be obtained as follows:

$$\Delta P' = (E' and C') \circ R = E' \circ (C' \circ R) = C' \circ (E' \circ R) \tag{5.11}$$

The bid amount for the auction is calculated as:

$$Bid\_Amount = p_i + \Delta P' \tag{5.12}$$

The fuzzy set E' depends on the bidding strategy selected for the bidding agent. *Ambitious* bidders always have a higher attitude to win the auction than *Sophisticated* bidders because the *Ambitious* bidders are desperate to get the item. Accordingly, the fuzzy set E' is described such that E is high for *Ambitious* bidders and low for *Sophisticated* bidders. However, *Sophisticated Strategic* bidders have a higher attitude to win the auction than the *Mystical* and *Sturdy* bidders of similar type, so *E* is also high for *Sophisticated Strategic* bidders.

5.5 Evaluating Bidding Strategies

The bidding strategies based on the NDFs and fuzzy reasoning were evaluated separately by performing two sets of experiments, as described in the following subsections:

5.5.1 Analysing the Negotiation Decision Function-based Bidding Agent

The ADBA system generates two types of agents for evaluating the bidding strategies based on *NDFs*: *NDF-DC* and *NDF-D*. *NDF-DC* agents design the bidding strategies using the *NDFs* in Section 5.3, which depend on both the $F(t)$ and $F_c(t)$. Here *DC* refers to the agents that calculate the bid amount using the remaining *D*uration of the auction and *C*ompeting bids. *NDF-D* agents design bidding strategies using *NDFs*, which only depend on $F(t)$ and not on $F_c(t)$. Here *D* refers to agents that calculate the bid amount using the remaining *D*uration of the auction. In order to evaluate the performance of both *NDF-DC* and *NDF-D* agents in a wide variety of test environments, the agents were subjected to different action settings and to different bidding restrictions (bargain level). In this set of experiments, to compute $F(t)$, values for k and β were chosen in line with the discussion in Section 5.3 To compute the function $F_c(t_{n+1})$, the initial values of $F^{'}(t_{n-2\delta+2})$, $F^{'}(t_{n-2\delta})$ and $F(t_{n-1})$ were calculated at $\delta=1$ for all the bidding strategy types i.e. *Mystical, Sturdy* and *Strategic*. For *Mystical*, *Sturdy* and *Strategic* behaviours, $\delta=1$ at the beginning of the last five seconds, the last five minutes and the last hour of the auction respectively. Initial values of $F^{'}(t_{n-2\delta+2})$, $F^{'}(t_{n-2\delta})$ and $F(t_{n-1})$ were assigned from the bid history of the auction in which the bidders were then participating. The current maximum bid was assigned to $F^{'}(t_{n-2\delta+2})$, and the previous bid was assigned to $F(t_{n-1})$ followed by $F^{'}(t_{n-2\delta})$.

As discussed above, two types of attitudes of bidders were considered: *Ambitious* and *Sophisticated*. Bidders with an *Ambitious* attitude start bidding at a higher price close to their reservation value p_r and their bid amount is not so affected by the bid amounts placed by the other bidders due to their desperate behavior. *AMST, ASTD* and *ASTG* bidding strategies follow this type of attitude. Bidders who are willing to bargain always bid strategically based on the bids placed by the other competitors. *SMST, SSTD* and *SSTG* follow this type of attitude. As the bidding strategies described above select bids based on the remaining time as well as on the bids placed by the other participants, these strategies will be successful when the bidder has a desire to bargain attitude. Thus, we need to evaluate the performance of agents who act strategically based on the bids placed by the other participants, i.e. bidders with a *Sophisticated* attitude.

5.5.2 Analysing Fuzzy Reasoning-based Bidding Agents

Fuzzy agents in the ADBA system design bidding strategies using fuzzy reasoning, as explained in Section 5.4. These *Fuzzy* agents were evaluated against the *NDF-DC* agents in a similar auction environment to that of the first set of experiments. In order to compute ΔP, linguistic variables for the bidder's attitude and competition assessment were chosen, as discussed in Section 5.4. The bidding strategies were analysed by considering the following sets of logical rules using various fuzzy sets:

Rule 1: IF the attitude of the agent to winning the auction is E_1 AND competition on the market for that product is C_1, THEN the bid increment will be P_1

Rule 2: IF the attitude of the agent to winning the auction is E_1 AND competition on the market for that product is C_2, THEN the bid increment will be P_2

Rule 3: IF the attitude of the agent to winning the auction is E_2 AND competition on the market for that product is C_1 ,THEN the bid increment will be P_2

Rule 4: IF the attitude of the agent to winning the auction is E_2 AND competition on the market for that product is C_2 ,THEN the bid increment will be P_3

These fuzzy sets represent the linguistic variables as follows: attitudes low as E_1 and high as E_2, competition low as C_1 and high as C_2. P_1, P_2 and P_3 were the bid increments based on the characteristics of the auction, where $P_3 \geq P_2 \geq P_1$. We assumed that the set of attitudes for buying any item was $E=[e_1,e_2,e_3]=[0,0.5,1.0]$, and the set of competition for the item on the market was $C=[c_1,c_2,c_3]=[0,0.5,1.0]$. The fuzzy sets used in the preceding four rules can be quantized as shown in Figure 5.6:

$E_1=[1.0,0.5,0]$ $C1=[1.0,0.5,0]$ $P_1=[1.0,0,0]$

$E_2=[0,0.5,1.0]$ $C2=[0,0.5,1.0]$ $P_2=[0,1.0,0]$

$P_3=[0,0,1.0]$

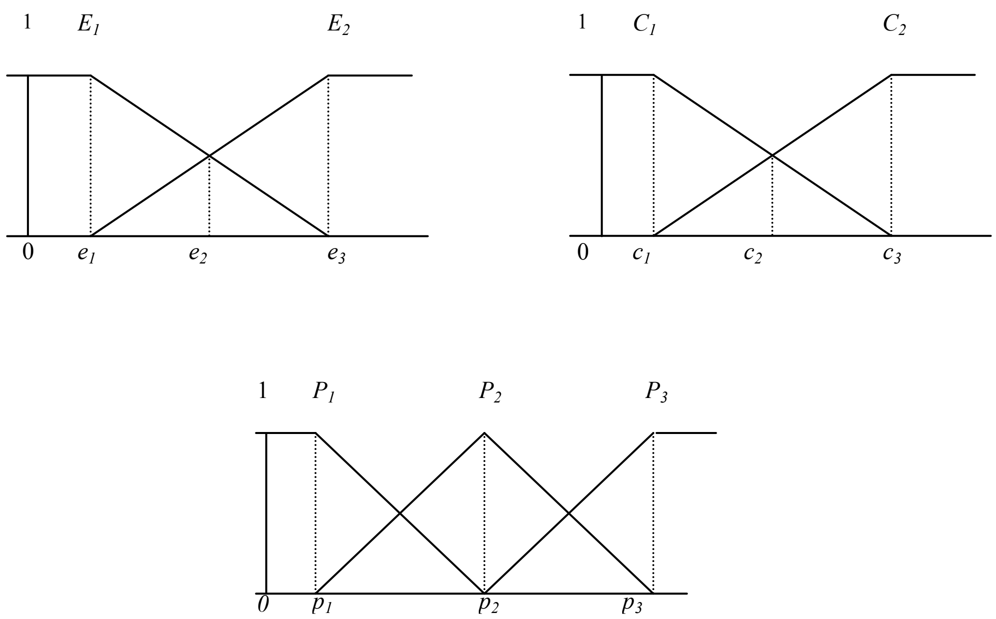


Figure 5.6 Fuzzy sets for bidding logic

For all these bidding strategies, competition was assessed based on the remaining duration and the bids placed by other participants (competing bids), as described in Section 5.4. These bidding strategies will be successful when the bidder has a desire to bargain attitude, in a similar vein to the situation with the bidding strategies using *NDFs*. This set of experiments, thus, needed to address the performance of bidding agents with a *Sophisticated* attitude who act strategically based on the bids placed by the other participants.

5.5.3 Performance Measures

The performance measures were the success rate percentage and the expected utility of the bidding agents.

Success rate percentage = $R_{success} = r_{success}*100$

where $r_{success}$ is the success rate and $r_{success} = N_{win} / N_{total}$, N_{win} is the number of auctions won by the agent and N_{total} is the total number of auctions.

Expected utility= $U_{exp} = U_{win} * r_{success}$

where U_{win} is the utility of the winning agent.

and $U_{win} = (p_r-v_i)/p_r$, where p_r is the reservation price (maximum amount the bidder is willing to pay) and v_i is the winning price of the auction.

5.5.4 Empirical Assessment of Bidding Strategies

A simulated electronic marketplace was developed and experiments were conducted separately for heterogeneous and homogeneous bidding agents when assessing the bidding strategies based on negotiation decision functions and fuzzy reasoning techniques. In this research, heterogeneous bidders have random or varying willingness-to-bargain levels, here called 'bargain levels'. In contrast, all homogeneous bidders have identical bargain levels.

These bidding strategies were assessed separately in different settings, i.e. low-, medium- and high- bid rate auctions for each bidding agent acting for the heterogeneous bidders. *NDF-D* and *NDF-DC* agents with varying bidding strategies competed against one another for each type of auction separately.

Bidding agents with various bidding strategies participated in each type of auction separately. Experiments were run 50, 100, 150 and 200 times to test and verify the statistical significance of the results. Details of these experiments are provided using Analysis of variance (ANOVA) in Chapter 6 of this thesis.

These experiments were also carried out separately for each type of homogeneous bidding agent in the different bidding environments, in auctions with various bid rates. To evaluate the bidding strategies, the U_{exp} of *NDF-D* and *NDF-DC* bidding agents was measured in two situations: firstly, against the various bid rates of the auctions, and secondly, against the different bargain levels of bidders.

The fuzzy bidding strategies were assessed for heterogeneous bidding agents in different bidding environments, i.e. auctions of the various bid rates. *Fuzzy* and *NDF-DC* agents with various bidding strategies competed against one another separately in each type of auction. Again, the experiments were run several times to test and verify the success rate and expected utility of these bidding agents.

5.6 Summary

This chapter has proposed a NDF-based technique to design bidding strategies for bidders. This technique aims to forecast bid amounts at a particular point in time based on the different bidding behaviours of buyers. Bidding strategies have been designed that emphasise bidding characteristics such as an auction's attributes, the bidder's own attitude to winning the auction and other bidders' 's behavior. The design has drawn on time- and behaviour- dependent negotiation decision functions; it has also invoked Mamdani's Method for fuzzy relations and the compositional rule of inference. This represents an attempt to handle vagueness in the value of the predictor variables in the uncertain environment of online auctions. This chapter has also set out a methodology for evaluating these approaches to designing bidding strategies. The following chapter proceeds with this evaluation in the context of a simulated electronic marketplace.

Chapter 6

Evaluating the Bidding Strategies Using a Simulated Marketplace

This chapter evaluates the bidding strategies designed in Chapter 5. It describes the use of a simulated electronic marketplace to implement the automated dynamic bidding agent (ADBA) framework. The discussion is set out as follows: Section 6.1 presents the design of the simulated electronic marketplace. Section 6.2 outlines the experiments which were carried out to evaluate the bidding strategies designed for bidders. Section 6.3 concludes with a brief summary of the achievements and contributions of this chapter.

6.1 A Simulated Electronic Marketplace for Online Auctions

A simulated electronic marketplace was set up to implement the ADBA framework and thus demonstrate the performance of the bidding strategies developed for buyers in this thesis. This simulation also gave effect to various agents, who could interact with each other and play a range of roles.

6.1.1 Market Architecture

The simulated electronic marketplace which was created by this study hosts online auctions where buyers can bid on and purchase items. It is an environment where buyers can negotiate using different bidding strategies rooted in their attitudes, all with the aim of winning an auction.

Figure 6.1 depicts the top-level conceptual client-server architecture of the electronic marketplace, illustrating the types of agents proposed for this domain and their interactions. The electronic auction market is managed by an auction server and can be used by various buyers. The auction server is implemented at the server end with *Administrator* agent and it receives information from *Initial Price Estimator* agent and auction database. Bidders are entered into the system at the client end using *ADBA Bidder* agents.

The *Initial Price Estimator* agent searches for a target auction by assessing the value of the auctioned item using the clustering approach and bid mapping and selection technique that are presented in Chapter 4. This agent is also responsible for providing information about the target auction to the *Administrator* agent. Such information includes the auction's type and identification (ID) details, the item's reserve price (set by the seller), the minimum bid increment, item (product) information, a product image, the bid history, participating bidders, time remaining, the current highest bidder and current maximum bid and the assessed value of the item.

The *Administrator* agent maintains the auction information provided by the *Initial Price Estimator* agent. Whenever bidder registers with the *Administrato*r agent to buy an item in an auction, the *Administrator* agent creates an *ADBA Bidder* agent based on the bidding behaviour of the bidder. The *Administrator* agent is also responsible for maintaining information about all registered bidders in the auction. It sends the information about the target auction to all the registered *ADBA Bidder* agents. The bidder agents compete to win in the auction by sending their bids to the *Administrator* agent. The *Administrator* agent updates participating *ADBA Bidder* agents with the current maximum bid in the auction. At the end of the auction, the *Administrator* agent declares the winner.

An *ADBA Bidd*er agent is responsible for placing bids automatically on behalf of its bidder in the target auction. Each bidder configures his *ADBA Bidder* agent according to his bidding strategy. The bidder agent computes and sends its maximum willingness to pay, at a particular moment in time, to the *Administrator* agent based on the current maximum bid in the auction and its bidding behavior.

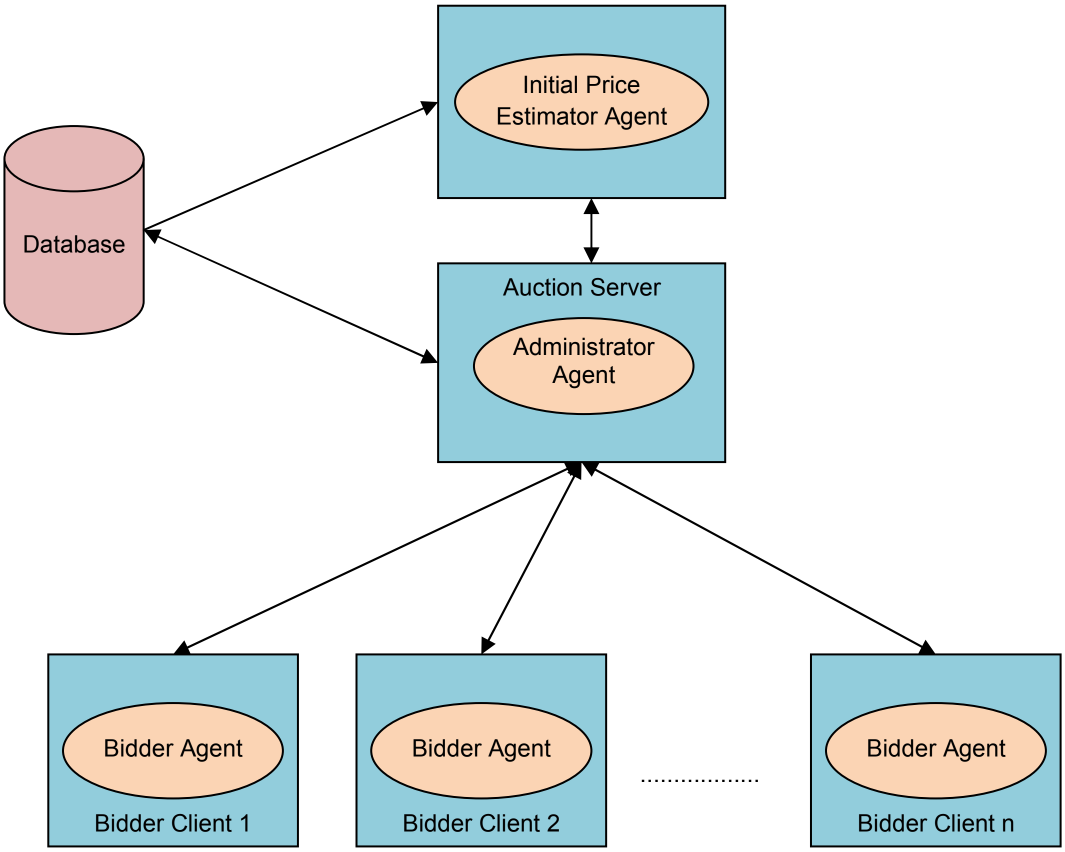


Figure 6.1 Architecture of the simulated electronic marketplace

6.1.2 The User Interface Module

Sellers and buyers launch a session in the system by creating the *Initial Price Estimator* agent, the *Administrator* agent and the *ADBA Bidder* agent via the graphical user interfaces for the simulated electronic marketplace and bidder agents.

The simulated electronic market interface is designed to provide complete information about a target auction, including its type, its ID, the item reserve price (set by the seller), the minimum increment, product (item) information, the product image, the bid history, participating bidders, time remaining, the current highest bidder and current maximum bid, the predicted closing price and the number of iterations for simulation run (Figure 6.2). This user interface has the following five panels: Auction information, Product description, Bid history, Participants and Current status. The information about the auction and its participants can be loaded from an XML file in the database or from the user interface for ADBA bidders and the bid history. The interface receives commands from the bidders and acts accordingly. It is responsible for updating the auction information when messages are received from other agents.

The user interface for *ADBA bidder* agents is designed to record information about each bidder's characteristics, which are defined by attributes including their name, bidder type, bidding preferences, bidder behaviour, bidding attitude and level of desire for bargain (Figure 6.3). Bidders select and enter bidding strategies according to their preferred bidding behaviour. This behaviour is distinguished according to the different characteristics of auctions and the bidder's own characteristics submitted via the user interface modules for the simulated electronic marketplace and the *ADBA bidder* agents. It will be helpful to look more closely now at what happens in the marketplace from the point in time when bidder makes a request to find and buy a particular item until this request is completed.

Following the bidder's entry in the system, an *ADBA Bidder* agent is registered with the *Administrator* agent. This *ADBA Bidder* agent obtains a new ID if it represents a new buyer; it recovers its original ID if it belongs to a returning buyer. The information that an agent with a given ID is active in the marketplace is stored in the *BidderList* database. (This step involves interactions between the *Administrator agent* and *BidderList*).

The *ADBA Bidder* agent requests that the *Administrator* agent search for an auction of a particular item. The *Administrator* agent relays this request to the *Initial Price Estimator* agent. The *Initial Price Estimator* then searches for a list of active auctions of the item and selects a target auction to participate in along with the assessed value of the item on auction; a clustering approach and bid mapping and selection technique are used for this purpose. The *Initial Price Estimator* sends the details of the target auction to the *Administrator* agent, who provides this information to all *ADBA Bidder* agents.

The *Administrator* agent checks periodically for the set of *ADBA Bidder* agents that are registered to bid for the item. If this set is nonempty, it will start an auction. During the auction, *ADBA Bidder* agents compete with each other to win the auction using different bidding strategies according to their set bidding behaviour. At the end of the auction, the *Administrator* agent informs the *ADBA Bidder* agents about the winner.

Figure 6.2 User interface for simulated electronic marketplace

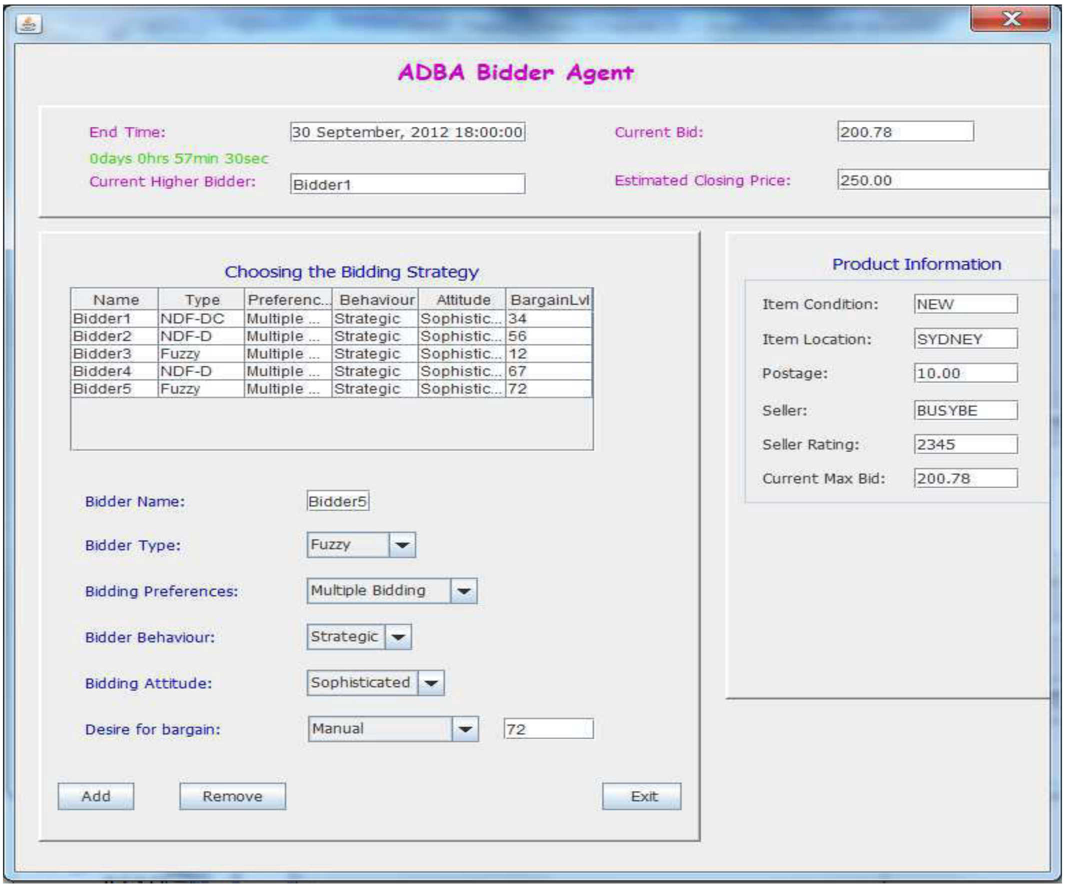


Figure 6.3 User interface for *ADBA bidder* agent

6.1.3 Market Implementation

The simulated electronic marketplace was implemented using JADE - Java Agent DEvelopment Framework environment, an optimal modern agent environment (Bellifemine et al. 2005). JADE is an open source software environment fully implemented in JAVA language; it has been designed with the aim of developing multi-agent systems that comply with Foundation for Intelligent Physical Agents (FIPA) specifications.

The JADE platform identifies three major system agents: the Agent Management System (AMS), the Agent Communication Channel (ACC) and the Directory Facilitator (DF). The AMS agent has supervisory control over access to and use of the platform; it is responsible for authenticating resident agents and registration control. The ACC provides a path for basic contact between agents inside and outside the platform; it is a default communication method and offers a reliable, orderly and accurate message routine service. It needs to support IIOP to allow interoperability between the different agent platforms. The DF agent provides a yellow pages service for the agent platform.

The main container is created in JADE. This hosts the *Administrator* agent, which creates further bidder agents to compete in the auction (Figure 6.4 and Figure 6.5). The simulated electronic marketplace can be extended, however, for use in a distributed environment by creating bidders in different JADE containers instead of the main container. The *Administrator* agent receives complete information about the target auction from the *Initial Price Estimator* agent, which is implemented using the XLMiner data mining tool (Frontline Systems Inc 2013).

```
// Main container hosting the Administrator Agent.

public class AuctionAgent extends GuiAgent{
        ...
        ...
        public AgentContainer myContainer;        // main container
        ...
        ...
}

@Override
protected void setup(){             // Automatically called on construction.
        ...
        ...
        myContainer = getContainerController();
        ...
        ...

}
```

Figure 6.4 Main container hosting the *Administrator agent*

```
// Creating bidder agents

private void CreateAgents() {
        for(int i=0; i<biddersList1.size();i++ ){
        ...
        ...
        CreateAgent(i);

        ...
        }
}

public void CreateAgent(int i){
        ...

        AgentController a = myContainer.createNewAgent(
                        biddersList1.get(i).getName(), "BiddingAgent.BidderAgent", args );
        a.start();
        ...
        ...
}
```

Figure 6.5 Creating *ADBA Bidder* agents

The *Administrator* agent and *ADBA Bidder* agents are run using the Java agent development framework, and they react to events in the simulated electronic marketplace using JADE behaviours. These events consist of receipt of a message. The behaviour in JADE basically identifies the agent's working mechanism during an event. The event handling code is defined in the action() method for the behaviour. These methods are executed one after another through the agent's thread after an event. This cannot pause without blocking all other activities within the agent, and the execution of each behaviour corresponds to a single instantaneous active phase. The behaviours are scheduled following a round robin algorithm. Rather than focusing on the writing of multithreaded agents, JADE introduces system behaviour to assist in building and reusing agent functionality. JADE behaviours are a set of cooperatively multithreading processes which run and perform a portion of a task before giving control back to the process scheduler. These are accessed using two methods: addBehaviour(), which adds a new behaviour to the set of running behaviours, and removeBehaviour() , which stops and removes a behaviour of the agent.

These agents are implemented using the JADE class "CyclicBehaviour". The behaviour stays active as long as its agent is alive, which is essential for handling the bids (messages) exchanged between these agents. *block()* is used to put the behaviour on hold or wait until the next message is received. After receiving the message, the behaviour is activated. The implementation processes of the *Administrator* agent and the *ADBA Bidder* agent are shown in Figure 6.6 and Figure 6.7 respectively.

```
// Implementation of Administrator's agent's action method.
class MyBehaviour extends CyclicBehaviour{

        @Override
        public void action() {
        ...
        ...
        ...
        }
}

// Adding behaviour to Admistarter Agent.
@Override
protected void onGuiEvent(GuiEvent ev)
{
        ...
        ...

        case Event3:
                ...
                ...
                addBehaviour(new MyBehaviour());
                break;

}
```

Figure 6.6 Implementation of the *Administrator agent*

```
// Adding behaviour to the bidder agent and implementing its action method.
public class BidderAgent extends Agent{

  ...
  ...

   @Override
   protected void setup()
   {
         ...
         ...
         ...

         // Create the behaviour of the agent.
         addBehaviour(new CyclicBehaviour(this){

                  @Override
                  public void action(){
                           ...
                           ...
                           ...
                  }//action()
         }
}
```

Figure 6.7 Implementation of the *ADBA Bidder* agent

The main states and transition conditions from one state to another of the *Administrator* agent are explained below (Figure 6.8):

Initiated: Agent is initiated.

Configure: Agent's properties are set as per the given auction information.

Active: Ready to send and process messages.

Wait: Agent behaviour is on hold until it receives bids from all *ADBA Bidder* agents.

Ended: Agent is finished.

The *Administrator* agent launches in an *Initiated* state. It is ready to create an auction according to the information given by the *Initial Price Estimator* agent. The *Administrator* agent accepts the information and designs an auction in the *Configure* state accordingly. If an event requiring it to start bidding (*startBidding()*) is thrown to the agent, then it enters into an *Active* state. Alternatively, it remains in the *Configure* state. In the *Active* state, the agent is ready to send and process the message (*send(msgToBidders)*). The messages from the *Administrator* agent may include announcements of the current highest bid or the winner of the auction. A transition from the *Active* state to a *Wait* state occurs after the messages are sent to bidders. In the *Wait* state, the agent receives messages from all bidders in the auction (*recFromAll()*). These messages include individual bid amounts at a particular moment according to the different bidding strategies for buyers which were designed in Chapter 5. The *Administrator* agent remains in the *Wait* state until it receives bids from all participating bidders. After receiving bids from all the bidders, the agent enters into the *Ended* state if the auction's end-time is reached ($t=t_{max}$) and the winner of the auction is declared. Alternatively, it enters into the *Active* state to process the bids from the bidders ($t!=t_{max}$) in the ongoing auction.

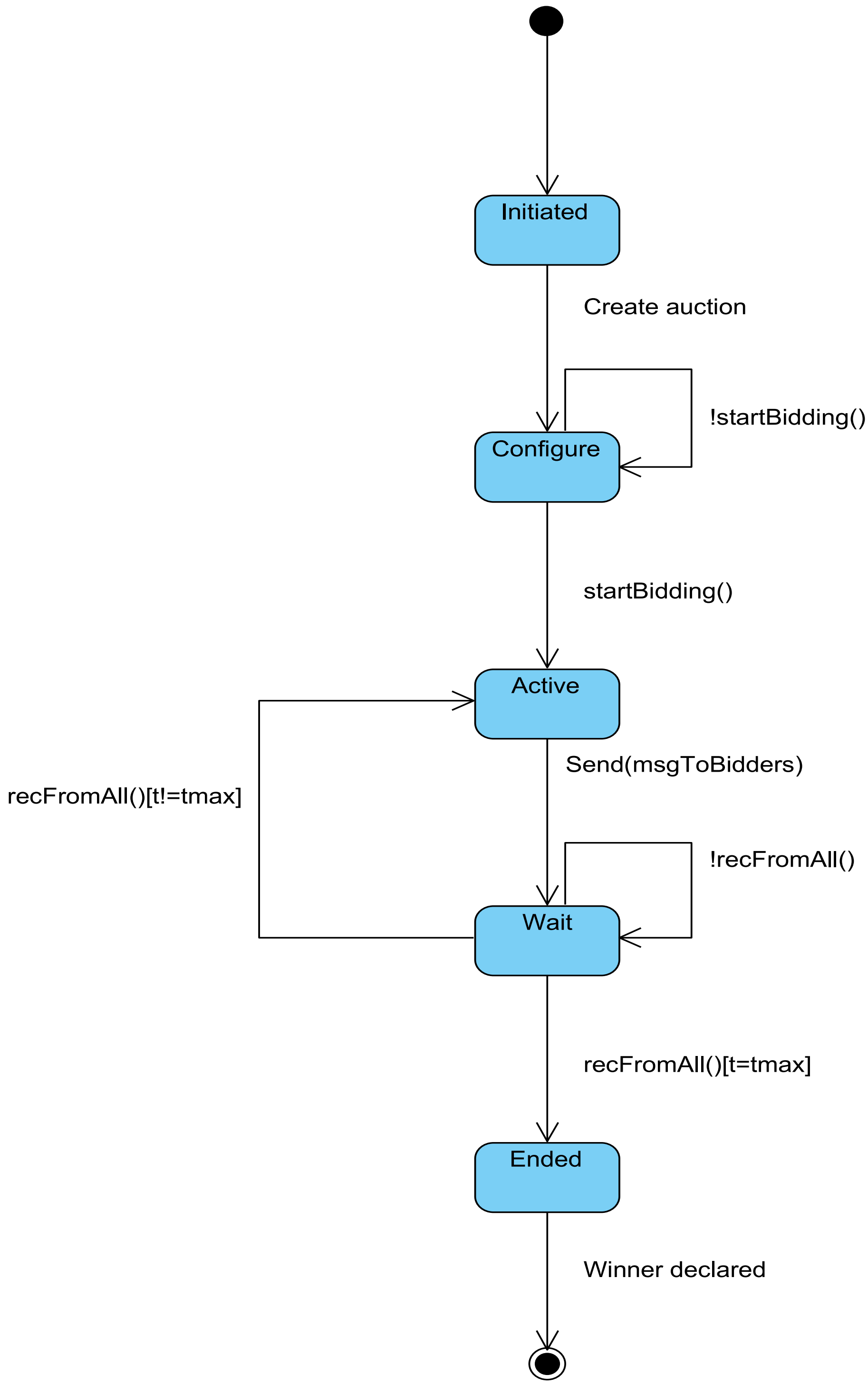


Figure 6.8 *Administrator agent* state diagram

The main states and transition conditions from one state to another for the *ADBA Bidder* agent are explained below (Figure 6.9):

Initiated: Agent is initiated.

Configure: Agent's properties are set as per the bidding strategies of a bidder.

Active: Ready to send message.

Wait: Agent behaviour is on hold until it receives current maximum bid from the *Administrator* agent.

Ended: Agent is deleted.

The *ADBA Bidder* agent begins in an *Initiated* state. It is ready to bid according to its bidding strategy. It accepts the bidder's characteristics submitted through the user interface of the *ADBA Bidder* agent or the *Administrator* agent, and designs an *ADBA Bidder* agent in the *Configure* state accordingly. If an event requiring it to start bidding (*startBidding()*) is thrown to the *ADBA Bidder* agent, then it enters into an *Active* state. Alternatively, it remains in the *Configure* state. In the *Active* state, the agent is ready to generate and send a new bid (sendNewBid()) according to the chosen bidding strategies for the *ADBA Bidder* agent that were designed in Chapter 5. This new bid is the maximum amount that the agent is willing to pay for the item at that particular moment in time. A transition occurs from the *Active* state to the *Wait* state after the new bid is sent to the *Administrator* agent. In the *Wait* state, the *ADBA Bidder* agent receives messages from the *Administrator* agent (msgReceive()). The messages may include an announcement of the current maximum bid in the auction or its winner. The agent remains in the *Wait* state until it receives a message from the *Administrator* agent. After that message is received, the *ADBA Bidder* agent enters into the *Ended* state if the auction's end-time is reached ($t=t_{max}$) and the winner of the auction is declared. Alternatively, it enters into the *Active* state to generate and send a new bid according to its bidding strategy ($t!=t_{max}$).

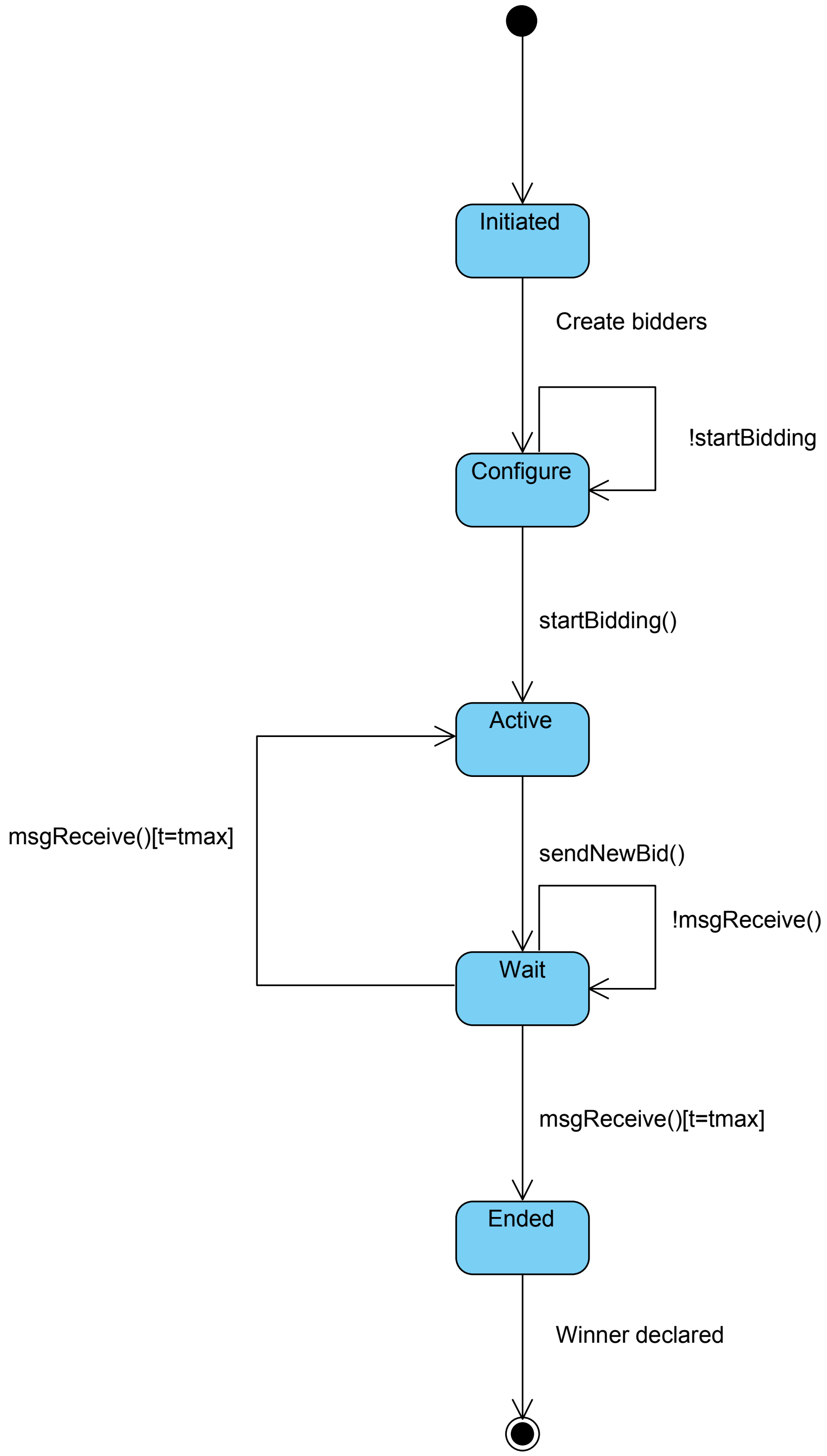


Figure 6.9 *ADBA Bidder agent* state diagram

6.1.4 Running the System

There are three types of *ADBA Bidder* agents that participate in this marketplace; *NDF-DC*, *NDF-D* and *Fuzzy*. The bidding strategies presented in Section 5.3, Chapter 5 are implemented by *NDF-DC* bidder agents. *NDF-D* bidder agents follow bidding strategies which depends only on the *F(t)*. The *Fuzzy* bidder agents adopt the bidding strategies in Section 5.4.

Bidders select bidding features defined by own attributes such as their name, bidder type, bidding preferences, bidding behaviour, bidding attitude and level of desire for bargain (Figure 6.3). Bidding preferences determine whether bidding agents make Single or Multiple bid placements, and the *Mystical, Sturdy* and *Strategic* bidding strategies are assigned based on the bidding behaviour attributes. Bidders' attitudes to bargaining, i.e. whether they have desire to bargain or they are desperate to get an item, are recorded as *Sophisticated* and *Ambitious* bidding attitudes, respectively.

The level of desire to bargain can be set manually or assigned randomly as an integer between 1 and 100 inclusive for *NDF-DC* and *NDF-D* bidders, where 1 represents the minimum and 100 represents the maximum level of desire to bargain in an auction to get an item. The constants k and β are determined from the level of desire to bargain for different bidding preferences and bidding behaviours of these bidders. The algorithm to calculate these constants is given in Figure 6.10. For *Fuzzy* bidders, the level of desire to bargain to get an item in an auction is expressed as the fuzzy attitude E of bidders to win the auction. Each participating bidder agent is assigned their private valuation or the maximum amount they are willing to pay as the predicted closing price of the auction based on the initial price estimation in Chapter 4.

```
beh ← bidding behaviour of the bidder
att ← bidding attitude of the bidder
betaArray1 ← [0.2, 0.3, 0.6, 0.9]
betaArray2 ← [10.00, 20.00, 30.00, 40.00]
b← number of bidders
bLvl ← bargain level of the bidder

while b≠ 0 do
        if att = "Sophisticated" then
              β ←betaArray1[(100 - bLvl)/25]
              if beh = "Strategic" then
                    k ← (100.0 - bLvl/100)*0.6
              else
                    k ← (100.0 - bLvl/100)*0.3
              end if
        else
          k ← ((100.0 - bLvl/100)*0.4)+0.6
          β ←betaArray2[(100 - bLvl)/25]
        end if
        b ← b-1
end do
```

Figure 6.10 Algorithm for calculating bidder constants

During the simulation process, the *NDF-DC*, *NDF-D* and *Fuzzy* Bidder agents compete to obtain a given item in the simulated marketplace. Simulation is completed in two phases: preparation and execution.

6.1.4.1 Preparation Phase

This phase involves the initialisation of the auction server by setting up the parameters required for the marketplace. Here only English-type auctions are considered. Ongoing auctions are generated for participation based on previously observed eBay auctions. Each simulated auction sells a single item and is assigned its own reservation price. The item will be sold to the winner if the winning bid exceeds the reservation price of the item set by the seller. In the experiments, these auctions were categorised according to the values of their bid rates.

The number of simulated auctions that run simultaneously for the same item can be set as an integer from 1 to 100 (inclusive). The number of bidders (*NDF-DC, NDF-D* or *Fuzzy*) in the marketplace can be assigned as an integer from 1 to 2000 (inclusive) while an individual auction can register a maximum of 20 bidders. The characteristics of these bidders are set on the basis of their attributes.

6.1.4.2 Execution Phase

The bidders select an auction to participate in, and they register for that auction. Each participating bidder has the same primary objective: to get the desired item at the minimal price. They all proceed to bid in the auction according to their attributes. Each auction has an active period consisting of a specified number of discrete and indivisible time steps for bidding in the auction. Within the active time steps, the *NDF-D, NDF-DC* and *Fuzzy* bidders bid to become the leading bidder in the auction. The auction server processes these bids and sends the message to participating bidders accordingly. The message may contain the current maximum bid or the name of the winner of the auction depending upon the bid amounts and the current time step.

6.2 Evaluating the Bidding Strategies

Two sets of experiments were performed separately to evaluate the NDF-based bidding agents and the fuzzy reasoning-based bidding agents. Details of these experiments are given in the following subsections:

6.2.1 Evaluating the Negotiation Decision Function-based Bidding Agents

In the first set of experiments, heterogeneous and homogeneous bidder agents were examined separately to evaluate the NDF-based bidding strategies. In this research, the heterogeneous bidders had random or varying bargain levels. In contrast, the homogeneous bidders had identical bargain levels.

6.2.1.1 *Experiments with Heterogeneous Bidders*

Bidding strategies were assessed separately in different settings of low-, medium- and high- bid-rate auctions for each bidding agent serving heterogeneous bidders. *NDF-D* bidder agents and *NDF-DC* bidder agents with different bidding strategies competed against one another. This was repeated separately for auctions with each type of bid rate.

For each type of auction, the bidding agents with various strategies were run independently. Experiments were run 50, 100, 150 and 200 times to test and verify the statistical significance of the results. For all the experiments, it was found that $p>0.05$ and $F < F_{crit}$, which shows that the results obtained are statistically significant (Table 6.1).

Table 6.1 Results of ANOVA test to compare the success rate means

Bidder Agent	***Bidding Strategy***	***Low-bid-rate auctions***			***Medium-bid-rate auctions***			***High-bid-rate auctions***		
		F	F_{crit}	*p-value*	F	F_{crit}	*p-value*	F	F_{crit}	*p-value*
NDF-D	***SMST***	1.276	3.490	0.327	0.221	3.490	0.879	0.903	3.490	0.06
NDF-DC	***SMST***	3.166	3.490	0.064	0.221	3.490	0.879	0.903	3.490	0.06
NDF-D	***SSTD***	0.023	3.490	0.995	0.012	3.490	0.998	-	-	-
NDF-DC	***SSTD***	0.023	3.490	0.995	0.523	3.490	0.675	-	-	-
NDF-D	***SSTG***	1.525	3.490	0.258	0.821	3.490	0.507	0.103	3.490	0.957
NDF-DC	***SSTG***	1.525	3.490	0.258	0.821	3.490	0.507	0.103	3.490	0.957

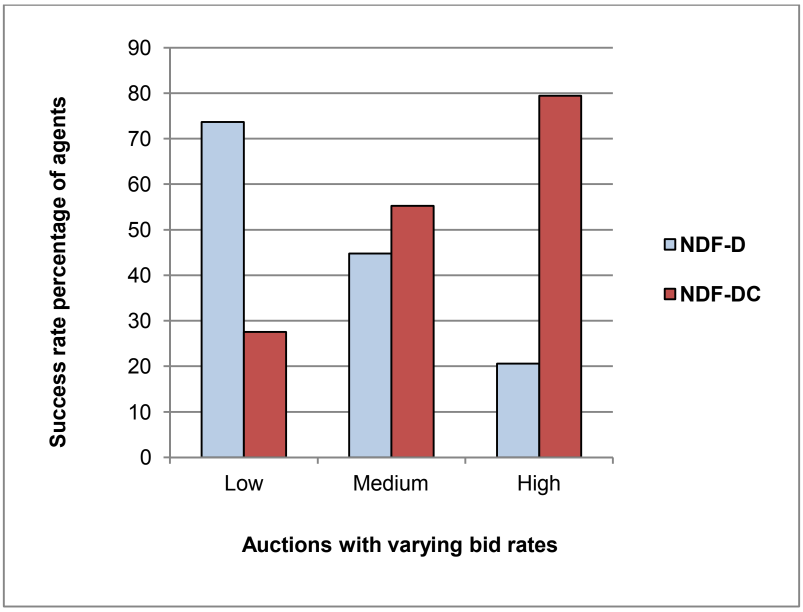


Figure 6.11 Success rate percentage comparison of NDF-based agents with Mystical behaviour

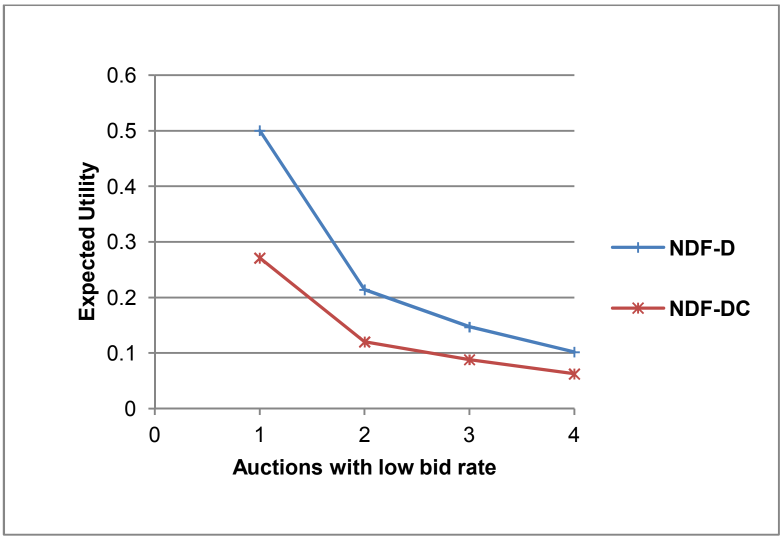


Figure 6.12 Expected utility comparison of NDF-based agents with Mystical behaviour in low-bid-rate auctions

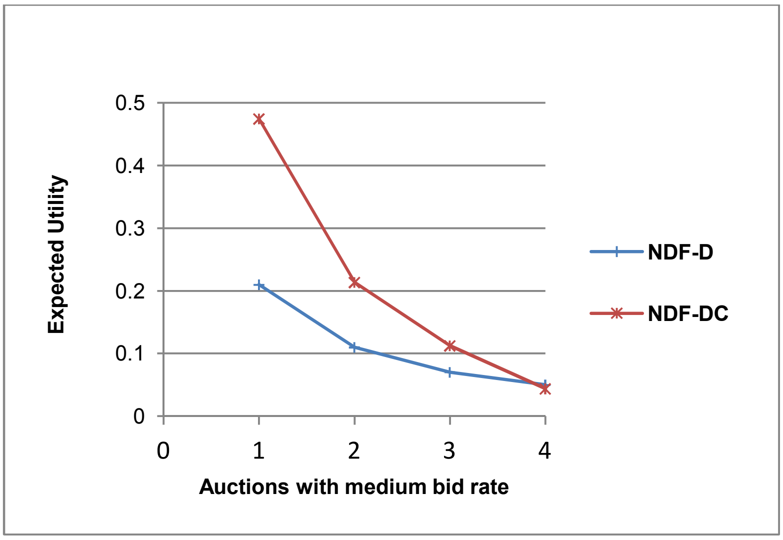


Figure 6.13 Expected utility comparison of NDF-based agents with Mystical behaviour in medium-bid-rate auctions

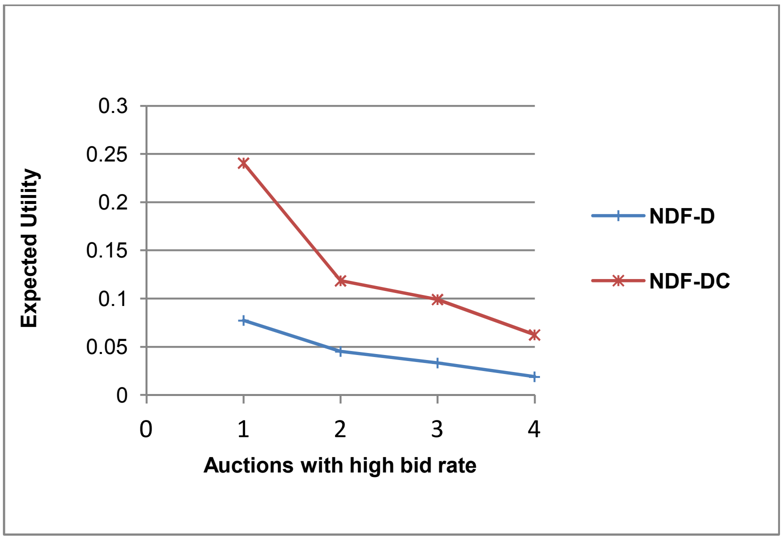


Figure 6.14 Expected utility comparison of NDF-based agents with Mystical behaviour in high-bid-rate auctions

From Figure 6.11, it can be seen that subject to varying bargain levels (heterogeneous bidders), the $R_{success}$ of *NDF-DC*s is clearly higher than that of *NDF-Ds* in situations when agents with *Mystical* behaviour compete in medium- and high-bid-rate auctions. However, in low-bid-rate auction settings, $R_{success}$ of *NDF-Ds* is higher. Similarly, U_{exp} of *NDF-DCs* is higher than that of *NDF-Ds* when agents with *Mystical* behaviour compete in medium- and high- bid-rate auctions (Figure 6.13 and Figure 6.14); in low-bid-rate auctions, the reverse is true for these agents and U_{exp} of *NDF-Ds* is higher (Figure 6.12). These results correspond with the intuition that in low-bid-rate auctions, the bid increments made by these *NDF-DCs* are limited. Agents with *Mystical* behaviour place bids in the closing five seconds of these auctions, i.e. at the point when all bidders' bids approach p_r. However, the *NDF-DC* agents' consideration of others' bids reduces their bid increments when the low bid rate of auctions also lowers the amounts that other participants bid.

The auctions in each set of experiments were arranged according to the bid amounts approaching the closing price of the auction. From the graphed results, it is evident that expected utility decreases as auctions approach the closing price. This is in line with expectations.

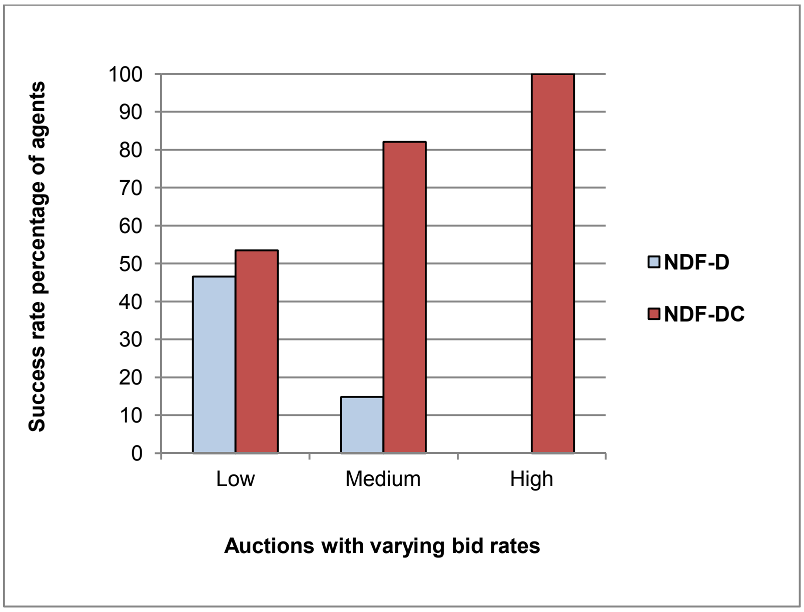


Figure 6.15 Success rate percentage comparison of NDF-based agents with Sturdy behaviour

$R_{success}$ of *NDF-DCs* is clearly higher than that of *NDF-Ds* when these heterogeneous agents with *Sturdy* behaviour compete in low-, medium- and high-bid-rate-auctions (Figure 6.15). Similarly, U_{exp} of these *NDF-DCs* is also higher than that of *NDF-Ds* competing in low-, medium- and high- bid-rate auctions (Figure 6.16 to Figure 6.18). In the high-bid-rate settings, $R_{success}$ and U_{exp} of the *NDF_Ds* are zero. This corresponds with the intuition that the bid increments made by these *NDF_Ds* are always lower than those of the *NDF-DCs*. Agents with *Sturdy* behaviour place bids at the beginning of the last five minutes of the auction, i.e. at a time-point when agents are under less pressure to reach p_r than they will be near the final moments of the auction. However, the *NDF-DC* agents' consideration of competing bids accelerates their own bidding when the high bid rate of an auction raises bid amounts.

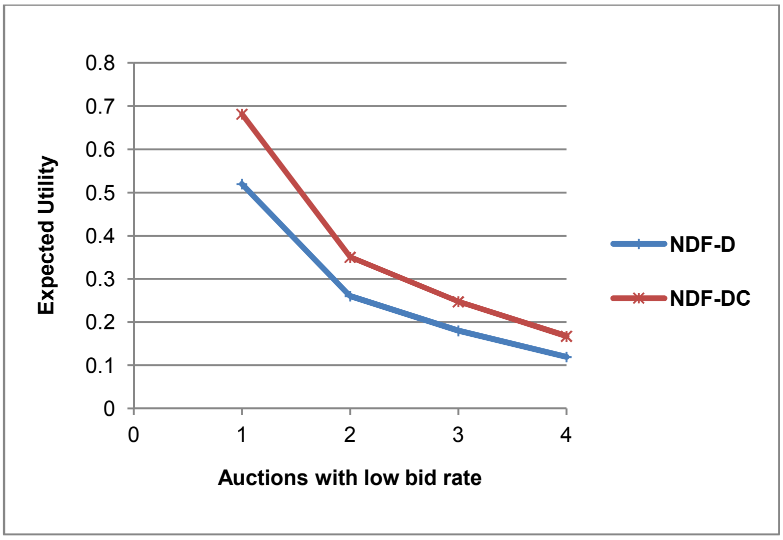


Figure 6.16 Expected utility comparison of NDF-based agents with Sturdy behaviour in low-bid-rate auctions

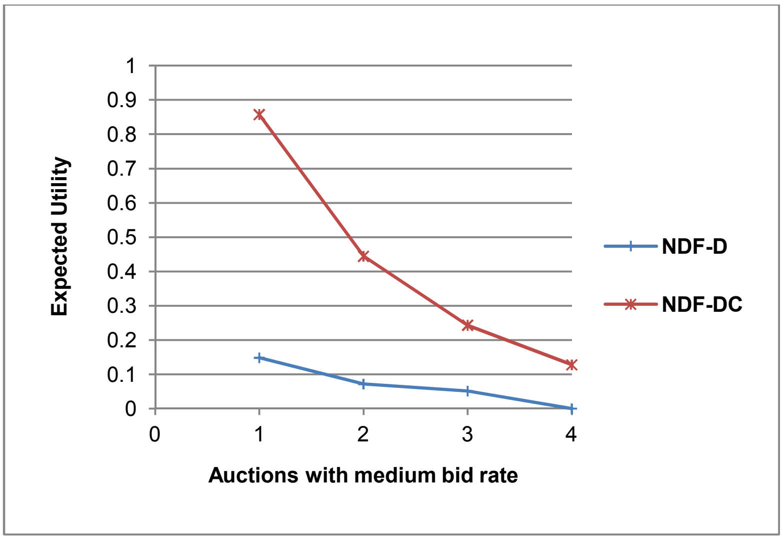


Figure 6.17 Expected utility comparison of NDF-based agents with Sturdy behaviour in medium-bid-rate auctions

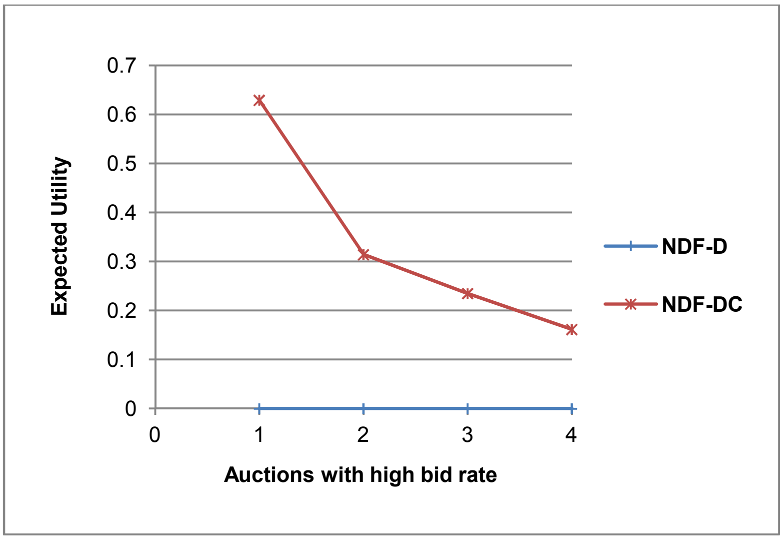


Figure 6.18 Expected utility comparison of NDF-based agents with Sturdy behaviour in high-bid-rate auctions

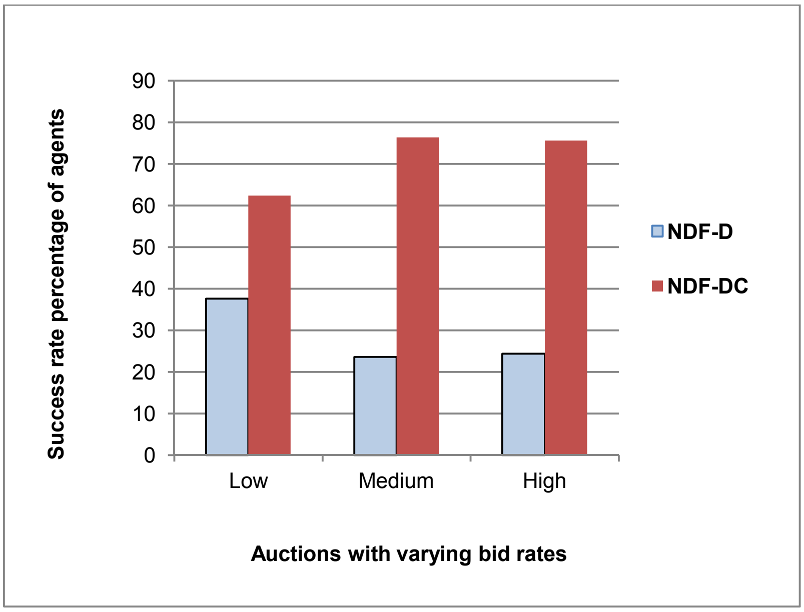


Figure 6.19 Success rate percentage comparison of NDF-based agents with Strategic behaviour

NDF-DC agents excel compared with *NDF-D* agents in terms of $R_{success}$ when we look at heterogeneous agents with *Strategic* behaviour competing across low-, medium- and high- bid-rate auction settings (Figure 6.19). The U_{exp} of *NDF-DCs* is also higher than that of *NDF-Ds* when these agents compete in medium- and high-bid-rate auctions (Figure 6.21 and Figure 6.22). However, in auctions with low bid rates, the U_{exp} of the *NDF-Ds* and that of *NDF-DCs* overlap with one another (Figure 6.20). Bidding agents with *Strategic* behaviour place bids throughout the last hour of the auction, and these bids approach p_r slowly for both *NDF-D* and *NDF-DC* agents. The *NDF-DC* agents' consideration of others' bids hardly affects their own bid increments in these low-bid-rate settings since all the participants' increments are approximately the same as those of *NDF-D*s due to their *Strategic* behaviour.

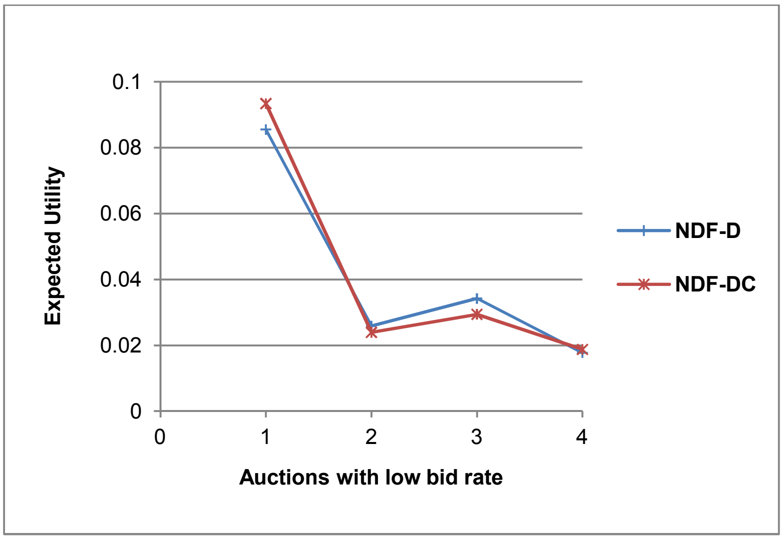


Figure 6.20 Expected utility comparison of NDF-based agents with Strategic behaviour in low-bid-rate auctions

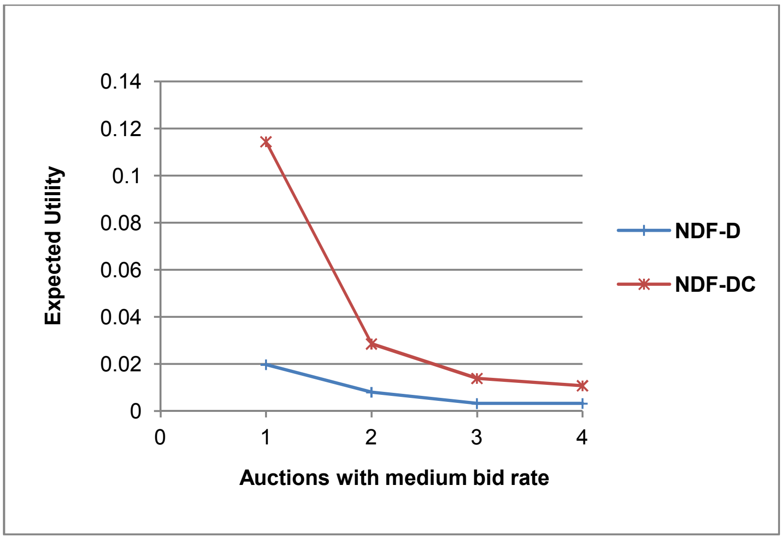


Figure 6.21 Expected utility comparison of NDF-based agents with Strategic behaviour in medium-bid-rate auctions

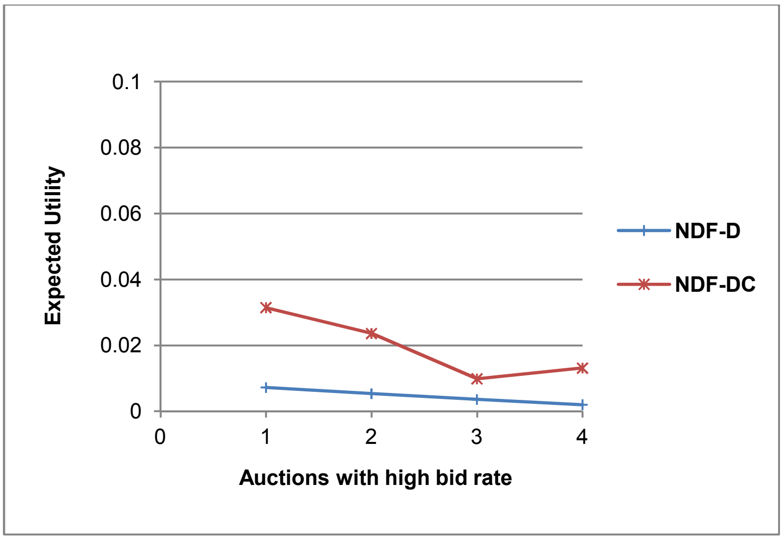


Figure 6.22 Expected utility comparison of NDF-based agents with Strategic behaviour in high-bid-rate auctions

6.2.1.2 Experiments with Homogeneous Bidders

The experiments were carried out separately for each type of the homogeneous bidder agents across various bidding environments, i.e. auctions with different bid rates. The bidding strategies followed by the homogeneous bidders for each value of bargain level value are presented in Table 6.2. The U_{exp} of the bidder agents was measured in two situations: firstly, against the various bid rates of the auctions and secondly, against the different bargain levels of the bidders.

The results of these experiments showed the following: Across auctions of different bid rates, homogeneous *NDF-DCs* with *Sturdy* behaviour and *NDF-DCs* with *Strategic* behaviour always achieve higher U_{exp} than their respective *NDF*-D equivalents (Figure 6.24 and Figure 6.25). In the case of agents with *Mystical* behaviour, *NDF-DCs* outperform *NDF-Ds* when they compete in medium-to high bid-rate auctions; in low-bid-rate auctions, however, these *NDF-Ds* achieve higher U_{exp} than the *NDF-DCs* do (Figure 6.23). This corresponds with an intuition similar to the one about the heterogeneous bidders. In low-bid-rate auctions, *NDF-DCs* with *Mystical* behaviour have limited bid increments. The agents with *Mystical* behaviour place bids in the closing five seconds of an auction at the point when all participants' bids approach p_r. However, the *NDF-DC* agents' consideration of others' bids reduces their bid increments when the low bid rate decreases the amounts that other participants bid.

Table 6.2 Bidding Strategies

Bargain level	k_{MST}	β_{MST}	k_{STD}	β_{STD}	k_{STG}	β_{STG}
10	0.27	0.9	0.27	0.9	0.54	0.9
20	0.24	0.9	0.24	0.9	0.48	0.9
30	0.21	0.6	0.21	0.6	0.42	0.6
40	0.18	0.6	0.18	0.6	0.36	0.6
50	0.15	0.6	0.15	0.6	0.3	0.6
60	0.12	0.3	0.12	0.3	0.24	0.3
70	0.09	0.3	0.09	0.3	0.18	0.3
80	0.06	0.2	0.06	0.2	0.12	0.2
90	0.03	0.2	0.03	0.2	0.06	0.2
100	0	0.2	0	0.2	0	0.2

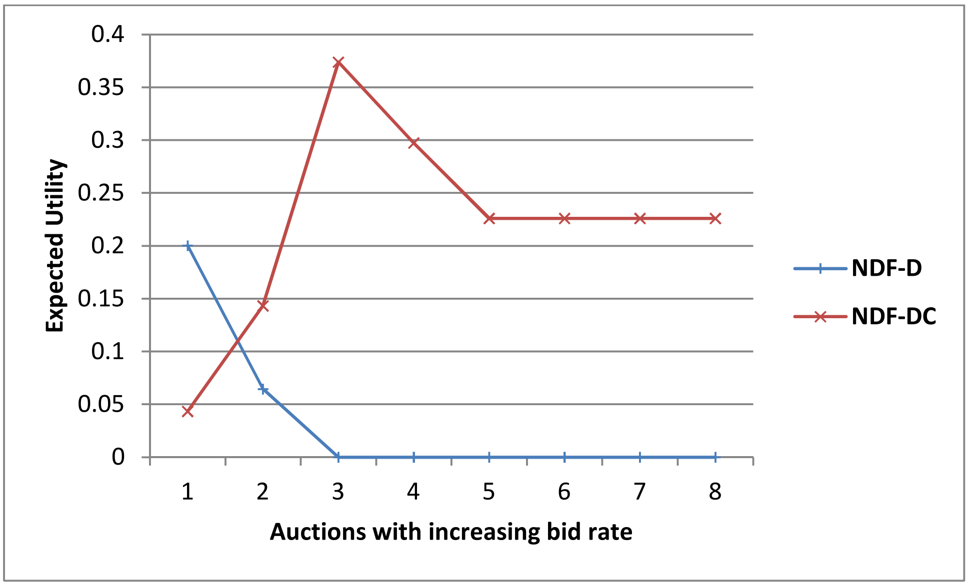


Figure 6.23 Expected utility comparison for homogeneous NDF-based agents with Mystical behaviour with respect to bid rate

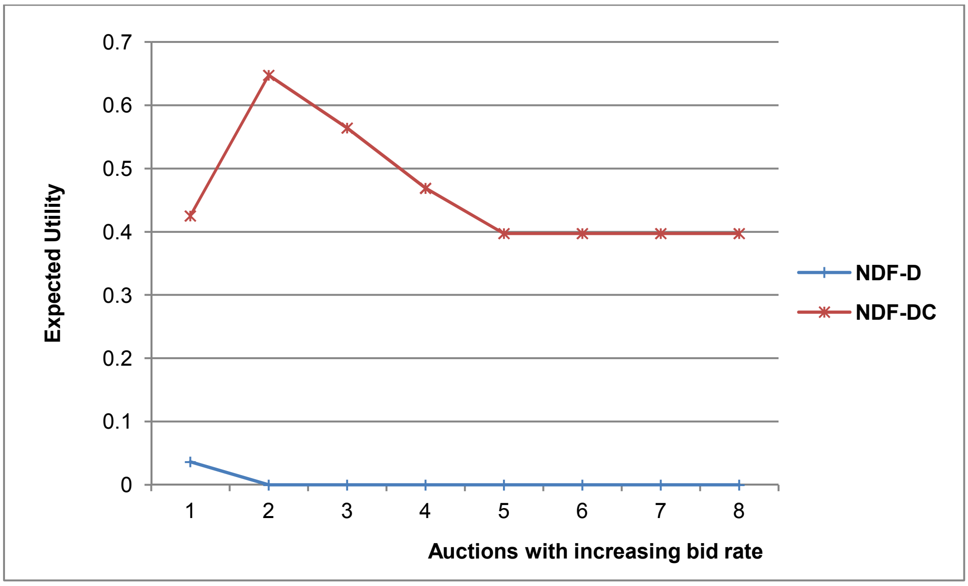


Figure 6.24 Expected utility comparison for homogeneous NDF-based agents with Sturdy behaviour with respect to bid rate

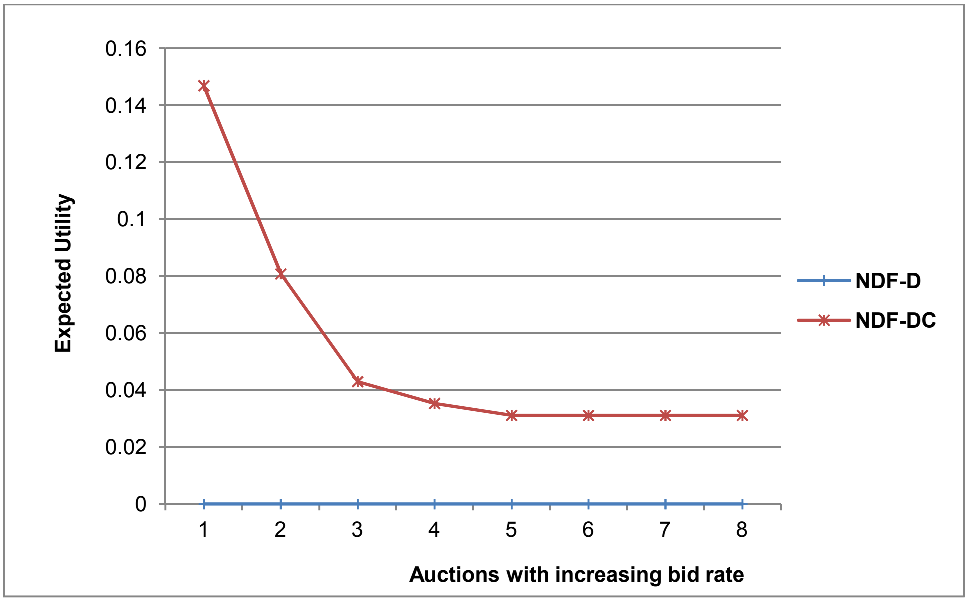


Figure 6.25 Expected utility comparison for homogeneous NDF-based agents with Strategic behaviour with respect to bid rate

The U_{exp} of homogeneous *NDF*-based bidding agents was also measured against their different bargain levels varying from 0 to 100 in steps of 5. Here U_{exp} represents the average of expected utilities of bidding agents in auctions with various bid-rates. The results showed that the *NDF-DCs* with *Mystical*, *Sturdy* and *Strategic* behaviours always achieved higher U_{exp} than the counterpart *NDF-Ds* of each of these types (Figure 6.26, Figure 6.27 and Figure 6.28).

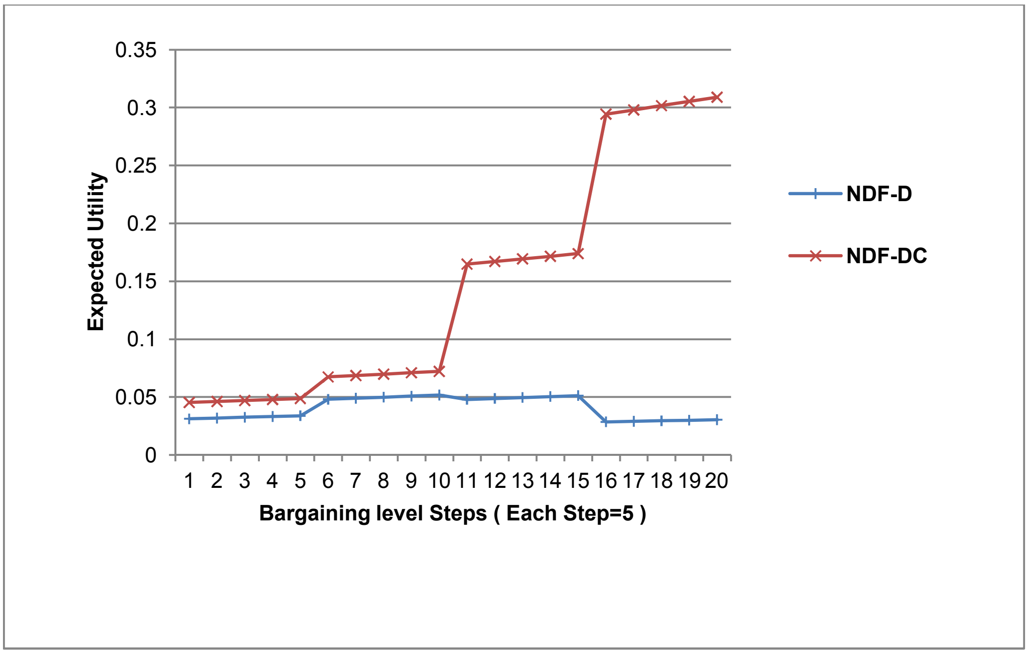


Figure 6.26 Expected utility comparisons for homogeneous NDF-based agents with Mystical behaviour with respect to their bargain level

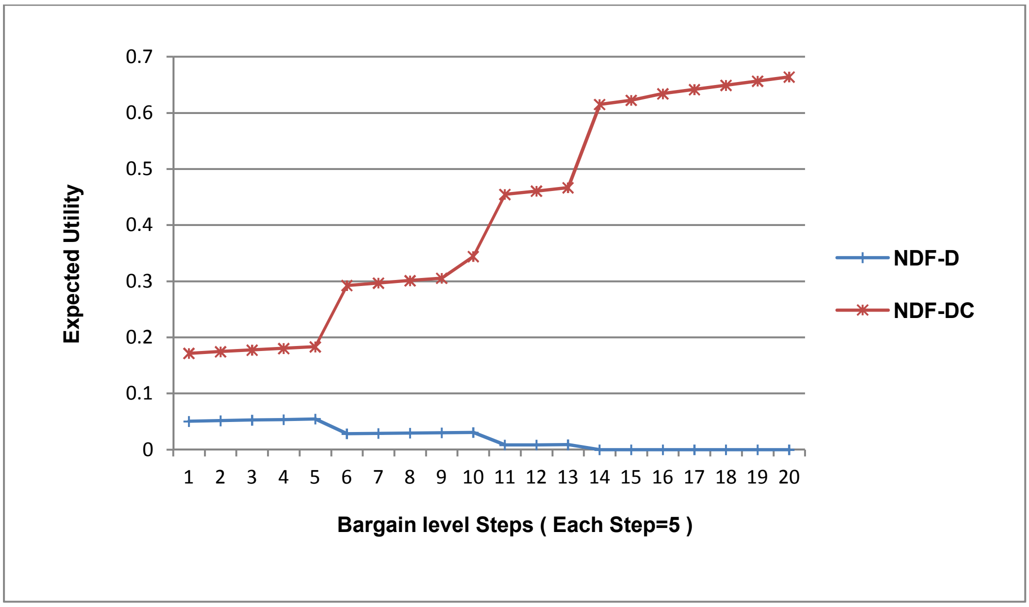


Figure 6.27 Expected utility comparisons for homogeneous NDF-based agents with Sturdy behaviour with respect to their bargain level

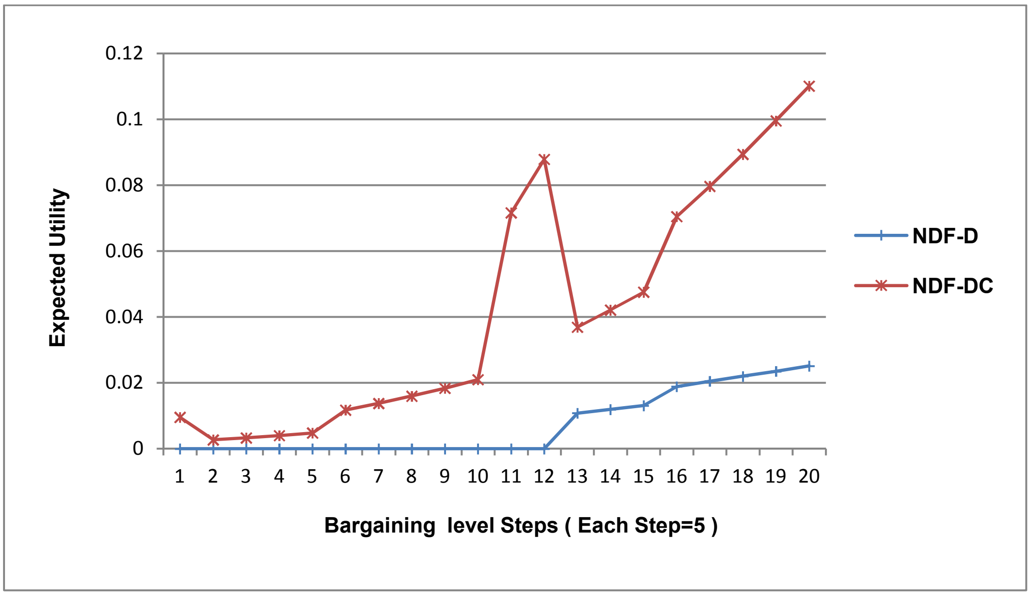


Figure 6.28 Expected utility comparisons for homogeneous NDF-based agents with Strategic behaviour with respect to their bargain level

6.2.2 Evaluating the Fuzzy Reasoning-based Bidding Agents

In the second set of experiments, *Fuzzy* bidding agents were evaluated against *NDF-DC* bidding agents in an auction environment similar to that described in Section 6.2.1 for heterogeneous bidders. For each rule given in Section 5.5.2, fuzzy relations (*R1*, *R2*, *R3* and *R4*) were constructed using Mamdani's method for fuzzy relations and compositional rule of inference (Figure 6.29). The total fuzzy relation R is given in Figure 6.30.

$$\begin{array}{c|ccc|} & \mu C_1(c_1) & \mu C_1(c_2) & \mu C_1(c_3) \\ \hline \mu E_1(e_1) & 1.0 & 0.5 & 0 \\ \mu E_1(e_2) & 0.5 & 0.5 & 0 \\ \mu E_1(e_3) & 0 & 0 & 0 \\ \hline \end{array}$$

$\mu P_1(p_1)$

$$\begin{array}{|ccc|} \mu C_1(c_1) & \mu C_1(c_2) & \mu C_1(c_3) \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline \end{array}$$

$\mu P_1(p_2)$

$$\begin{array}{|ccc|} \mu C_1(c_1) & \mu C_1(c_2) & \mu C_1(c_3) \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline \end{array}$$

$\mu P_1(p_3)$

(a)

$\mu P_2(p_1)$

	$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
$\mu E_1(e_1)$	0	0	0
$\mu E_1(e_2)$	0	0	0
$\mu E_1(e_3)$	0	0	0

$\mu P_2(p_2)$

	$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
$\mu E_1(e_1)$	0	0.5	1. 0
$\mu E_1(e_2)$	0	0.5	0.5
$\mu E_1(e_3)$	0	0	0

$\mu P_2(p_3)$

	$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
$\mu E_1(e_1)$	0	0	0
$\mu E_1(e_2)$	0	0	0
$\mu E_1(e_3)$	0	0	0

(b)

$\mu P_2(p_1)$

	$\mu C_1(c_1)$	$\mu C_1(c_2)$	$\mu C_1(c_3)$
$\mu E_2(e_1)$	0	0	0
$\mu E_2(e_2)$	0	0	0
$\mu E_2(e_3)$	0	0	0

$\mu P_2(p_2)$

	$\mu C_1(c_1)$	$\mu C_1(c_2)$	$\mu C_1(c_3)$
$\mu E_2(e_1)$	0	0	0
$\mu E_2(e_2)$	0.5	0.5	0
$\mu E_2(e_3)$	1.0	0.5	0

$\mu P_2(p_3)$

	$\mu C_1(c_1)$	$\mu C_1(c_2)$	$\mu C_1(c_3)$
$\mu E_2(e_1)$	0	0	0
$\mu E_2(e_2)$	0	0	0
$\mu E_2(e_3)$	0	0	0

(c)

$\mu P_3(p_1)$

	$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
$\mu E_2(e_1)$	1.0	0.5	0
$\mu E_2(e_2)$	0.5	0.5	0
$\mu E_2(e_3)$	0	0	0

$\mu P_3(p_2)$

$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
0	0	0
0	0	0
0	0	0

$\mu P_3(p_3)$

$\mu C_2(c_1)$	$\mu C_2(c_2)$	$\mu C_2(c_3)$
0	0	0
0	0.5	0.5
0	0.5	1.0

(d)

Figure 6.29 Fuzzy relations for the fuzzy rules

1	0.5	0
0.5	0.5	0
0	0	0

0	0.5	1.0
0.5	0.5	0.5
1.0	0.5	0

0	0	0
0	0.5	0.5
0	0.5	1

Figure 6.30 Total fuzzy relation R

The output fuzzy set *ΔP’* on *ΔP* was calculated for different bidding strategies of bidders using input fuzzy sets *E’* on *E* and *C’* on *C* by applying Mamdani’s compositional rule of inference (see Section 5.4) for different levels of competition (low and high). A definite value of the bid increment was calculated by defuzzifying *ΔP'* using a centre of gravity with the weighted mean method. The experiments were carried out for each type of heterogeneous bidder agents separately in different auction environments with various bid rates. $R_{success}$ and U_{exp} of the bidder agents were averaged over auctions with various bid rates for each type of heterogeneous bidder. The results are clear in Figure 6.31 and Figure 6.32. The *NDF-DCs* with *Mystical, Sturdy* and *Strategic* behaviour outperform their *Fuzzy* counterparts of their same behavioural types with respect to $R_{success}$ and U_{exp}.

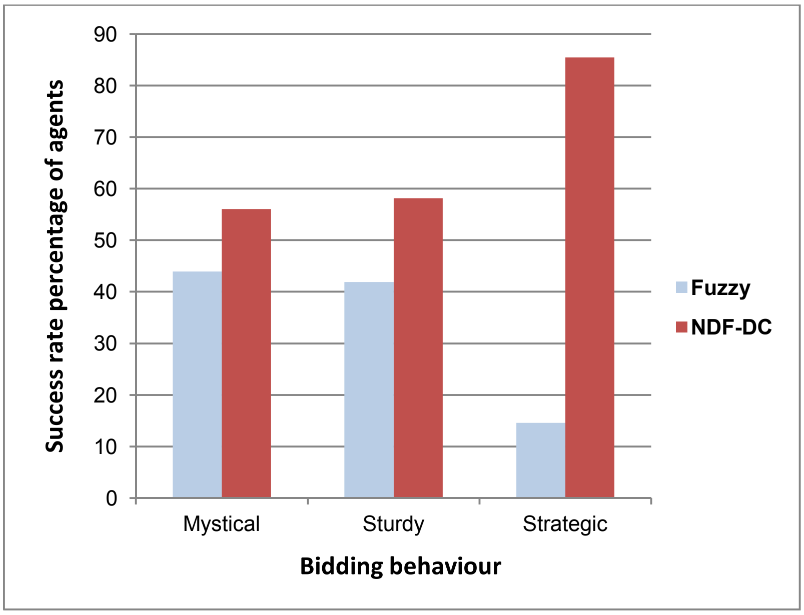


Figure 6.31 Success rate comparisons for Fuzzy and NDF-DC bidding agents

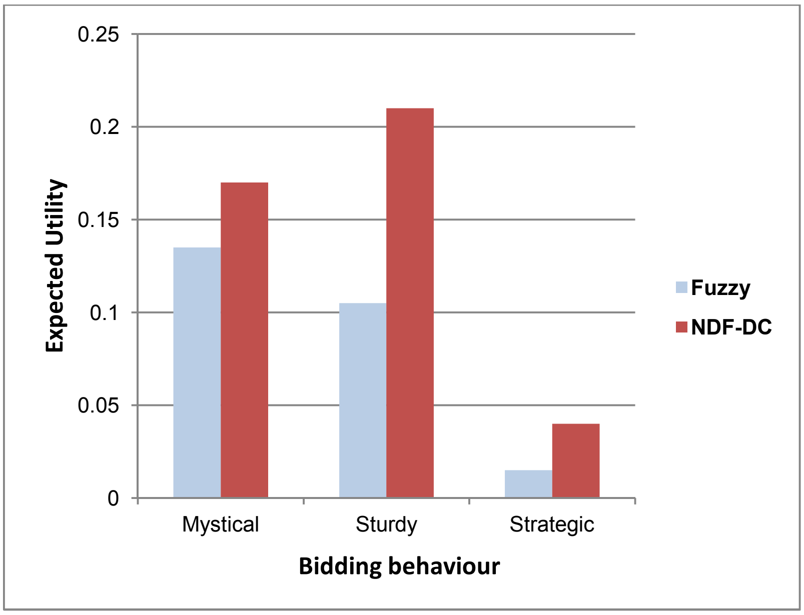


Figure 6.32 Expected utility comparison for Fuzzy and NDF-DC bidding agents

6.2.3 Comparison with Other Bidding Agents

The performance of the *ADBA Bidder* agents was compared against an alternative model agent designed by Anthony & Jennings (2003) whose bidding strategies were based on different bidding constraints. Those constraints were the remaining time left, the remaining auction left, the bidder's desire to get a bargain and his level of desperateness. The constraints were assigned weights and their combination is used to formulate a bid at a particular moment in time. These constraints also followed the negotiation decision functions based on the time dependent negotiation decision functions. This contrasts with the current ADBA framework, which models the bid amount at a particular moment of time based on time - and behavior- dependent negotiation decision functions.

The values of k and β for agent designed by Anthony & Jennings (2003) are set separately for different bidding constraints. For the remaining time left and remaining auction left constraints, $0 \leq k \leq 1$ and $0.005 \leq \beta \leq 1000$. These wide ranges of values of k and β cover a broad spectrum of bidders from a high desire of bargain behaviour to desperate for winning an auction. For desire to get a bargain bidding constraint $0.1 \leq k \leq 0.3$ and $0.005 \leq \beta \leq 0.5$, which corresponds to the fact that an agent with a desire to get a bargain maintains a low bid value until the deadline is almost reached. The values of k and beta are set as $0.7 \leq k \leq 0.9$ and $1.67 \leq \beta \leq 1000$ for the desperate bidding constraint. These values reflect the fact that bidders with desperate behavior starts bidding at a value that is near their reservation price (p_r) and eventually bids at p_r when deadline is reached.

For a genuine comparison, the bidding strategies of these two agents, *ADBA* and the agent developed by Anthony & Jennings (2003), were matched prior to their competition for an item in an ongoing auction. As *ADBA Bidder* agents are designed for bidders who desire to bargain and are not desperate to get an auctioned item, a weight of zero was assigned to the desperateness bidding constraint in the competing model. The same weights were given to all other constraints. The ADBA and alternative model agent have the same level of desire to bargain when they compete in an ongoing auction. Further, both the ADBA and alternative model agent may follow *Mystical*, *Sturdy* or *Strategic* bidding strategies.

Table 6.3 Comparison with other bidding agents

	Success rate		***Expected utility***	
Bidding strategy	***ADBA***	***(Anthony & Jennings 2003)***	***ADBA***	***(Anthony & Jennings 2003)***
Mystical	0.55	0.45	0.133	0.215
Sturdy	0.7	0.3	0.33	0.208
Strategic	0.85	0.15	0.045	0.035

These competing agents with different bidding strategies were assessed separately across a range of auctions with various bid rates. The findings can be observed in Table 6.3: *ADBA Bidder* agents following *Mystical*, *Sturdy* and

Strategic bidding strategies achieve higher $R_{success}$ than each of the equivalent type competing model agents. Further, the U_{exp} of *ADBA Bidder* agents with *Sturdy* and *Strategic* bidding strategies are clearly higher than those of the counterpart agents from the competing model. However, the U_{exp} of the competing model agent exceeds that of *ADBA Bidder* agents when they both follow *Mystical* bidding behaviour. Agents with *Mystical* bidding behaviour place bids in the closing moments of an auction when participants' bids are high and nearly approach their p_r, i.e. the maximum amount they are willing to pay. *ADBA* agents' consideration of the bidding of others, thus, leads them to raise their bid increments. This increases their success rate but at the same time decreases their expected utility.

6.3 Summary

This chapter has described the development of a simulated electronic marketplace in order to implement the ADBA framework and so evaluate the performance its bidding strategies for buyers. To establish this marketplace, a Java Agent DEvelopment framework (JADE) is used which is fully implemented in JAVA language and is designed to develop multi-agent systems that comply with FIPA specifications. The performance of the heterogeneous and homogeneous bidders following the bidding strategies designed in Chapter 5 were then measured separately across wide-ranging test environments subject to different auction settings and bidding restrictions. The results demonstrate that ADBA agents who adopt the bidding approach in this thesis outperform agents following the methodology of Anthony and Jennings (2003) in terms of success and expected utility across most settings.

Chapter 7

Conclusions and Future Study

This chapter draws conclusions about the research presented in this thesis and also suggests some directions for future research.

7.1 Conclusions

The goal of this study was to develop an automated bidding agent that will help bidders to design bidding strategies to win an auction at maximum surplus based on their different bidding behaviours. To achieve this goal, this thesis has proposed an automated dynamic bidding agent (ADBA) framework for the selection of an auction and forecasting of its price at a particular moment in time. These are the main challenges and key processes which a bidding agent must manage to compete successfully in simultaneous online auctions of the same or similar items.

In this study, bidding strategies were designed for use in auction systems like eBay which follow an English format and apply fixed deadlines. Six main research issues have been addressed: (i) how to develop a framework for an automated dynamic bidding agent for bidders participating in simultaneous auctions of the same or similar items; (ii) how to choose the auction where participation will give maximum surplus; (iii) how to assess the value of the item being auctioned and so help bidders to finalise their maximum offer; (iv) how to incorporate the diverse price dynamics of auctions in the auction selection and value assessment processes; (v) how to design bidding strategies for buyers based on their different bidding behaviours; and (vi) how to evaluate the performance of the bidding strategies so designed.

The contributions made in each of these areas are summarised below:

7.1.1 Contribution 1: Automated Dynamic Bidding Agent Framework

An automated dynamic bidding agent (ADBA) for online auctions, introduced in Section 3.2, has been proposed to address the first research question. This ADBA framework is presented in two phases: phase 1 estimates an initial price, using the the price dynamics of diverse auctions to choose an auction to participate in and give a maximum valuation of the auctioned item. Phase 2 designs bidding strategies for buyers with different bidding behaviours by using the output of phase 1 and exploiting negotiation decision functions and fuzzy reasoning techniques.

7.1.2 Contribution 2: Using Diverse Price Dynamics for Price Estimation

A methodology for closing price estimation has been presented in Section 4.1 to address the second, third and fourth research questions related to auction selection and value assessment. Price dynamics—the path taken by bid amounts over the course of an auction—is carefully considered in this study. This is one of the main contributions to decision-making when it comes to estimating an auction's closing price that is used, in turn, to assess the value of the auctioned item. It is unrealistic to assume that the price dynamics of simultaneous auctions for the same or similar items remain the same across any auction environment. This study has therefore characterised auctions of the same or similar items based on their price dynamics before selecting an auction to compete in and assessing the true value of the auctioned item. A bid mapper and selector (BMS) algorithm has been presented which chooses a target auction to compete in. Machine learning techniques are used to estimate the closing price.

This closing price prediction method for value assessment has been validated using a dataset from eBay auctions for a new Palm M515 PDA. Overall, the methodology in this thesis has produced a prediction model superior in accuracy to the closing price prediction methodologies of Raykhel and Ventura (2009) and Van Heijst et al. (2008)

7.1.3 Contribution 3: Designing Bidding Strategies for Different Bidding Behaviours of Buyers

This study has designed bidding strategies for the ADBA agent to address the fifth research question using time- and behaviour-dependent NDFs (Section 5.3), and Mamdani's Method for fuzzy relations and compositional rule of inference (Section 5.4) for buyers with different bidding behaviours. Bidding characteristics such as an auction's attributes, the bidder's own attitude to bidding to get the item in the auction and other bidders' behaviour have been considered in developing these bidding strategies.

7.1.4 Contribution 4: Development of a Simulated Electronic Marketplace

A simulated electronic marketplace, outlined in Section 6.1, has been developed in response to the sixth research question. The simulated marketplace implements the ADBA framework in order to demonstrate how the bidding strategies designed in this thesis perform. The ADBA framework is established using Java Agent DEvelopment Framework environment (JADE), a software platform fully implemented in JAVA language and designed to develop multi-agent systems that comply with FIPA specifications.

The performances of heterogeneous and homogeneous (*NDF-DC*) bidders have been evaluated across a wide variety of test environments subject to different auction settings and different bidding restrictions (bargain levels). They have been compared with *NDF-D* bidders whose bidding strategies are designed using time-dependent negotiation decision functions. The *NDF-DC* bidders always outperform the *NDF-D* bidder, except when heterogeneous bidders with mystical behavior take part in low-bid-rate auctions. This is because participants in these auctions bid lower amounts; *NDF-DC* bidders' consideration of the bids placed by other participants lowers their own bid increments, which reduces their chances of winning the auction.

This study has also assessed the performances of ADBA agents who adopt the different bidding strategies designed in this thesis against the output of agents following a competing model (Anthony & Jennings 2003). The ADBA agents outscore the competing model agents in terms of success rate and expected utility for wide-ranging (*Mystical, Sturdy* and *Strategic*) bidding behaviours in the majority of settings.

7.2 Future Work

Future directions for this research can be summarised into the following tasks:

For its initial price prediction, this study has used a *K-NN* algorithm, which could be improved further by accelerating the process to find the nearest neighbour for a large training dataset. In future, two approaches are planned to speed up the nearest neighbour classification step: first, sophisticated data structures such as search trees will be applied since these take an "almost nearest neighbour" approach to classification, and second, redundant points will be edited out of the training data as these have no effect on the classification and are surrounded by records that belong to the same class.

This study has focused on auctions with hard closing rules in order to design bidding strategies for buyers with different bidding behaviours. It would be interesting to explore the possibility of adapting these *NDF* and *Fuzzy*-based bidding strategies to auctions with soft closing rules and comparing the performance with hard-closing-rules auctions for the same item. This evaluation could be done using a paired design where identical items are auctioned at the same time, with one item in the pair sold off in a hard-closing-rules auction and the other in a soft-closing-rules setting.

The *Fuzzy* bidding agents in this thesis have been designed using Mamdani controller. One research goal is to explore the use of other fuzzy controllers such as a conventional Sugeno controller for this purpose (Sugeno 1985). To accomplish this, it will be essential to determine the suitability of this study's fuzzy-reasoning-based bidding strategies for fuzzy controllers besides the Mamdani controller. A point of interest is the effect of the fixed implication and aggregation methods of the Sugeno controller on *Fuzzy* bidding agent outcomes. In a dynamic auction environment, the fitness of this fuzzy controller—which works well for nonlinear systems by interpolating multiple linear model—will be measured against the intuitive Mamdani controller.

The *NDF* and *Fuzzy* agents evaluated in this thesis were competing to win an individual item in an environment of simultaneous auctions of the same or similar objects. In contrast, combinatorial auctions—where participants can bid on combinations of interdependent items— are gripping the attention of today's smart market. These auctions present computational challenges beyond those of traditional auctions, such as the set packing problem of pair-wise disjoint subsets and the winner determination problem (WDP) (Sandholm 2002). WDP asks how we can efficiently work out the allocation of goods once bids have been submitted to the auctioneers. Extending the negotiation decision functions and fuzzy rule base to deal with these types of problems is one attractive research direction.

Another challenge is to generalise the *NDF* and *Fuzzy* agents so that they can be adapted for other auction types including Dutch, First-Price Sealed Bid and Second-Price Sealed Bid auctions. Each of these auction protocols has its own pricing rules, which complicates the design of buyers' bidding strategies. This will attract more *NDF*-based bidding strategies in the case of *NDF* agents and more fuzzy sets and fuzzy rules in the case of *Fuzzy* agents.

Our *NDF* and *Fuzzy* bidding agents might also be extended fruitfully to implement Continuous Double Auction protocols—an auction format in which buyers submit increasingly higher bids and sellers set increasingly lower asks. In these types of auctions, a transaction occurs when the highest bid is at least the same as the lowest ask. Agents competing in these auctions decide whether to accept/submit a bid/ask based on the current bid/ask. Different strategies need to be designed for sellers and buyers to reflect their opposing behaviour when choosing the value of asks and bids.

These are only some of the rich areas of future study to which the work in this thesis points.

Bibliography

Adomavicius, G., Gupta, A. & Zhdanov, D. 2009, 'Designing intelligent software agents for auctions with limited information feedback', *Information Systems Research*, vol. 20, no. 4, pp. 507-26.

Anthony, P., Hall, W., Dang, V. & Jennings, N. 2001, 'Autonomous agents for participating in multiple online auctions', in *Proc IJCAI Workshop on E-Business and Intelligent Web*, Citeseer, Seattle, USA, pp. 54-64.

Anthony, P. & Jennings, N. 2002, 'Evolving bidding strategies for multiple auctions', in *Proceedings of 15th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, pp. 178-82.

Anthony, P. & Jennings, N.R. 2003, 'Developing a bidding agent for multiple heterogeneous auctions', *ACM Transactions on Internet Technology (TOIT)*, vol. 3, no. 3, pp. 185-217.

Ariely, D. & Simonson, I. 2003, 'Buying, bidding, playing, or competing? Value assessment and decision dynamics in online auctions', *Journal of Consumer Psychology*, vol. 13, no. 1, pp. 113-23.

Ashri, R., Rahwan, I. & Luck, M. 2003, 'Architectures for negotiating agents', in V. Marik ,M. Pechoucek, J. Muller (eds.) *Multi-Agent Systems and Applications III*, Springer, Berlin Heidelberg, , pp. 136-46.

Ba, S. & Pavlou, P.A. 2002, 'Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior', *MIS Quarterly*, Vol. 26, no. 3, pp. 243-68.

Bajari, P. & Hortacsu, A. 2003a, 'Cyberspace auctions and pricing issues: A review of empirical findings', Derek C. Jones, *New Economy Handbook*, Academic Press, San Diego, CA.

Bajari, P. & Hortacsu, A. 2003b, 'The winner's curse, reserve prices, and endogenous entry: empirical insights from eBay auctions', *RAND Journal of Economics*, vol. 34, no. 2, pp. 329-55.

Bapna, R., Goes, P. & Gupta, A. 2000, 'A theoretical and empirical investigation of multi item on line auctions', *Information Technology and Management*, vol. 1, no. 1, pp. 1-23.

Bapna, R., Goes, P. & Gupta, A. 2003, 'Analysis and design of business-to-consumer online auctions', *Management Science*, vol. 49, no. 1, pp. 85-101.

Bapna, R., Goes, P., Gupta, A. & Jin, Y. 2004, 'User heterogeneity and its impact on electronic auction market design: An empirical exploration', *Mis Quarterly*, vol. 28, no. 1, pp. 21-43.

Bellifemine, F., Bergenti, F., Caire, G. & Poggi, A. 2005, 'JADE—a java agent development framework', in R.H. Bordini, M. Dastani, J. Dix & A.E.F. Seghrouchni (eds), *Multi-Agent Programming*, Springer, New York, NY, USA, pp. 125-47.

Beltratti, A., Margarita, S. & Terna, P. 1996, *Neural networks for economic and financial modelling*, International Thomson Computer Press, London, UK.

Bertsimas, D., Hawkins, J. & Perakis, G. 2009, 'Optimal bidding in online auctions', *Journal of Revenue & Pricing Management*, vol. 8, no. 1, pp. 21-41.

Borgs, C., Chayes, J., Immorlica, N., Jain, K., Etesami, O. & Mahdian, M. 2007, 'Dynamics of bid optimization in online advertisement auctions', in *Proceedings of the 16th international conference on World Wide Web*, ACM, New York, NY, USA, pp. 531-40.

Borle, S., Boatwright, P. & Kadane, J.B. 2006, 'The timing of bid placement and extent of multiple bidding: An empirical investigation using eBay online auctions', *Statistical Science*, vol. 21, no. 2, pp. 194-205.

Bouguessa, M., Wang, S. & Sun, H. 2006, 'An objective approach to cluster validation', *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1419-30.

Brannman, L., Klein, J. & Weiss, L. 1987, 'The price effects of increased competition in auction markets', *The Review of Economics and Statistics*, vol. 69, no. 1, pp. 24-32.

Braubach, L. & Pokahr, A. 2009, 'Jadex: BDI Agent System', *Distributed Systems and Information Systems Group, University of Hamburg.*

Brooks, R.A. 1991, 'Intelligence without representation', *Artificial Intelligence*, vol. 47, no. 1, pp. 139-59.

Brzostowski, J. & Kowalczyk, R. 2005, 'On possibilistic case-based reasoning for selecting partners in multi-agent negotiation', G. Webb & X. Yu (eds.) AI

2004: Advances in Artificial Intelligence, Springer, Berlin Heidelberg, pp. 694-705.

Brzostowski, J. & Kowalczyk, R. 2012, 'Efficient algorithm for estimation of qualitative expected utility in possibilistic case-based reasoning', *21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, AUAI Press, Edinburgh, Scotland, pp. 69-76.

Byde, A. 2001, *A dynamic programming model for algorithm design in simultaneous auctions*, Springer, Berlin Heidelberg.

Byde, A. 2002, *A comparison among bidding algorithms for multiple auctions*, Springer, Berlin Heidelberg.

Byde, A., Preist, C. & Jennings, N. 2002, 'Decision procedures for multiple auctions', *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, ACM, New York, pp. 613-20.

Cao, Q., Leggio, K.B. & Schniederjans, M.J. 2005, 'A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market', *Computers & Operations Research*, vol. 32, no. 10, pp. 2499-512.

Carbo, J., Molina, J.M. & Davila, J. 2001, 'A BDI agent architecture for reasoning about reputation', *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, IEEE, pp. 817-22.

Chakraborty, S., Sen, A.K. & Bagchi, A. 2009, 'Designing proxy bidders for online combinatorial auctions', *42nd Hawaii International Conference on System Sciences*, IEEE, Washington, DC, USA, pp. 1-10.

Chalmers, S., Gray, P.M. & Preece, A. 2002, 'Supporting virtual organisations using BDI agents and constraints', in Matthias Klusch, Sascha Ossowski & Onn Shehory (ed). *Cooperative Information Agents VI*, Springer, Berlin Heidelberg, pp. 226-40.

Chan, C.C.H. 2008, 'Intelligent spider for information retrieval to support mining-based price prediction for online auctioning', *Expert Systems with Applications*, vol. 34, no. 1, pp. 347-56.

Chang, C.-h. & Chen, Y. 1996, 'Autonomous intelligent agent and its potential applications', *Computers & Industrial Engineering*, vol. 31, no. 1-2, pp. 409-12.

Dass, M., Jank, W. & Shmueli, G. 2010, 'Dynamic price forecasting in simultaneous online art auctions', in J. Casillas & F.J. Martínez-López (eds), *Marketing Intelligent Systems using Soft Computing: Managerial and Research Applications*, Springer, Berlin Heidelberg, pp. 417-45.

Dass, M., Jank, W. & Shmueli, G. 2011, 'Maximizing bidder surplus in simultaneous online art auctions via dynamic forecasting', *International Journal of Forecasting*, vol. 27, no. 4, pp. 1259-70.

Dayanik, S. & Parlar, M. 2013, 'Dynamic bidding strategies in search-based advertising', *Annals of Operations Research*, vol. 211, no. 1, pp. 103-36.

Dimou, C., Symeonidis, A. L., Mitkas, P.A. 2007, ' Data Mining and Agent Technology: a fruitful symbiosis', in Oded Maimon, Lior Rokach (eds), *Soft Computing for Knowledge Discovery and Data Mining* , Springer, New York, NY, USA, pp. 327-62.

Du, L., Chen, Q. & Bian, N. 2010, 'An empirical analysis of bidding behavior in simultaneous ascending-bid auctions', *International Conference on E-Business and E-Government (ICEE),* IEEE, pp. 249-51.

Dumas, M., Aldred, L., Governatori, G. & ter Hofstede, A.M. 2005, 'Probabilistic Automated Bidding in Multiple Auctions', *Electronic Commerce Research*, vol. 5, no. 1, pp. 25-49.

Faratin, P., Sierra, C. & Jennings, N.R. 1998, 'Negotiation decision functions for autonomous agents', *Robotics and Autonomous Systems*, vol. 24, no. 3, pp. 159-82.

Fasli, M. 2007, *Agent Technology for e-commerce*, John Wiley & Sons, Chichester.

Ford, B.J., Xu, H., Bates, C.K. & Shatz, S.M. 2010, 'Visual Specification of Layered Bidding Strategies for Autonomous Bidding Agents', *Journal of Computers*, vol. 5, no. 6, pp. 940-50.

Fraley, C. & Raftery, A.E. 1998, 'How many clusters? Which clustering method? Answers via model-based cluster analysis', *The Computer Journal*, vol. 41, no. 8, pp. 578-88.

Fred, A.L. & Jain, A.K. 2002, 'Data clustering using evidence accumulation', *16th International Conference on Pattern Recognition*, vol. 4, IEEE, pp. 276-80.

Frontline Systems Inc 2013, *Analytic Solver Platform XLMiner Data Mining User Guide*, viewed 20 January 2013, <http://www.solver.com>.

Fruchter, G.E. & Dou, W. 2005, 'Optimal budget allocation over time for keyword ads in web portals', *Journal of Optimization Theory and Applications*, vol. 124, no. 1, pp. 157-74.

Gan Kim, S., Anthony, P., Teo, J. & Chin Kim, O. 2009, 'Evolving Bidding Strategies Using Self-Adaptation Genetic Algorithm', *International Symposium on Intelligent Ubiquitous Computing and Education,* IEEE, pp. 222-5.

Georgeff, M.P. & Ingrand, F.F. 1989, 'Decision-making in an embedded reasoning system', *11th international joint conference on Artificial intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 972-978.

Ghani, R. & Simmons, H. 2004, 'Predicting the end-price of online auctions', in *Proceedings of the International Workshop on Data Mining and Adaptive Modelling Methods for Economics and Management*, Pisa, Italy.

Goyal, M., Kaushik, S. & Kaur, P. 2010, 'Automated Fuzzy Bidding Strategy for Continuous Double Auctions Using Trading Agent's Attitude and Market Competition', *International Journal of Agent Technologies and Systems*, vol. 2, no. 4, pp. 56-74.

Goyal, M., Lu, J. & Zhang, G. 2005, 'Negotiating Multi-Issue e-Market Auction through Fuzzy Attitudes', *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 2, IEEE, pp. 922-7.

Goyal, M., Lu, J. & Zhang, G. 2008, 'Decisión Making in Multi-lssue e-Market Auction Using Fuzzy Techniques and Negotiable Attitudes', *Journal of Theoretical and applied Electronic Commerce Research*, vol. 3, no. 2, pp. 97-110.

Greenwald, A. & Boyan, J. 2004, 'Bidding under uncertainty: Theory and experiments', in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, AUAI Press, pp. 209-16.

Greenwald, A. & Stone, P. 2001, 'Autonomous bidding agents in the trading agent competition', *IEEE Internet Computing*, vol. 5, no. 2, IEE, pp. 52-60.

Grossman, J. 2013, *Predicting eBay Auction Sales with Machine Learning*, accessed 20 September 2013 <http://jaygrossman.com/post/2013/06/10/Predicting-eBay-Auction-Sales-with-Machine-Learning.aspx>.

Grudnitski, G. & Osburn, L. 1993, 'Forecasting S&P and gold futures prices: an application of neural networks', *Journal of Futures Markets*, vol. 13, no. 6, pp. 631-43.

Halkidi, M., Batistakis, Y. & Vazirgiannis, M. 2001, 'On clustering validation techniques', *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107-45.

Han, J. & Kamber, M. 2006, *Data mining: concepts and techniques*, Morgan Kaufmann.

Haruvy, E. & Popkowski Leszczyc, P.T.L. 2010, 'Internet Auctions', *Foundations and Trends® in Marketing*, vol. 4, no. 1, pp. 1-75.

Haubl, G. & Popkowski-Leszczyc, P. 2000, 'Going, Going, Gone: Determinants of Bidding Behavior and Selling Prices in Internet Auctions', paper presented to the *Division 23, APA conference*, San Francisco, CA.

He, M. & Jennings, N.R. 2004, 'Designing a successful trading agent: A fuzzy set approach', *Fuzzy Systems, IEEE Transactions on Fuzzy Systems*, vol. 12, no. 3, pp. 389-410.

He, M., Jennings, N.R. & Prügel-Bennett, A. 2006, 'A heuristic bidding strategy for buying multiple goods in multiple English auctions', *ACM Transactions on Internet Technology (TOIT)*, vol. 6, no. 4, pp. 465-96.

He, M. & Leung, H.-f. 2001, 'An agent bidding strategy based on fuzzy logic in a continuous double auction', *IEEE International Conference on Systems, Man, and Cybernetics* vol. 1, IEEE, pp. 583-8.

He, M., Leung, H.-f. & Jennings, N.R. 2003, 'A fuzzy-logic based bidding strategy for autonomous agents in continuous double auctions', *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1345-63.

Hirose, H., Soejima, Y. & Hirose, K. 2012, 'NNRMLR: A combined method of nearest neighbor regression and multiple linear regression', *IIAI International Conference on Advanced Applied Informatics (IIAIAAI)* IEEE, pp. 351-6.

Hou, C. 2004, *Modelling agents behaviour in automated negotiation*, Technical Report KMI-TR-144 Knowledge Media Institute, The Open University, Milton Keynes, UK.

Houser, D. & Wooders, J. 2006, 'Reputation in auctions: Theory and evidence from eBay', *Journal of Economics & Management Strategy*, vol. 15, no. 2, pp. 353-69.

Hu, J. & Guan, C. 2005, 'Model of game agent for auction-based negotiation', in Y. Hao, J. Liu, Y. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma & Y.-C. Jiao (eds), *Computational Intelligence and Security*, vol. 3801, Springer, Berlin Heidelberg, pp. 387-92.

Indrawan, M. 2001, 'Software Agents in Electronic Commerce: An Overview', *Internet Commerce and Software Agents: Cases, Technologies, and Opportunities*, Idea Group Publishing, USA, pp. 75-87.

Jain, P. & Dahiya, D. 2012, 'An Intelligent Multi Agent Framework for E-commerce Using Case Based Reasoning and Argumentation for Negotiation', *Information Systems, Technology and Management*, Springer, Berlin Heidelberg, pp. 164-75.

Jank, W. & Shmueli, G. 2005, *Profiling price dynamics in online auctions using curve clustering*, Working paper, Smith School of Business, University of Maryland.

Jank, W. & Shmueli, G. 2009, 'Studying heterogeneity of price evolution in eBay auctions via functional clustering', in G. Adomavicius & A. Gupta (eds), *Business Computing*, vol. 3, Emerald, pp. 237-261.

Jank, W. & Shmueli, G. 2010, *Modeling Online Auctions*, vol. 91, electronic book, John Wiley & Sons, Hoboken New Jersey.

Jank, W. & Zhang, S. 2011, 'An automated and data-driven bidding strategy for online auctions', *INFORMS Journal on Computing*, vol. 23, no. 2, pp. 238-53.

Jenamani, M., Routray, A. & Singh, V. 2011, 'A procedure using support vector data description and mutual information for end price assessment in online C2C auction', *Electronic Commerce Research*, vol. 11, no. 3, pp. 321-40.

Jennings, N. & Wooldridge, M. 1996, 'Software agents', *IEE Review*, vol. 42, no. 1, pp. 17-20.

Jiang, A.X. & Leyton-Brown, K. 2007, 'Bidding agents for online auctions with hidden bids', *Machine Learning*, vol. 67, no. 1-2, pp. 117-43.

Johns, C. & Zaichkowsky, J. 2003, 'Bidding behavior at the auction', *Psychology and Marketing*, vol. 20, no. 4, pp. 303-22.

Jong Yih, K. & Lee, J. 2004, 'Evolution of intelligent agent in auction market', *Proceedings of IEEE International Conference on Fuzzy Systems*, vol. 1, IEEE, pp. 331-336.

Ju-Long, D. 1982, 'Control problems of grey systems', *Systems & Control Letters*, vol. 1, no. 5, pp. 288-94.

Kaushik, S. & Tagra, H. 2005, 'CBR Based learning in e-auctions', *Journal of Theoretical & Applied Information Technology*, vol. 3, no. 1, pp. 18-29.

Kehagias, D. & Mitkas, P.A. 2005, 'Adaptive pricing functions for open outcry auctions', *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IEEE, pp. 653-6.

Kehagias, D., Symeonidis, A. & Mitkas, P. 2005, 'Designing pricing mechanisms for autonomous agents based on bid-forecasting', *Electronic Markets*, vol. 15, no. 1, pp. 53-62.

Kehagias, D.D. & Mitkas, P.A. 2007, 'Efficient E-Commerce Agent Design Based on Clustering eBay Data', *International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM,* pp. 495-8.

Kim, H. & Baek, S. 2004, 'Forecasting Winning Bid Prices in an Online Auction Market', *Journal of Electronic Science and Technology of China*, vol. 2, no. 3, pp. 213-9.

Kimoto, T., Asakawa, K., Yoda, M. & Takeoka, M. 1990, 'Stock market prediction system with modular neural networks', *International Joint Conference on Neural Networks, IJCNN* IEEE, pp. 1-6.

Kohzadi, N., Boyd, M.S., Kermanshahi, B. & Kaastra, I. 1996, 'A comparison of artificial neural network and time series models for forecasting commodity prices', *Neurocomputing*, vol. 10, no. 2, pp. 169-81.

Kolodner, J.L. 1992, 'An introduction to case-based reasoning', *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3-34.

Ku, G. 2000, 'Auctions and auction fever: Explanations from competitive arousal and framing', *Kellogg Journal of Organization Behavior*, vol. 2000, pp. 1-41.

Ku, G., Malhotra, D. & Murnighan, J.K. 2005, 'Towards a competitive arousal model of decision-making: A study of auction fever in live and Internet auctions', *Organizational Behavior and Human Decision Processes*, vol. 96, no. 2, pp. 89-103.

Lim, D., Anthony, P. & Ho, C. 2008, 'Agent for Predicting Online Auction Closing Price in a Simulated Auction Environment', in T.-B. Ho & Z.-H. Zhou (eds), *PRICAI 2008: Trends in Artificial Intelligence*, vol. 5351, Springer Berlin Heidelberg, pp. 223-34.

Lim, D., Anthony, P. & Ho, C.M. 2011, 'The Performance of Grey System Agent and ANN Agent in Predicting Closing Prices for Online Auctions', *International Journal of Agent Technologies and Systems (IJATS)*, vol. 3, no. 4, pp. 37-56.

Lim, D., Anthony, P. & Mun, H.C. 2007, 'Evaluating the accuracy of grey system theory against time series in predicting online auction closing price', *IEEE International Conference on Grey Systems and Intelligent Services, 2007*, IEEE, pp. 463-70.

Lin, C.-S., Chou, S., Weng, S.-M. & Hsieh, Y.-C. 2013, 'A final price prediction model for english auctions: a neuro-fuzzy approach', *Quality & Quantity*, vol. 47, no. 2, pp. 599-613.

Lin, X., Yang, Z. & Song, Y. 2009, 'Short-term stock price prediction based on echo state networks', *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 7313-7.

Liu, S. & Lin, Y. 2005, *Grey Information: Theory and Practical Applications*, Springer-Verlag New York, Inc.

Lucking Reiley, D., Bryan, D., Prasad, N. & Reeves, D. 2007, 'Pennies from eBAy: The Determinants of Price in Online Auctions', *The Journal of Industrial Economics*, vol. 55, no. 2, pp. 223-33.

Ma, H. & Leung, H.-F. 2008, 'An adaptive attitude bidding strategy for agents in continuous double auctions', *Electronic Commerce Research and Applications*, vol. 6, no. 4, pp. 383-98.

Ma, J. & Goyal, M.L. 2010, 'Designing a Successful Bidding Strategy Using Fuzzy Sets and Agent Attitudes', in JingTao Yao (ed), *Web-based Support Systems*, Springer, pp. 359-70.

Maamar, Z. 2002, 'Association of users with software agents in e-commerce', *Electronic Commerce Research and Applications*, vol. 1, no. 1, pp. 104-12.

Mackie-Mason, J., Osepayshvili, A., Reeves, D. & Wellman, M. 2004, 'Price prediction strategies for market-based scheduling', paper presented to the *Fourteenth International Conference on Automated Planning and Scheduling*, Whistler (British Columbia), Canada, 3-7 June.

Milligan, G.W. & Cooper, M.C. 1985, 'An examination of procedures for determining the number of clusters in a data set', *Psychometrika*, vol. 50, no. 2, pp. 159-79.

Min, Z. & Qiwan, W. 2009, 'The On-line Electronic Commerce Forecast Based on Least Square Support Vector Machine', *Second International Conference on Information and Computing Science, 2009. ICIC '09* , vol. 2, pp. 75-8.

Minghua, H., Jennings, N.R. & Prugel-Bennett, A. 2004, 'An adaptive bidding agent for multiple English auctions: a neuro-fuzzy approach', *IEEE International Conference on Fuzzy Systems,* vol. 3, IEEE, pp. 1519-24.

Montani, S. & Jain, L. 2010, 'Innovations in Case-Based Reasoning Applications', in S. Montani & L. Jain (eds), *Successful Case-based Reasoning Applications - I*, vol. 305, Springer Berlin Heidelberg, pp. 1-5.

Mooi, E.A. & Sarstedt, M. 2011, *A Concise Guide to Market Research: The Process, Data, and Methods (Using IBM SPSS Statistics)*, Springer, Heidelberg.

Naser, A.M.A. 2012, 'Genetic Algorithm Strategies and Tactics Model for Agent Negotiation in E-Commerce Systems', *Basic Science and Engineering*, vol. 7, no. 1, pp. 51-8.

Nikolaidou, V. & Mitkas, P.A. 2009, 'A sequence mining method to predict the bidding strategy of trading agents', *Agents and Data Mining Interaction*, Springer, Berlin Heidelberg, pp. 139-51.

Ockenfels, A., Reiley, D.H. & Sadrieh, A. 2006, 'Online auctions', in T. Hendershott (ed.), *Economics and Information Systems Handbook*, Elsevier Science, Amsterdam, pp. 571-622.

Ockenfels, A. & Roth, A.E. 2002, 'The timing of bids in internet auctions: Market design, bidder behavior, and artificial agents', *AI Magazine*, vol. 23, no. 3, p. 79.

Ockenfels, A. & Roth, A.E. 2006, 'Late and multiple bidding in second price Internet auctions: Theory and evidence concerning different rules for ending an auction', *Games and Economic Behavior*, vol. 55, no. 2, pp. 297-320.

Pai, P.-F. & Lin, C.-S. 2005, 'A hybrid ARIMA and support vector machines model in stock price forecasting', *Omega*, vol. 33, no. 6, pp. 497-505.

Park, Y.H. & Bradlow, E.T. 2005, 'An integrated model for bidding behavior in Internet auctions: Whether, who, when, and how much', *Journal of Marketing Research*, vol. 42, no. 4, pp. 470-82.

Pavanje, N. 2012, 'Bidding Strategy in Simultaneous English Auctions Using Game Theory', in *Proceedings of International Conference on Advances in Computing*, Springer, pp. 763-71.

Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T. & Provost, F. 2012, 'Bid optimizing and inventory scoring in targeted online advertising', in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 804-12.

Pinyol, I., Centeno, R., Hermoso, R., da Silva, V.T. & Sabater-Mir, J. 2010, 'Norms Evaluation through Reputation Mechanisms for BDI Agents', in *Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*, pp. 9-18.

Pinyol, I., Sabater-Mir, J. & Dellunde, P. 2008, 'Cognitive social evaluations for multi-context bdi agents', *Eleventh International Conference of the Catalan Association for Artificial Intelligence CCIA'08*, 22-24 October 2008, St. Martin Empúries.

Preist, C., Bartolini, C. & Phillips, I. 2001, *Algorithm design for agents which participate in multiple simultaneous auctions*, Springer.

Rao, A.S. & Georgeff, M.P. 1991, 'Modeling rational agents within a BDI-architecture', *KR*, vol. 91, pp. 473-84.

Rasmusen, E.B. 2006, 'Strategic implications of uncertainty over one's own private value in auctions', *BE Press Journal*, vol. 6, no. 1, p. Article 7.

Raykhel, I. & Ventura, D. 2008, *Real-time Automatic Price Prediction for eBay Online Trading*, Brigham Young University. Dept. of Computer Science.

Raykhel, I. & Ventura, D. 2009, 'Real-time Automatic Price Prediction for eBay online trading', in *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, pp. 135-40.

Resnick, P., Zeckhauser, R., Swanson, J. & Lockwood, K. 2006, 'The value of reputation on eBay: A controlled experiment', *Experimental Economics*, vol. 9, no. 2, pp. 79-101.

Rivera-Borroto, O.M., Rabassa-Gutiérrez, M., Grau-Abalo, R.d.C., Marrero-Ponce, Y. & García-de la Vega, J.M. 2012, 'Dunn's index for cluster tendency assessment of pharmacological data sets', *Canadian Journal of Physiology and Pharmacology*, vol. 90, no. 4, pp. 425-33.

Roth, A.E. & Ockenfels, A. 2002, 'Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet', *The American Economic Review*, vol. 92, no. 4, pp. 1093-103.

Rousseeuw, P.J. 1987, 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65.

Sandholm, T. 2002, 'Algorithm for optimal winner determination in combinatorial auctions', *Artificial Intelligence*, vol. 135, no. 1, pp. 1-54.

Schindler, J. 2003, 'Late bidding on the Internet', *unpublished paper*.

Schindler, R. & Schindler, R.M. 2011, *Pricing Strategies: A Marketing Approach*, SAGE Publications.

Schmid, E. & Paulussen, T.O. 2005, *Decision-theoretic bidding in online-auctions*, Universität Mannheim/Fakultät für Betriebswirtschaftslehre.

Shah, H., Joshi, N., Sureka, A. & Wurman, P. 2003, 'Mining eBay: Bidding strategies and shill detection', in O.R. Zaïane, J. Srivastava, M. Spiliopoulou & B. Masand (eds), *WEBKDD 2002-MiningWeb Data for Discovering Usage Patterns and Profiles*, vol. 2703, Springer Berlin Heidelberg, pp. 17-34.

Shang, R., Chen, Y. & Cheng, M. 2006, 'An Empirical Classification of Bidders in Online Auctions', paper presented to the *The Tenth Pacific Asia Conference on Information Systems*, Kuala Lumpur, Malaysia.

Shmueli, G., Patel, N.R. & Bruce, P.C. 2007, *Data mining in excel: Lecture notes and cases*, Arlington, VA: Resampling Stats Inc.

Sidnal, N. & Manvi, S. 2012, 'Automated Bidding in English Auctions Using Cognitive Agents in Mobile e-Commerce', *International Journal of Information and Education Technology* vol. 2, no. 2, pp. 128-34.

Sow, J., Anthony, P. & Chong Mun, H. 2010, 'Competition among intelligent agents and standard bidders with different risk behaviors in simulated English auction marketplace', *IEEE International Conference on Systems Man and Cybernetics (SMC)*, IEEE, pp. 1022-8.

Sow, J., Anthony, P. & Ho, C.M. 2012, 'Homogeneous and Heterogeneous Agents in Electronic Auctions', in H. Xu (ed.), *Practical Applications of Agent-Based Technology*, InTech, pp. 45-70.

Still, S. & Bialek, W. 2004, 'How many clusters? an information-theoretic perspective', *Neural Computation*, vol. 16, no. 12, pp. 2483-506.

Stone, P., Schapire, R.E., Csirik, J.A., Littman, M.L. & McAllester, D. 2002, 'ATTac-2001: A learning, autonomous bidding agent', in J,Padget et al. (eds.) *Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems*, Springer, pp. 143-60.

Sugeno, M. 1985, 'An introductory survey of fuzzy control', *Information Sciences*, vol. 36, no. 1, pp. 59-83.

Sun, E. 2005, *The effects of auctions parameters on price dispersion and bidder entry on eBay: a conditional logit analysis*, working paper, Stanford University.

Symeonidis, A.L. & Mitkas, P.A. 2005, *Agent intelligence through data mining*, vol. 14, Springer Verlag.

Tanaka, K. & Niimura, T. 1997, *An introduction to fuzzy logic for practical applications*, Springer New York.

Tesauro, G. & Bredin, J.L. 2002, 'Strategic sequential bidding in auctions using dynamic programming', in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, ACM, pp. 591-8.

Trevathan, J. & Read, W. 2009, 'Detecting shill bidding in online English auctions', *Handbook of Research on Social and Organizational Liabilities* *in Information Security*, Information Science Reference, PA, USA, pp. 446-70.

Tsang, P.M., Kwok, P., Choy, S.O., Kwan, R., Ng, S.C., Mak, J., Tsang, J., Koong, K. & Wong, T.-L. 2007, 'Design and implementation of NN5 for Hong Kong stock price forecasting', *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 453-61.

Ullah, S. & Finch, C.F. 2013, 'Applications of functional data analysis: A systematic review', *BMC Medical Research Methodology*, vol. 13, no. 1, p. 43.

Van Heijst, D., Potharst, R. & Van Wezel, M. 2008, 'A support system for predicting eBay end prices', *Decision Support Systems*, vol. 44, no. 4, pp. 970-82.

Vergano, D. 2006, 'On eBay, It Pays to Snipe', *USA Today*, vol. 6. 25 June. viewed 30 January 2014, <http://usatoday30.usatoday.com/tech/science/columnist/2006-06-25-physics-of-ebay_x.htm>

Vorobeychik, Y. 2011, 'A Game Theoretic Bidding Agent for the Ad Auction Game', in Joaquim Filipe & Ana L. N. Fred (ed), *International Conference on Agents and Artificial Intelligence (2)*, SciTePress, pp. 35-44.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G. & Guo, S.-P. 2011, 'Forecasting stock indices with back propagation neural network', *Expert Systems with Applications*, vol. 38, no. 11, pp. 14346-55.

Wang, S., Jank, W. & Shmueli, G. 2008, 'Explaining and forecasting online auction prices and their dynamics using functional data analysis', *Journal of Business and Economic Statistics*, vol. 26, no. 2, pp. 144-60.

White, H. 1988, 'Economic prediction using neural networks: The case of IBM daily stock returns', *IEEE International Conference on Neural Networks*, IEEE, pp. 451-8.

Wilcox, R.T. 2000, 'Experts and amateurs: The role of experience in Internet auctions', *Marketing Letters*, vol. 11, no. 4, pp. 363-74.

Wintr, L. 2008, 'Some evidence on late bidding in eBay auctions', *Economic Inquiry*, vol. 46, no. 3, pp. 369-79.

Wooldridge, M. 2008, *An Introduction to Multiagent Systems*, electronic book, John Wiley & Sons, London, UK.

Xuefeng, L., Lu, L., Lihua, W. & Zhao, Z. 2006, 'Predicting the final prices of online auction items', *Expert Systems with Applications*, vol. 31, no. 3, pp. 542-50.

Yoon, Y. & Swales, G. 1991, 'Predicting stock price performance: A neural network approach', in *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, vol. 4, IEEE, pp. 156-62.

Yu, H. & Liu, Z. 2008, 'Heuristic bidding algorithm using fuzzy neural networks in multiple multi-attribute auctions', *IEEE International Conference on Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI* vol. 1, IEEE, pp. 1415-20.

Zeng, D. & Sycara, K. 1997, 'How can an agent learn to negotiate?', in J.P. Müller, M.J. Wooldridge & N.R. Jennings (eds), *Intelligent Agents III Agent Theories, Architectures, and Languages*, Springer, pp. 233-44.

Zhang, J., Prater, E.L. & Lipkin, I. 2012, 'Feedback Reviews and Bidding in Online Auctions: An Integrated Hedonic Regression and Fuzzy Logic Expert System Approach', *Decision Support Systems*, vol. 55, no. 4, pp. 894-902.

Zhang, S., Jank, W. & Shmueli, G. 2010, 'Real-time forecasting of online auctions via functional K-nearest neighbors', *International Journal of Forecasting*, vol. 26, no. 4, pp. 666-83.

Appendices

A. Simulation Results

This appendix shows the expected utility of heterogeneous as well as homogenous groups of *NDF-DC* bidder agents across three different auction settings: low-, medium- and high- bid rate. The results of a sample of completed simulation runs are given in the tables below. Each table shows the expected utility of bidder agents with specific bidding characteristics (categorised as *Mystical*, *Sturdy* or *Strategic* in this thesis) based on their bidding behaviour.

Table A.1 Heterogeneous Mystical bidders in low-bid-rate auctions

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*
1	85	0.045	0.2	0.368
2	13	0.261	0.9	0.246
3	95	0.015	0.2	0.373
4	18	0.246	0.9	0.247
5	84	0.048	0.2	0.368
6	78	0.066	0.2	0.364
7	33	0.201	0.6	0.270
8	64	0.108	0.3	0.327
9	50	0.150	0.6	0.274
10	9	0.273	0.9	0.245
11	25	0.225	0.9	0.248
12	91	0.027	0.2	0.371
13	79	0.063	0.2	0.365
14	68	0.096	0.3	0.328
15	22	0.234	0.9	0.248
16	80	0.060	0.2	0.365
17	82	0.054	0.2	0.367
18	63	0.111	0.3	0.326
19	1	0.297	0.9	0.244
20	54	0.138	0.3	0.322
21	79	0.063	0.2	0.365
22	32	0.204	0.6	0.269
23	6	0.282	0.9	0.245
24	70	0.090	0.3	0.329
25	99	0.003	0.2	0.376
26	35	0.195	0.6	0.270
27	9	0.273	0.9	0.245
28	58	0.126	0.3	0.324
29	11	0.267	0.9	0.246
30	69	0.093	0.3	0.329
31	49	0.153	0.6	0.274
32	73	0.081	0.3	0.331
33	18	0.246	0.9	0.247
34	3	0.291	0.9	0.244
51	31	0.207	0.6	0.269
52	54	0.138	0.3	0.322
53	77	0.069	0.2	0.364
54	47	0.159	0.6	0.273
55	1	0.297	0.9	0.244
56	73	0.081	0.3	0.331
57	37	0.189	0.6	0.271
58	62	0.114	0.3	0.326
59	13	0.261	0.9	0.246
60	60	0.120	0.3	0.325
61	29	0.213	0.6	0.269
62	8	0.276	0.9	0.245
63	72	0.084	0.3	0.330
64	8	0.276	0.9	0.245
65	91	0.027	0.2	0.371
66	60	0.120	0.3	0.325
67	20	0.240	0.9	0.247
68	33	0.201	0.6	0.270
69	45	0.165	0.6	0.273
70	49	0.153	0.6	0.274
71	33	0.201	0.6	0.270
72	4	0.288	0.9	0.244
73	74	0.078	0.3	0.331
74	88	0.036	0.2	0.370
75	40	0.180	0.6	0.272
76	75	0.075	0.3	0.331
77	76	0.072	0.2	0.363
78	15	0.255	0.9	0.246
79	6	0.282	0.9	0.245
80	58	0.126	0.3	0.324
81	96	0.012	0.2	0.374
82	6	0.282	0.9	0.245
83	79	0.063	0.2	0.365
84	70	0.090	0.3	0.329

35	68	0.096	0.3	0.328	85	79	0.063	0.2	0.365
36	14	0.258	0.9	0.246	86	45	0.165	0.6	0.273
37	72	0.084	0.3	0.330	87	8	0.276	0.9	0.245
38	83	0.051	0.2	0.367	88	94	0.018	0.2	0.373
39	4	0.288	0.9	0.244	89	49	0.153	0.6	0.274
40	38	0.186	0.6	0.271	90	56	0.132	0.3	0.323
41	32	0.204	0.6	0.269	91	16	0.252	0.9	0.247
42	54	0.138	0.3	0.322	92	75	0.075	0.3	0.331
43	58	0.126	0.3	0.324	93	100	0.000	0.2	0.376
44	65	0.105	0.3	0.327	94	43	0.171	0.6	0.272
45	96	0.012	0.2	0.374	95	37	0.189	0.6	0.271
46	97	0.009	0.2	0.374	96	1	0.297	0.9	0.244
47	78	0.066	0.2	0.364	97	30	0.210	0.6	0.269
48	57	0.129	0.3	0.324	98	81	0.057	0.2	0.366
49	12	0.264	0.9	0.246	99	72	0.084	0.3	0.330
50	98	0.006	0.2	0.375	100	52	0.144	0.3	0.322

Table A.2 Heterogeneous Mystical bidders in medium-bid-rate auctions

Simulation Run	***Bargain Level***	***k***	***β***	***Expected Utility***	***Simulation Run***	***Bargain Level***	***k***	***β***	***Expected Utility***
1	64	0.108	0.3	0.416	51	34	0.198	0.6	0.324
2	23	0.231	0.9	0.289	52	9	0.273	0.9	0.284
3	78	0.066	0.2	0.477	53	28	0.216	0.6	0.322
4	47	0.159	0.6	0.330	54	9	0.273	0.9	0.284
5	23	0.231	0.9	0.289	55	93	0.021	0.2	0.490
6	3	0.291	0.9	0.282	56	71	0.087	0.3	0.421
7	14	0.258	0.9	0.286	57	61	0.117	0.3	0.414
8	90	0.03	0.2	0.488	58	52	0.144	0.3	0.408
9	85	0.045	0.2	0.483	59	10	0.27	0.9	0.285
10	72	0.084	0.3	0.422	60	6	0.282	0.9	0.283
11	70	0.09	0.3	0.421	61	58	0.126	0.3	0.412
12	24	0.228	0.9	0.289	62	63	0.111	0.3	0.416
13	88	0.036	0.2	0.486	63	80	0.06	0.2	0.479
14	96	0.012	0.2	0.493	64	1	0.297	0.9	0.282
15	59	0.123	0.3	0.413	65	27	0.219	0.6	0.321
16	37	0.189	0.6	0.326	66	97	0.009	0.2	0.494
17	42	0.174	0.6	0.328	67	48	0.156	0.6	0.330
18	67	0.099	0.3	0.418	68	83	0.051	0.2	0.482
19	10	0.27	0.9	0.285	69	11	0.267	0.9	0.285
20	47	0.159	0.6	0.330	70	71	0.087	0.3	0.421
21	88	0.036	0.2	0.486	71	32	0.204	0.6	0.324
22	94	0.018	0.2	0.491	72	13	0.261	0.9	0.286
23	35	0.195	0.6	0.325	73	74	0.078	0.3	0.423
24	30	0.21	0.6	0.323	74	75	0.075	0.3	0.424
25	15	0.255	0.9	0.286	75	78	0.066	0.2	0.477
26	73	0.081	0.3	0.423	76	10	0.27	0.9	0.285
27	55	0.135	0.3	0.410	77	89	0.033	0.2	0.487
28	84	0.048	0.2	0.483	78	100	0	0.2	0.496
29	72	0.084	0.3	0.422	79	74	0.078	0.3	0.423
30	34	0.198	0.6	0.324	80	27	0.219	0.6	0.321
31	31	0.207	0.6	0.323	81	29	0.213	0.6	0.322
32	54	0.138	0.3	0.409	82	14	0.258	0.9	0.286
33	70	0.09	0.3	0.421	83	90	0.03	0.2	0.488
34	18	0.246	0.9	0.287	84	42	0.174	0.6	0.328
35	92	0.024	0.2	0.489	85	8	0.276	0.9	0.284
36	60	0.12	0.3	0.414	86	45	0.165	0.6	0.329
37	55	0.135	0.3	0.410	87	84	0.048	0.2	0.483
38	66	0.102	0.3	0.418	88	70	0.09	0.3	0.421
39	20	0.24	0.9	0.288	89	4	0.288	0.9	0.283
40	21	0.237	0.9	0.288	90	38	0.186	0.6	0.326
41	4	0.288	0.9	0.283	91	43	0.171	0.6	0.328
42	83	0.051	0.2	0.482	92	86	0.042	0.2	0.484
43	3	0.291	0.9	0.282	93	18	0.246	0.9	0.287
44	16	0.252	0.9	0.286	94	98	0.006	0.2	0.495
45	31	0.207	0.6	0.323	95	50	0.15	0.6	0.331
46	6	0.282	0.9	0.283	96	88	0.036	0.2	0.486
47	15	0.255	0.9	0.286	97	87	0.039	0.2	0.485
48	79	0.063	0.2	0.478	98	44	0.168	0.6	0.329
49	24	0.228	0.9	0.289	99	65	0.105	0.3	0.417
50	79	0.063	0.2	0.478	100	28	0.216	0.6	0.322

Table A.3 Heterogeneous Mystical bidders in high-bid-rate auctions

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*
1	91	0.027	0.2	0.469
2	91	0.027	0.2	0.469
3	16	0.252	0.9	0.129
4	2	0.294	0.9	0.122
5	62	0.114	0.3	0.345
6	91	0.027	0.2	0.469
7	53	0.141	0.3	0.335
8	49	0.153	0.6	0.204
9	30	0.210	0.6	0.190
10	77	0.069	0.2	0.449
11	88	0.036	0.2	0.465
12	79	0.063	0.2	0.452
13	17	0.249	0.9	0.130
14	64	0.108	0.3	0.348
15	44	0.168	0.6	0.200
16	33	0.201	0.6	0.192
17	50	0.150	0.6	0.205
18	10	0.270	0.9	0.126
19	55	0.135	0.3	0.337
20	46	0.162	0.6	0.202
21	38	0.186	0.6	0.196
22	22	0.234	0.9	0.132
23	97	0.009	0.2	0.478
24	43	0.171	0.6	0.200
25	61	0.117	0.3	0.344
26	38	0.186	0.6	0.196
27	73	0.081	0.3	0.358
28	40	0.180	0.6	0.197
29	13	0.261	0.9	0.128
30	7	0.279	0.9	0.125
31	74	0.078	0.3	0.359
32	31	0.207	0.6	0.191
33	13	0.261	0.9	0.128
34	1	0.297	0.9	0.122
35	24	0.228	0.9	0.133
36	67	0.099	0.3	0.351
37	62	0.114	0.3	0.345
38	38	0.186	0.6	0.196
39	19	0.243	0.9	0.131
40	56	0.132	0.3	0.338
41	71	0.087	0.3	0.356
42	4	0.288	0.9	0.123
43	15	0.255	0.9	0.129
44	86	0.042	0.2	0.462
45	66	0.102	0.3	0.350
46	64	0.108	0.3	0.348
47	55	0.135	0.3	0.337
48	40	0.180	0.6	0.197
49	8	0.276	0.9	0.125
50	23	0.231	0.9	0.133
51	43	0.171	0.6	0.200
52	84	0.048	0.2	0.459
53	50	0.150	0.6	0.205
54	59	0.123	0.3	0.342
55	1	0.297	0.9	0.122
56	4	0.288	0.9	0.123
57	79	0.063	0.2	0.452
58	77	0.069	0.2	0.449
59	15	0.255	0.9	0.129
60	19	0.243	0.9	0.131
61	85	0.045	0.2	0.460
62	3	0.291	0.9	0.123
63	43	0.171	0.6	0.200
64	51	0.147	0.3	0.332
65	84	0.048	0.2	0.459
66	85	0.045	0.2	0.460
67	97	0.009	0.2	0.478
68	19	0.243	0.9	0.131
69	70	0.090	0.3	0.355
70	46	0.162	0.6	0.202
71	45	0.165	0.6	0.201
72	94	0.018	0.2	0.473
73	90	0.030	0.2	0.468
74	73	0.081	0.3	0.358
75	34	0.198	0.6	0.193
76	51	0.147	0.3	0.332
77	88	0.036	0.2	0.465
78	25	0.225	0.9	0.134
79	27	0.219	0.6	0.188
80	62	0.114	0.3	0.345
81	25	0.225	0.9	0.134
82	35	0.195	0.6	0.194
83	77	0.069	0.2	0.449
84	12	0.264	0.9	0.127
85	88	0.036	0.2	0.465
86	35	0.195	0.6	0.194
87	66	0.102	0.3	0.350
88	3	0.291	0.9	0.123
89	68	0.096	0.3	0.352
90	54	0.138	0.3	0.336
91	68	0.096	0.3	0.352
92	86	0.042	0.2	0.462
93	22	0.234	0.9	0.132
94	22	0.234	0.9	0.132
95	79	0.063	0.2	0.452
96	49	0.153	0.6	0.204
97	11	0.267	0.9	0.127
98	71	0.087	0.3	0.356
99	69	0.093	0.3	0.354
100	56	0.132	0.3	0.338

Table A.4 Heterogeneous Sturdy bidders in low-bid-rate auctions

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*	*Simulation Run*	*Bargain Level*	*k*	*β*	*Expected Utility*
1	68	0.096	0.3	0.659	51	75	0.075	0.3	0.667
2	81	0.057	0.2	0.677	52	99	0.003	0.2	0.697
3	2	0.294	0.9	0.531	53	55	0.135	0.3	0.645
4	38	0.186	0.6	0.598	54	2	0.294	0.9	0.531
5	67	0.099	0.3	0.658	55	30	0.210	0.6	0.590
6	65	0.105	0.3	0.656	56	58	0.126	0.3	0.648
7	44	0.168	0.6	0.605	57	7	0.279	0.9	0.536
8	92	0.024	0.2	0.689	58	36	0.192	0.6	0.596
9	42	0.174	0.6	0.602	59	71	0.087	0.3	0.663
10	27	0.219	0.6	0.587	60	53	0.141	0.3	0.643
11	89	0.033	0.2	0.686	61	29	0.213	0.6	0.589
12	95	0.015	0.2	0.693	62	91	0.027	0.2	0.688
13	55	0.135	0.3	0.645	63	5	0.285	0.9	0.534
14	100	0.000	0.2	0.698	64	58	0.126	0.3	0.648
15	42	0.174	0.6	0.602	65	64	0.108	0.3	0.655
16	40	0.180	0.6	0.600	66	90	0.030	0.2	0.687
17	69	0.093	0.3	0.660	67	89	0.033	0.2	0.686
18	52	0.144	0.3	0.641	68	4	0.288	0.9	0.533
19	46	0.162	0.6	0.607	69	52	0.144	0.3	0.641
20	59	0.123	0.3	0.649	70	88	0.036	0.2	0.685
21	61	0.117	0.3	0.651	71	35	0.195	0.6	0.595
22	64	0.108	0.3	0.655	72	78	0.066	0.2	0.674
23	33	0.201	0.6	0.593	73	84	0.048	0.2	0.680
24	70	0.090	0.3	0.662	74	37	0.189	0.6	0.597
25	23	0.231	0.9	0.550	75	44	0.168	0.6	0.605
26	22	0.234	0.9	0.549	76	41	0.177	0.6	0.601
27	49	0.153	0.6	0.610	77	18	0.246	0.9	0.545
28	67	0.099	0.3	0.658	78	46	0.162	0.6	0.607
29	37	0.189	0.6	0.597	79	49	0.153	0.6	0.610
30	53	0.141	0.3	0.643	80	67	0.099	0.3	0.658
31	46	0.162	0.6	0.607	81	29	0.213	0.6	0.589
32	43	0.171	0.6	0.604	82	57	0.129	0.3	0.647
33	49	0.153	0.6	0.610	83	65	0.105	0.3	0.656
34	48	0.156	0.6	0.609	84	1	0.297	0.9	0.530
35	30	0.210	0.6	0.590	85	77	0.069	0.2	0.672
36	76	0.072	0.2	0.671	86	46	0.162	0.6	0.607
37	98	0.006	0.2	0.696	87	85	0.045	0.2	0.681
38	58	0.126	0.3	0.648	88	57	0.129	0.3	0.647
39	88	0.036	0.2	0.685	89	53	0.141	0.3	0.643
40	15	0.255	0.9	0.543	90	72	0.084	0.3	0.664
41	90	0.030	0.2	0.687	91	20	0.240	0.9	0.547
42	52	0.144	0.3	0.641	92	25	0.225	0.9	0.552
43	82	0.054	0.2	0.678	93	97	0.009	0.2	0.695
44	64	0.108	0.3	0.655	94	37	0.189	0.6	0.597
45	7	0.279	0.9	0.536	95	59	0.123	0.3	0.649
46	4	0.288	0.9	0.533	96	45	0.165	0.6	0.606
47	75	0.075	0.3	0.667	97	46	0.162	0.6	0.607
48	3	0.291	0.9	0.532	98	90	0.030	0.2	0.687
49	75	0.075	0.3	0.667	99	39	0.183	0.6	0.599
50	55	0.135	0.3	0.645	100	3	0.291	0.9	0.532

Table A.5 Heterogeneous Sturdy bidders in medium-bid-rate auctions

Simulation Run	***Bargain Level***	***k***	***β***	***Expected Utility***	***Simulation Run***	***Bargain Level***	***k***	***β***	***Expected Utility***
1	63	0.111	0.3	0.863	51	41	0.177	0.6	0.779
2	33	0.201	0.6	0.766	52	32	0.204	0.6	0.765
3	28	0.216	0.6	0.758	53	89	0.033	0.2	0.914
4	73	0.081	0.3	0.880	54	5	0.285	0.9	0.671
5	91	0.027	0.2	0.918	55	87	0.039	0.2	0.910
6	23	0.231	0.9	0.697	56	51	0.147	0.3	0.841
7	92	0.024	0.2	0.919	57	73	0.081	0.3	0.880
8	70	0.09	0.3	0.875	58	61	0.117	0.3	0.859
9	28	0.216	0.6	0.758	59	14	0.258	0.9	0.684
10	54	0.138	0.3	0.846	60	21	0.237	0.9	0.694
11	69	0.093	0.3	0.873	61	64	0.108	0.3	0.864
12	75	0.075	0.3	0.884	62	88	0.036	0.2	0.912
13	11	0.267	0.9	0.680	63	46	0.162	0.6	0.787
14	62	0.114	0.3	0.861	64	97	0.009	0.2	0.928
15	81	0.057	0.2	0.900	65	99	0.003	0.2	0.932
16	37	0.189	0.6	0.773	66	26	0.222	0.6	0.755
17	4	0.288	0.9	0.670	67	57	0.129	0.3	0.852
18	85	0.045	0.2	0.907	68	52	0.144	0.3	0.843
19	71	0.087	0.3	0.877	69	38	0.186	0.6	0.774
20	55	0.135	0.3	0.848	70	22	0.234	0.9	0.695
21	72	0.084	0.3	0.879	71	17	0.249	0.9	0.688
22	52	0.144	0.3	0.843	72	13	0.261	0.9	0.683
23	14	0.258	0.9	0.684	73	9	0.273	0.9	0.677
24	86	0.042	0.2	0.909	74	69	0.093	0.3	0.873
25	53	0.141	0.3	0.845	75	81	0.057	0.2	0.900
26	30	0.21	0.6	0.761	76	97	0.009	0.2	0.928
27	97	0.009	0.2	0.928	77	21	0.237	0.9	0.694
28	5	0.285	0.9	0.671	78	42	0.174	0.6	0.781
29	3	0.291	0.9	0.669	79	73	0.081	0.3	0.880
30	50	0.15	0.6	0.794	80	99	0.003	0.2	0.932
31	49	0.153	0.6	0.792	81	71	0.087	0.3	0.877
32	48	0.156	0.6	0.791	82	35	0.195	0.6	0.769
33	96	0.012	0.2	0.927	83	9	0.273	0.9	0.677
34	13	0.261	0.9	0.683	84	20	0.24	0.9	0.693
35	30	0.21	0.6	0.761	85	23	0.231	0.9	0.697
36	50	0.15	0.6	0.794	86	71	0.087	0.3	0.877
37	17	0.249	0.9	0.688	87	53	0.141	0.3	0.845
38	9	0.273	0.9	0.677	88	9	0.273	0.9	0.677
39	28	0.216	0.6	0.758	89	87	0.039	0.2	0.910
40	52	0.144	0.3	0.843	90	88	0.036	0.2	0.912
41	95	0.015	0.2	0.925	91	50	0.15	0.6	0.794
42	25	0.225	0.9	0.700	92	74	0.078	0.3	0.882
43	27	0.219	0.6	0.756	93	78	0.066	0.2	0.894
44	5	0.285	0.9	0.671	94	51	0.147	0.3	0.841
45	32	0.204	0.6	0.765	95	39	0.183	0.6	0.776
46	70	0.09	0.3	0.875	96	64	0.108	0.3	0.864
47	27	0.219	0.6	0.756	97	93	0.021	0.2	0.921
48	23	0.231	0.9	0.697	98	67	0.099	0.3	0.870
49	5	0.285	0.9	0.671	99	70	0.09	0.3	0.875
50	95	0.015	0.2	0.925	100	66	0.102	0.3	0.868

Table A.6 Heterogeneous Sturdy bidders in high-bid-rate auctions

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*
1	20	0.240	0.9	0.478
2	72	0.084	0.3	0.726
3	92	0.024	0.2	0.780
4	11	0.267	0.9	0.461
5	55	0.135	0.3	0.685
6	65	0.105	0.3	0.709
7	37	0.189	0.6	0.584
8	48	0.156	0.6	0.608
9	50	0.150	0.6	0.613
10	92	0.024	0.2	0.780
11	31	0.207	0.6	0.571
12	81	0.057	0.2	0.754
13	44	0.168	0.6	0.600
14	14	0.258	0.9	0.466
15	37	0.189	0.6	0.584
16	78	0.066	0.2	0.746
17	79	0.063	0.2	0.749
18	87	0.039	0.2	0.768
19	87	0.039	0.2	0.768
20	80	0.060	0.2	0.751
21	90	0.030	0.2	0.775
22	58	0.126	0.3	0.692
23	96	0.012	0.2	0.790
24	9	0.273	0.9	0.457
25	20	0.240	0.9	0.478
26	89	0.033	0.2	0.773
27	56	0.132	0.3	0.688
28	57	0.129	0.3	0.690
29	47	0.159	0.6	0.606
30	13	0.261	0.9	0.464
31	61	0.117	0.3	0.699
32	13	0.261	0.9	0.464
33	73	0.081	0.3	0.728
34	64	0.108	0.3	0.707
35	68	0.096	0.3	0.716
36	50	0.150	0.6	0.613
37	40	0.180	0.6	0.591
38	91	0.027	0.2	0.778
39	68	0.096	0.3	0.716
40	77	0.069	0.2	0.744
41	2	0.294	0.9	0.444
42	75	0.075	0.3	0.733
43	36	0.192	0.6	0.582
44	31	0.207	0.6	0.571
45	22	0.234	0.9	0.481
46	43	0.171	0.6	0.597
47	19	0.243	0.9	0.476
48	12	0.264	0.9	0.463
49	70	0.090	0.3	0.721
50	31	0.207	0.6	0.571
51	54	0.138	0.3	0.683
52	9	0.273	0.9	0.457
53	62	0.114	0.3	0.702
54	13	0.261	0.9	0.464
55	9	0.273	0.9	0.457
56	79	0.063	0.2	0.749
57	69	0.093	0.3	0.718
58	53	0.141	0.3	0.680
59	42	0.174	0.6	0.595
60	54	0.138	0.3	0.683
61	11	0.267	0.9	0.461
62	5	0.285	0.9	0.449
63	47	0.159	0.6	0.606
64	1	0.297	0.9	0.442
65	63	0.111	0.3	0.704
66	34	0.198	0.6	0.578
67	10	0.270	0.9	0.459
68	94	0.018	0.2	0.785
69	86	0.042	0.2	0.766
70	77	0.069	0.2	0.744
71	32	0.204	0.6	0.574
72	66	0.102	0.3	0.711
73	1	0.297	0.9	0.442
74	58	0.126	0.3	0.692
75	44	0.168	0.6	0.600
76	27	0.219	0.6	0.563
77	65	0.105	0.3	0.709
78	6	0.282	0.9	0.451
79	15	0.255	0.9	0.468
80	50	0.150	0.6	0.613
81	37	0.189	0.6	0.584
82	81	0.057	0.2	0.754
83	37	0.189	0.6	0.584
84	86	0.042	0.2	0.766
85	14	0.258	0.9	0.466
86	22	0.234	0.9	0.481
87	5	0.285	0.9	0.449
88	23	0.231	0.9	0.483
89	15	0.255	0.9	0.468
90	80	0.060	0.2	0.751
91	77	0.069	0.2	0.744
92	44	0.168	0.6	0.600
93	74	0.078	0.3	0.730
94	43	0.171	0.6	0.597
95	25	0.225	0.9	0.487
96	78	0.066	0.2	0.746
97	11	0.267	0.9	0.461
98	88	0.036	0.2	0.770
99	87	0.039	0.2	0.768
100	21	0.237	0.9	0.480

Table A.7 Heterogeneous Strategic bidders in low-bid-rate auctions

Simulation Run	***Bargain Level***	***k***	***β***	***Expected Utility***	***Simulation Run***	***Bargain Level***	***k***	***β***	***Expected Utility***
1	43	0.342	0.6	0.029	51	29	0.426	0.6	0.047
2	11	0.534	0.9	0.020	52	20	0.480	0.9	0.024
3	97	0.018	0.2	0.106	53	73	0.162	0.3	0.238
4	43	0.342	0.6	0.026	54	78	0.132	0.2	0.028
5	61	0.234	0.3	0.021	55	6	0.564	0.9	0.005
6	46	0.324	0.6	0.037	56	97	0.018	0.2	0.241
7	59	0.246	0.3	0.062	57	9	0.546	0.9	0.019
8	60	0.240	0.3	0.343	58	86	0.084	0.2	0.042
9	73	0.162	0.3	0.216	59	14	0.516	0.9	0.021
10	54	0.276	0.3	0.046	60	43	0.342	0.6	0.114
11	83	0.102	0.2	0.586	61	11	0.534	0.9	0.020
12	97	0.018	0.2	0.708	62	72	0.168	0.3	0.427
13	66	0.204	0.3	0.067	63	36	0.384	0.6	0.034
14	29	0.426	0.6	0.014	64	68	0.192	0.3	0.026
15	93	0.042	0.2	0.259	65	65	0.210	0.3	0.073
16	44	0.336	0.6	0.119	66	73	0.162	0.3	0.434
17	12	0.528	0.9	0.005	67	58	0.252	0.3	0.140
18	3	0.582	0.9	0.017	68	79	0.126	0.2	0.392
19	3	0.582	0.9	0.017	69	48	0.312	0.6	0.014
20	81	0.114	0.2	0.025	70	74	0.156	0.3	0.441
21	82	0.108	0.2	0.102	71	45	0.330	0.6	0.100
22	57	0.258	0.3	0.018	72	53	0.282	0.3	0.295
23	3	0.582	0.9	0.017	73	82	0.108	0.2	0.269
24	14	0.516	0.9	0.007	74	30	0.420	0.6	0.028
25	25	0.450	0.9	0.019	75	72	0.168	0.3	0.023
26	85	0.090	0.2	0.205	76	99	0.006	0.2	0.348
27	98	0.012	0.2	0.043	77	66	0.204	0.3	0.384
28	28	0.432	0.6	0.025	78	82	0.108	0.2	0.578
29	23	0.462	0.9	0.026	79	19	0.486	0.9	0.024
30	92	0.048	0.2	0.032	80	32	0.408	0.6	0.061
31	93	0.042	0.2	0.673	81	32	0.408	0.6	0.013
32	7	0.558	0.9	0.007	82	33	0.402	0.6	0.066
33	10	0.540	0.9	0.005	83	83	0.102	0.2	0.037
34	52	0.288	0.3	0.289	84	34	0.396	0.6	0.071
35	83	0.102	0.2	0.079	85	64	0.216	0.3	0.146
36	7	0.558	0.9	0.005	86	60	0.240	0.3	0.246
37	59	0.246	0.3	0.069	87	77	0.138	0.2	0.241
38	13	0.522	0.9	0.019	88	8	0.552	0.9	0.019
39	37	0.378	0.6	0.085	89	89	0.066	0.2	0.031
40	55	0.270	0.3	0.019	90	23	0.462	0.9	0.026
41	14	0.516	0.9	0.014	91	76	0.144	0.2	0.075
42	80	0.120	0.2	0.312	92	77	0.138	0.2	0.041
43	62	0.228	0.3	0.144	93	71	0.174	0.3	0.420
44	94	0.036	0.2	0.110	94	80	0.120	0.2	0.069
45	32	0.408	0.6	0.061	95	84	0.096	0.2	0.038
46	64	0.216	0.3	0.028	96	87	0.078	0.2	0.298
47	30	0.420	0.6	0.030	97	42	0.348	0.6	0.109
48	29	0.426	0.6	0.047	98	72	0.168	0.3	0.427
49	44	0.336	0.6	0.029	99	25	0.450	0.9	0.027
50	9	0.546	0.9	0.006	100	43	0.342	0.6	0.011

Table A.8 Heterogeneous Strategic bidders in medium-bid-rate auctions

Simulation Run	***Bargain Level***	***k***	***β***	***Expected Utility***
1	57	0.258	0.3	0.083
2	71	0.174	0.3	0.033
3	61	0.234	0.3	0.097
4	51	0.294	0.3	0.078
5	16	0.504	0.9	0.006
6	10	0.540	0.9	0.004
7	71	0.174	0.3	0.024
8	72	0.168	0.3	0.121
9	86	0.084	0.2	0.035
10	1	0.594	0.9	0.003
11	74	0.156	0.3	0.135
12	86	0.084	0.2	0.037
13	72	0.168	0.3	0.121
14	28	0.432	0.6	0.019
15	58	0.252	0.3	0.091
16	42	0.348	0.6	0.029
17	11	0.534	0.9	0.004
18	15	0.510	0.9	0.006
19	38	0.372	0.6	0.026
20	56	0.264	0.3	0.020
21	93	0.042	0.2	0.368
22	93	0.042	0.2	0.339
23	86	0.084	0.2	0.038
24	26	0.444	0.6	0.018
25	60	0.240	0.3	0.066
26	75	0.150	0.3	0.143
27	25	0.450	0.9	0.004
28	53	0.282	0.3	0.079
29	47	0.318	0.6	0.015
30	7	0.558	0.9	0.004
31	99	0.006	0.2	0.135
32	37	0.378	0.6	0.025
33	24	0.456	0.9	0.008
34	96	0.024	0.2	0.103
35	39	0.366	0.6	0.027
36	99	0.006	0.2	0.402
37	79	0.126	0.2	0.230
38	96	0.024	0.2	0.343
39	15	0.510	0.9	0.006
40	19	0.486	0.9	0.007
41	57	0.258	0.3	0.075
42	77	0.138	0.2	0.220
43	37	0.378	0.6	0.025
44	82	0.108	0.2	0.265
45	22	0.468	0.9	0.008
46	51	0.294	0.3	0.078
47	8	0.552	0.9	0.004
48	30	0.420	0.6	0.020
49	72	0.168	0.3	0.121
50	79	0.126	0.2	0.034

Simulation Run	***Bargain Level***	***k***	***β***	***Expected Utility***
51	24	0.456	0.9	0.005
52	82	0.108	0.2	0.265
53	46	0.324	0.6	0.032
54	52	0.288	0.3	0.080
55	47	0.318	0.6	0.016
56	50	0.300	0.6	0.014
57	1	0.594	0.9	0.003
58	66	0.204	0.3	0.108
59	6	0.564	0.9	0.004
60	3	0.582	0.9	0.003
61	22	0.468	0.9	0.008
62	32	0.408	0.6	0.022
63	28	0.432	0.6	0.019
64	51	0.294	0.3	0.078
65	72	0.168	0.3	0.023
66	24	0.456	0.9	0.004
67	46	0.324	0.6	0.032
68	47	0.318	0.6	0.033
69	72	0.168	0.3	0.121
70	2	0.588	0.9	0.003
71	97	0.018	0.2	0.406
72	98	0.012	0.2	0.306
73	24	0.456	0.9	0.006
74	65	0.210	0.3	0.106
75	13	0.522	0.9	0.005
76	3	0.582	0.9	0.003
77	26	0.444	0.6	0.018
78	78	0.132	0.2	0.040
79	74	0.156	0.3	0.036
80	20	0.480	0.9	0.007
81	98	0.012	0.2	0.416
82	19	0.486	0.9	0.007
83	89	0.066	0.2	0.045
84	97	0.018	0.2	0.406
85	31	0.414	0.6	0.009
86	9	0.546	0.9	0.004
87	68	0.192	0.3	0.024
88	44	0.336	0.6	0.014
89	82	0.108	0.2	0.094
90	59	0.246	0.3	0.084
91	19	0.486	0.9	0.004
92	55	0.270	0.3	0.085
93	96	0.024	0.2	0.397
94	66	0.204	0.3	0.085
95	92	0.048	0.2	0.092
96	89	0.066	0.2	0.330
97	52	0.288	0.3	0.073
98	41	0.354	0.6	0.012
99	86	0.084	0.2	0.302
100	94	0.036	0.2	0.377

Table A.9 Heterogeneous Strategic bidders in high-bid-rate auctions

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*	*Simulation Run*	*Bargain Level*	*k*	*β*	*Expected Utility*
1	48	0.312	0.6	0.011	51	21	0.474	0.9	0.002
2	97	0.018	0.2	0.099	52	71	0.174	0.3	0.044
3	99	0.006	0.2	0.103	53	3	0.582	0.9	0.001
4	88	0.072	0.2	0.046	54	97	0.018	0.2	0.094
5	91	0.054	0.2	0.046	55	76	0.144	0.2	0.041
6	80	0.12	0.2	0.071	56	48	0.312	0.6	0.011
7	7	0.558	0.9	0.001	57	95	0.03	0.2	0.088
8	73	0.162	0.3	0.046	58	22	0.468	0.9	0.002
9	53	0.282	0.3	0.026	59	42	0.348	0.6	0.009
10	41	0.354	0.6	0.009	60	82	0.108	0.2	0.074
11	32	0.408	0.6	0.007	61	27	0.438	0.6	0.006
12	4	0.576	0.9	0.001	62	96	0.024	0.2	0.097
13	44	0.336	0.6	0.010	63	88	0.072	0.2	0.048
14	27	0.438	0.6	0.006	64	88	0.072	0.2	0.073
15	10	0.54	0.9	0.002	65	99	0.006	0.2	0.103
16	94	0.036	0.2	0.077	66	10	0.54	0.9	0.002
17	4	0.576	0.9	0.001	67	85	0.09	0.2	0.079
18	21	0.474	0.9	0.002	68	40	0.36	0.6	0.009
19	63	0.222	0.3	0.037	69	53	0.282	0.3	0.029
20	9	0.546	0.9	0.002	70	61	0.234	0.3	0.028
21	13	0.522	0.9	0.002	71	4	0.576	0.9	0.001
22	95	0.03	0.2	0.096	72	98	0.012	0.2	0.101
23	63	0.222	0.3	0.037	73	69	0.186	0.3	0.042
24	82	0.108	0.2	0.073	74	100	0	0.2	0.105
25	69	0.186	0.3	0.022	75	81	0.114	0.2	0.073
26	71	0.174	0.3	0.044	76	58	0.252	0.3	0.028
27	70	0.18	0.3	0.022	77	26	0.444	0.6	0.006
28	85	0.09	0.2	0.041	78	24	0.456	0.9	0.003
29	33	0.402	0.6	0.007	79	100	0	0.2	0.051
30	30	0.42	0.6	0.007	80	82	0.108	0.2	0.074
31	94	0.036	0.2	0.089	81	91	0.054	0.2	0.083
32	6	0.564	0.9	0.001	82	91	0.054	0.2	0.089
33	4	0.576	0.9	0.001	83	51	0.294	0.3	0.027
34	10	0.54	0.9	0.002	84	20	0.48	0.9	0.002
35	91	0.054	0.2	0.030	85	59	0.246	0.3	0.033
36	97	0.018	0.2	0.099	86	79	0.126	0.2	0.070
37	79	0.126	0.2	0.064	87	94	0.036	0.2	0.036
38	25	0.45	0.9	0.003	88	93	0.042	0.2	0.079
39	100	0	0.2	0.105	89	50	0.3	0.6	0.011
40	33	0.402	0.6	0.007	90	79	0.126	0.2	0.070
41	95	0.03	0.2	0.096	91	98	0.012	0.2	0.101
42	40	0.36	0.6	0.009	92	84	0.096	0.2	0.077
43	61	0.234	0.3	0.035	93	8	0.552	0.9	0.001
44	17	0.498	0.9	0.002	94	35	0.39	0.6	0.008
45	43	0.342	0.6	0.010	95	8	0.552	0.9	0.001
46	9	0.546	0.9	0.002	96	33	0.402	0.6	0.007
47	76	0.144	0.2	0.066	97	52	0.288	0.3	0.028
48	59	0.246	0.3	0.033	98	95	0.03	0.2	0.032
49	3	0.582	0.9	0.001	99	41	0.354	0.6	0.009
50	63	0.222	0.3	0.037	100	93	0.042	0.2	0.092

Table A.10 Homogeneous Mystical bidders

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*
1	5	0.285	0.9	0.138
2	10	0.27	0.9	0.139
3	15	0.255	0.9	0.140
4	20	0.24	0.9	0.141
5	25	0.225	0.9	0.142
6	30	0.21	0.6	0.165
7	35	0.195	0.6	0.166
8	40	0.18	0.6	0.168
9	45	0.165	0.6	0.169
10	50	0.15	0.6	0.171
11	55	0.135	0.3	0.225
12	60	0.12	0.3	0.228
13	65	0.105	0.3	0.230
14	70	0.09	0.3	0.233
15	75	0.075	0.3	0.235
16	80	0.06	0.2	0.273
17	85	0.045	0.2	0.276
18	90	0.03	0.2	0.279
19	95	0.015	0.2	0.282
20	100	0	0.2	0.285

Table A.11 Homogeneous Sturdy bidders

Simulation Run	*Bargain Level*	*k*	*β*	*Expected Utility*
1	5	0.285	0.9	0.448
2	10	0.270	0.9	0.453
3	15	0.255	0.9	0.459
4	20	0.240	0.9	0.465
5	25	0.225	0.9	0.471
6	30	0.210	0.6	0.522
7	35	0.195	0.6	0.529
8	40	0.180	0.6	0.536
9	45	0.165	0.6	0.543
10	50	0.150	0.6	0.550
11	55	0.135	0.3	0.595
12	60	0.120	0.3	0.602
13	65	0.105	0.3	0.610
14	70	0.090	0.3	0.617
15	75	0.075	0.3	0.625
16	80	0.060	0.2	0.636
17	85	0.045	0.2	0.644
18	90	0.030	0.2	0.651
19	95	0.015	0.2	0.659
20	100	0.000	0.2	0.666

Table A.12 Homogeneous Strategic bidders

Simulation Run	***Bargain Level***	***k***	***ß***	***Expected Utility***
1	5	0.570	0.9	0.002
2	10	0.540	0.9	0.003
3	15	0.510	0.9	0.003
4	20	0.480	0.9	0.004
5	25	0.450	0.9	0.004
6	30	0.420	0.6	0.009
7	35	0.390	0.6	0.011
8	40	0.360	0.6	0.012
9	45	0.330	0.6	0.014
10	50	0.300	0.6	0.015
11	55	0.270	0.3	0.039
12	60	0.240	0.3	0.044
13	65	0.210	0.3	0.048
14	70	0.180	0.3	0.053
15	75	0.150	0.3	0.059
16	80	0.120	0.2	0.087
17	85	0.090	0.2	0.095
18	90	0.060	0.2	0.106
19	95	0.030	0.2	0.119
20	100	0.000	0.2	0.132

B. Samples of Java Code

```
//******************************************************************************************************************************
//                                   Project:     Design of Simulated Electronic Marketplace
//                                   Author:    Preetinder Kaur
//                                   Platform:  JADE/Java
//                                   Remarks: Software for electronic marketplace simulation.
//******************************************************************************************************************************


//----------------------------------------------------------------------------------------------------------------------------
//          Class:                   AuctionAgent
//          Note:                    Administor Agent
//----------------------------------------------------------------------------------------------------------------------------

package BiddingAgent;

import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.*;
import jade.wrapper.AgentContainer;
import jade.gui.GuiAgent;
import jade.gui.GuiEvent;
import jade.wrapper.AgentController;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;

public class AuctionAgent extends GuiAgent{
   private AuctionGui_22 myGui;
   private BidderGui myBidderGui;
   public AgentContainer myContainer;
   public static final int Event1 = 1000;  // Event fired on exit from bidder gui
   public static final int Event2 = 1001;  // Event fired on exit from bid history gui
   public static final int Event3 = 1002;  // Event fired on start bid button
   public static final int ST_START = 0;
   public static final int ST_SBID = 1;
   public static final int ST_WAIT = 2;
   public static final int ST_END = 3;
   public static final int ST_KILL = 4;
   public static final int Slots   = 3;
   public int slotsRemaining;
   boolean FlAgentsCreated = false;
   public int cntBidders = 0;

   String winner = "";
   Vector<BidderProperties> biddersList1 = new Vector<BidderProperties>();
   Vector<BidHistory> bidHistory1 = new Vector<BidHistory>();
   ArrayList<BidderBids> bidderBids = new ArrayList();
   ArrayList<String> myAmt = new ArrayList<>();
   AuctionProperties myAuctionProperties = new AuctionProperties();

   ACLMessage msgToBidders = new ACLMessage(ACLMessage.INFORM);
   public int state = 0;
```

```
    String currMaxBid = new String();
    String slotMaxBid = new String();
    public int cnt = 149;


    @Override
    protected void setup(){

        myContainer = getContainerController();
        myAuctionProperties.setClosingPrice(100);
        myAuctionProperties.setReservePrice(50);
        myAuctionProperties.setMinIncrement(10);
        myAuctionProperties.setSlots(Slots);
        myAuctionProperties.setCount(1);
        myBidderGui = new BidderGui(this);
        myGui = new AuctionGui_22(this);
        myGui.setVisible(true);
        System.out.println("-----------Trying New GUI agent--------"+getLocalName());
        state = 0;
        slotsRemaining = Slots;
    }

    private void GetCurrMaxBid() {

        if( bidHistory1.size() > 0){
            currMaxBid = String.valueOf(bidHistory1.lastElement().getBidAmount());
            slotMaxBid = currMaxBid;
        }
    }

class MyBehaviour extends CyclicBehaviour{

        @Override
        public void action() {

            switch(state){

                case ST_START:
                    // some dummy delay
                    System.out.println("\nCounter:"+cnt);
                    state = 1;
                    break;
                case ST_SBID:
                    clearPlaceBids();
                    msgToBidders.setContent("SBID"+","+currMaxBid);
                    send(msgToBidders);
                    state = 2;
                    break;
                case ST_WAIT:
                    ACLMessage msg = receive();
                    if (msg!=null){
                        if( checkBidder(msg.getSender().getLocalName())){   // Check if the bidder isin the list
                            state = processMsg(msg);    //Returns the state to go to
                            break;
                        }
                    }
                    block();
                    break;
                case ST_END:
                    for(int i=0; i<bidHistory1.size(); i++)
                        System.out.println("bidHistory  elements" +bidHistory1.get(i).getBidAmount());
                    System.out.println("!!! WINNER IS " + winner + " , " + "Bid Amount: "+ currMaxBid );
                    try {
                        genFinalReport();
                        genWinnersReport();
```

```
            }
            catch (IOException ex) {
            }
            msgToBidders.setContent("RESET"+","+currMaxBid);
            send(msgToBidders);

            if(cnt > 0){
               cnt--;
            }

            if( cnt > 0){
               ClearBidderBids();
               initAuction();
               state = ST_START;
            }
            else{
               state = ST_WAIT;
            }

            break;
      }
   }

   private int processMsg(ACLMessage msg) {

      int stateRet = ST_WAIT;
      String content = msg.getContent();
      String name = msg.getSender().getLocalName();
      ArrayList<String> values = new ArrayList<>();
      DecodeMsg(content, values);

      switch( values.get(0)){         // Get the type of the message
         case "NBID":                 //Send bid message
            String amt = values.get(1);
            String Const_k = values.get(3);
            String Const_beta = values.get(5);
            String Const_blvl = values.get(7);

            if( processBid(amt) ){
               winner = name;
            }
            processBidder(name, amt, Const_k, Const_beta, Const_blvl);

            if( recFromAll() ){
               System.out.println(" ");
               slotsRemaining--;
               if( Float.parseFloat(slotMaxBid) > Float.parseFloat(currMaxBid) ){
                  currMaxBid = slotMaxBid;
               }
               if( slotsRemaining <= 0 ){
                 stateRet =  ST_END;
               }
               else{
                  stateRet = ST_SBID;
               }
            }
            else{
               stateRet = ST_WAIT;
            }
            break;

         default:
            break;
      }
      return stateRet;
```

```
    }

    // Decode the message and put the comma separated values in the array
    private void DecodeMsg(String content, ArrayList<String> values) {
        String[] paras = content.split(",");
        values.addAll(Arrays.asList(paras));
    }

    // Checks if the bidder is valid
    private boolean checkBidder(String nm) {
        boolean ret = false;
        for( int i=0; i < bidderBids.size(); i++ ){
            if( nm.equals(bidderBids.get(i).getName())){
                ret = true;
                break;
            }
        }
        return ret;
    }

    // Check if the bid received is a valid bid
    // If valid then set it as the currMaxBid
    private boolean processBid(String amt) {
        boolean ret = false;
        float value = Float.valueOf(amt);
        float curr = Float.valueOf(currMaxBid);
        float slot = Float.valueOf(slotMaxBid);

        if( value >= myAuctionProperties.getReservePrice() && value <=myAuctionProperties.getClosingPrice()){
            if( value >= (myAuctionProperties.getMinIncrement() + curr) ){
                if( value > slot){
                    slot = value;
                    slotMaxBid = String.valueOf(slot);
                    ret = true;
                }
            }
            else {System.out.println("bid ignored is:"+amt);
                myAmt.add(amt);
            }
        }
        return ret;
    }

    private void clearPlaceBids(){
        for(int i=0; i<bidderBids.size(); i++){
            bidderBids.get(i).setBidPlaced(false);
        }
    }

    private void processBidder(String name, String amt, String Const_k, String Const_beta, String Const_blvl) {
        for( int i=0; i<bidderBids.size(); i++){
            BidderBids temp;
            temp = bidderBids.get(i);

            if( temp.getName().equals(name)){
                if( temp.getBidPlaced() == false ){
                    temp.setBidPlaced(true);
                    temp.addBid(amt);
                    temp.setK(Const_k);
                    temp.setBeta(Const_beta);
                    biddersList1.get(i).setBargainLevel(Const_blvl);
                    System.out.println("TestBidder: " + name + "," + amt + " , " +"Const_K:" + Const_k + " , " + "Const_Beta:
" + Const_beta +" , " +"Const_bLvl" + Const_blvl);
                }
```

```
            }
        }
    }

    // Checks if all the listed bidder have placed the bids for a slot
    private boolean recFromAll() {
        boolean ret = true;
        for( int i=0; i < bidderBids.size(); i++){
            if( bidderBids.get(i).getBidPlaced() == false ){
                ret = false;
                break;
            }
        }
        return ret;
    }

    private void genFinalReport() throws IOException {
            String auctionId = String.valueOf(myAuctionProperties.getId());
            String fileName = auctionId + ".csv";
            String fileName1 = auctionId + "_winner" +".csv";

        try (
            BufferedWriter writer = new BufferedWriter(new FileWriter(fileName, true))) {
            String header = "Auction Id,Bidder Name,Bid Amount,Slot,Est Price,";
            header += "Reserve Price,Min Increment,Type,Preference,Behaviour,Attitude,Lvl,Const K,Beta";
            writer.write(header);
            writer.newLine();

            for(int i=0; i<bidderBids.size(); i++){
                for(int j=0; j< bidderBids.get(i).getBidsSz(); j++){
                    String result = String.valueOf(myAuctionProperties.getId());
                    result += "," + bidderBids.get(i).getName();
                    result += "," + bidderBids.get(i).getBid(j);
                    result += "," + String.valueOf(myAuctionProperties.getSlots());
                    result += "," + String.valueOf(myAuctionProperties.getClosingPrice());
                    result += "," + String.valueOf(myAuctionProperties.getReservePrice());
                    result += "," + String.valueOf(myAuctionProperties.getMinIncrement());
                    result += "," + biddersList1.get(i).getType();
                    result += "," + biddersList1.get(i).getPreference();
                    result += "," + biddersList1.get(i).getBehaviour();
                    result += "," + biddersList1.get(i).getAttitude();
                    result += "," + biddersList1.get(i).getBargainLevel();
                    result += "," + bidderBids.get(i).getK();
                    result += "," + bidderBids.get(i).getBeta();
                    writer.write(result);
                    writer.newLine();
                    block(1000);
                }
            }

            writer.newLine();
            String Winner = "Winner is:";
            Winner += winner;
            writer.write(Winner);
            writer.newLine();
            String winningBid = "Winning Bid is:";
            winningBid += currMaxBid;
            writer.write(winningBid);
            writer.newLine();
            String BidsIgnored = "Ignored bids are:";

            for( int i=0; i<myAmt.size();i++)
            {
                BidsIgnored += myAmt.get(i);
                BidsIgnored += "  ";
```

```
            }
            writer.write(BidsIgnored);
            writer.close();

         }
      }

      private void genWinnersReport() throws IOException {
         String auctionId = String.valueOf(myAuctionProperties.getId());
         String fileName1 = auctionId + "_winner" +".csv";
         try (
            BufferedWriter writer1 = new BufferedWriter(new FileWriter(fileName1, true))) {
            String Winner = "Winner is:";
            Winner += winner;
            writer1.write(Winner);
            String winningBid = "      and the Winning Bid is:";
            winningBid += currMaxBid;
            writer1.write(winningBid);
            writer1.newLine();
         }
      }

      private void initAuction() {
         GetCurrMaxBid();
         slotsRemaining = Slots;
      }

      private void ClearBidderBids() {
         for(int i=0; i< bidderBids.size(); i++){
            bidderBids.get(i).clearBids();
         }
         clearPlaceBids();
      }
   }// Cyclic behaviour

   @Override
   protected void onGuiEvent(GuiEvent ev)
   {
      switch(ev.getType()){
        case Event1:
            biddersList1 =(Vector<BidderProperties>) ev.getParameter(0);
            cntBidders = biddersList1.size();
            Send();
            break;
        case Event2:
            bidHistory1 = (Vector<BidHistory>) ev.getParameter(0);
            Send1();
            break;
        case Event3:
            myAuctionProperties = (AuctionProperties)ev.getParameter(0);
            cnt = myAuctionProperties.getCount();
            CreateAgents();
            GetCurrMaxBid();
            state = 0;
            addBehaviour(new MyBehaviour());
            break;
      }
   }

   private void Send() {
      System.out.println("The participating bidders are:");
      for(int i=0; i<biddersList1.size(); i++){
       System.out.println(biddersList1.get(i).getName()+ " " +biddersList1.get(i).getType()+" "
                  +biddersList1.get(i).getPreference()+ " "+biddersList1.get(i).getBehaviour()+ " "
                     +biddersList1.get(i).getAttitude()+" "+biddersList1.get(i).getBargainLevel());
```

```
        }
        }

    private void Send1() {
        System.out.println("The participating bidders are:");
        for(int i=0; i<bidHistory1.size(); i++){
         System.out.println(""+bidHistory1.get(i).getBidderName()+" "+bidHistory1.get(i).getBidAmount()+"
"+bidHistory1.get(i).getTime());
        }
    }

    public void CreateAgent(int i){
        try {

            if( i > biddersList1.size())
              return;
            Object[] args = new Object[4];
            args[0] = (Object)biddersList1.get(i);      // BidderAgent
            args[1] = (Object)bidHistory1;             // Bid history
            args[2] = myAuctionProperties;             // Auction properties
            args[3] = this.getLocalName();             // Name of aution agent
            AgentController a = myContainer.createNewAgent(biddersList1.get(i).getName(), "BiddingAgent.BidderAgent",
args );

            a.start();

            msgToBidders.addReceiver(new AID(biddersList1.get(i).getName(), AID.ISLOCALNAME));
            BidderBids temp = new BidderBids();
            temp.setName(biddersList1.get(i).getName());
            bidderBids.add(temp);
          }
        catch (Exception e){
          System.out.println("Exception:" + e.getMessage());
        }
     }

    private void CreateAgents() {
        for(int i=0; i<biddersList1.size();i++ ){
           System.out.println("-----------------while creating bidder Agents--------------"+biddersList1.get(i).getName());
          CreateAgent(i);
        }
        FlAgentsCreated = true;
    }
}

//---------------------------------------------------------------------------------------------------------------------------
//          Class:                  BidderAgent
//          Note:                   ADBA Bidder Agent
//---------------------------------------------------------------------------------------------------------------------------

package BiddingAgent;
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Random;
import java.util.Vector;

public class BidderAgent extends Agent{
    // Values inherited from the aution agent
    BidderProperties    myProperties = new BidderProperties();
    Vector<BidHistory>  myHistory = new Vector<BidHistory>();
    int Slots;      // No. of times the bid is to be sent
    int slotsRemaining;
```

```
AuctionProperties   myAuctionProperties = new AuctionProperties();
BidderConstants     myConstants = new BidderConstants();
String myAuctionAgent = "";         // Name of the aution agent
String predictedClosingPrice = "0"; // Predicted closing price of aution
ArrayList<String>   maxBids = new ArrayList();     // Previous max bids sent by aution
ArrayList<String>   myBids = new ArrayList();      // Bids send by the agent
ArrayList<String>   fConsts = new ArrayList();      // Last three max bids for calcilating F's
Random generator = new Random();
 int bLvl = 0;

 @Override
protected void setup()
{
   Object[] myArgs = getArguments();
   maxBids.clear();       // Clear the list at startup
   myBids.clear();        // Clear the list at startup
   // Get the arguments
  if( myArgs != null){
      if( myArgs.length == 4){
         myProperties = (BidderProperties)myArgs[0];     // First argument is bidder properties
         myHistory = (Vector<BidHistory>)( myArgs[1]);  // Second argument is bid history
         myAuctionProperties =  (AuctionProperties)( myArgs[2]);
         myAuctionAgent = (String)myArgs[3];            // Name of the aution agent

         Slots = myAuctionProperties.getSlots();        // No. of time slots
         slotsRemaining = Slots;                        // Slots counter

         System.out.println( "My Test Agent Name: " + myProperties.getName());
         System.out.println( "Size of history: " + myHistory.size());
         String pref = myProperties.getPreference();
         String att = myProperties.getAttitude();

         CalBidderConstants(pref,att);
         for(int i=0; i< myHistory.size(); i++){
            float bid = myHistory.get(i).getBidAmount();
            fConsts.add(String.valueOf(bid));
         }
      }
   }
   System.out.println( "-------before bidder's action---------");

  // Create the behaviour of the agent
   addBehaviour(new CyclicBehaviour(this){
      @Override
      public void action(){
         System.out.println( "-------inside bidder's action---------");
         ACLMessage msg = receive();
         if (msg!=null) {                // if message received
            if( msg.getSender().getLocalName().equals(myAuctionAgent) ){    // if the sender is my aution agent
               String content = msg.getContent();
               ACLMessage reply = msg.createReply();
               ProcessMsg(content, reply);
               System.out.println( "------------------testing1111111111----------");
            }
         }// msg != null
                    block();
      }//action()

      // Process the messages recived from the aution agent
      private void ProcessMsg(String content, ACLMessage reply) {
         ArrayList<String> values = new ArrayList<>();
         DecodeMsg(content, values);
         switch( values.get(0)){        // Get the type of the message
            case "SBID":             // Send bid message
               if( slotsRemaining > 0){
```

```
                slotsRemaining--;
            }
            System.out.println("SLOT NO:--->>> " + (Slots-slotsRemaining) );
            System.out.println();
            SendNewBid(values.get(1), reply);
            break;
        case "TEST":
            System.out.println("Received Test Message:" + values.get(0) + " , " + values.get(1));
            break;
        case "RESET":
            reset();
            initBidders();
            System.out.println("---------deleting bidder Agents--------"+myProperties.getName());
            break;
        case "EXIT":
            break;
    }
}

// Decode the message and put the comma separated values in the array
private void DecodeMsg(String content, ArrayList<String> values) {
    String[] paras = content.split(",");
    values.addAll(Arrays.asList(paras));
}

private void SendNewBid(String currBid, ACLMessage reply) {
    String myBid;   // = new String();
    int sz = fConsts.size();
    // Add the currmax to the table if it is not same as previous
    if( currBid.equals(fConsts.get(sz-1)) ){
        // do nothing
    }
    else{
        fConsts.add(currBid);
    }
    if( myProperties.getPreference().equals("Single Bidding") ){
        myBid = GetSingleBid(currBid);      // Single behaviour agent
    }
    else{
        myBid = GetMultipleBid(currBid);    // Multiple behaviour agent
    }
    maxBids.add(currBid);
    myBids.add(myBid);              // Bids sent by the myself
    SendNewBidReply(myBid, reply);
 }

private void SendNewBidReply(String myBid, ACLMessage reply) {
    reply.setPerformative( ACLMessage.INFORM );
    reply.setContent("NBID"+"," + myBid +"," +
            "Const_K" + "," + String.valueOf(myConstants.getConstK()) + "," +
            "Const_beta" + "," + String.valueOf(myConstants.getConstbeta()) + "," +
            "Const_blvl" + "," + String.valueOf(myConstants.getConstBlvl()) );
    send(reply);
}

private String GetSingleBid(String currBid) {
    String bid = "0";
    switch (myProperties.getBehaviour()) {      // Mystical bidder
        case "Mystical":
            if( slotsRemaining == 0){          // Bids on the last slot
                bid = CalMysticalBid(currBid);
            }
            // else the bid is 0
            break;
        case "Sturdy":                    // Sturdy bidder
```

```
            if( slotsRemaining == Slots-1 ){    // Bids only on first slot
               bid = CalSturdyBid(currBid);
            }
            else{                               // Repeat the previous bid
               if( myBids.size() > 0)
                  bid = myBids.get(0);
            }
            break;
         default:
            break;
      }
      return bid;
   }

   // Calculate multiple bid amount
   private String GetMultipleBid(String currBid) {
      float bid = Float.parseFloat(currBid);

      if( myProperties.getType().equals("Novice")){
         bid += myAuctionProperties.getMinIncrement();
      }
      else{
         String type = myProperties.getType();
         bid = formulaBidAmount(currBid, "Multiple", type);
      }
         return( String.valueOf(bid));
   }

   // Cal sophisticated bid amount
   private String CalSophisticatedBid(String currBid) {

      return( CalSingleBid(currBid));
   }

   // Cal Ambitiuos bid amount
   private String CalAmbitiousBid(String currBid) {
      return( CalSingleBid(currBid));
   }

   private String CalSingleBid(String currBid) {

      float bidAmount;
      String type = myProperties.getType();
      bidAmount = formulaBidAmount(currBid, "Single", type);

      return( String.valueOf(bidAmount));
   }

   private float formulaBidAmount(String currBid, String pref, String type) {
      String bType = type;
      float k = myConstants.getConstK();
      float beta = myConstants.getConstbeta();
      float currMax = Float.parseFloat(currBid);
      final int t = Slots-slotsRemaining;
      final int t_max = Slots+1;
      float   alpha_t;
      alpha_t = (float)(k+(1.0-k)*(float)Math.pow(((float)Math.min(t,t_max)/(float)t_max),(float)(1/beta)));
       //reference:ADMI paper
      System.out.println("alpha_t" +alpha_t );
      float min_b = currMax;
                     float max_b = myAuctionProperties.getClosingPrice();

      //Calculate F_t
      float F_t;
                     F_t = (float)(min_b+alpha_t*(max_b-min_b)); //Bid amount to be sent
```

```
                    //Calculate F_c_t
                    float F_ct;
        float F_x = currMax;
                    float F_z = 1;
        float F_y = 0;
       int sz = fConsts.size();
       if( sz >= 3){
          F_x = Float.valueOf(fConsts.get(sz-1));
          F_y = Float.valueOf(fConsts.get(sz-2));
          F_z = Float.valueOf(fConsts.get(sz-3));
       }
       else if( sz == 2){   // should never happen
          F_y = Float.valueOf(fConsts.get(sz-1));
          F_z = Float.valueOf(fConsts.get(sz-2));
       }
       else if( sz == 1){   // should never happen
          F_z = Float.valueOf(fConsts.get(sz-2));
       }

                    F_ct= (float)(Math.min(Math.max((F_x * F_y)/F_z , min_b) , max_b));
        System.out.println("F_ct:------" +F_ct +"f_t:-----------" +F_t +"Alpha:----" +alpha_t);
                    float F_c= (float)(F_t + F_ct)/2;
        System.out.println(myProperties.getName() + ":" + " F_x:" + F_x + " F_y:" + F_y + "F_z:" + F_z );
        if("Moderate".equals(bType))
           return F_t;
        else
           return F_c;
    }

    private String CalMysticalBid(String currBid) {
        return( CalSingleBid(currBid));
    }

    private String CalSturdyBid(String currBid) {
        return( CalSingleBid(currBid));
    }

    private void initBidders() {
        slotsRemaining = Slots;
        String pref = myProperties.getPreference();
        String att = myProperties.getAttitude();
        CalBidderConstants(pref,att);
        maxBids.clear();
        myBids.clear();
        fConsts.clear();
        for(int i=0; i< myHistory.size(); i++){
           float bid = myHistory.get(i).getBidAmount();
           fConsts.add(String.valueOf(bid));
        }
    }
       }); // addBhaviour
}

private void CalBidderConstants(String pref, String att) {
    switch (pref) {
       case "Single Bidding":
          switch (att) {
             case "Sophisticated":
                {
                float [] betaArray = {0.2f, 0.3f, 0.6f, 0.9f};
                String bLvls = myProperties.getBargainLevel();
                int bLvl = generator.nextInt(100) + 1;
                float k =((float)((float)100.0 - (float)bLvl)/100)*(float)0.3;
                int idx = (100 - bLvl)/25;
                float beta = betaArray[idx];
```

```
                System.out.println("****************:" +k +":****:"+beta +"******" +bLvl);
                myConstants.setConstK(k);
                myConstants.setConstbeta(beta);
                myConstants.setConstBlvl(bLvl);
                break;
                }
            case "Ambitious":
                {
                float [] betaArray = {10.00f, 20.00f, 30.00f, 40.00f};
                String bLvls = myProperties.getBargainLevel();
                int bLvl = generator.nextInt(100) + 1;
                float k =(((float)((float)100.0 - (float)bLvl)/100)*(float)0.4)+(float)0.6;
                int idx = (100 - bLvl)/25;
                float beta = betaArray[idx];
                myConstants.setConstK(k);
                myConstants.setConstbeta(beta);
                myConstants.setConstBlvl(bLvl);
                break;
                }
          }
          break;
       case "Multiple Bidding":
    switch (att) {
       case "Sophisticated":
          {
             float [] betaArray = {0.2f, 0.3f, 0.6f, 0.9f};
             String bLvls = myProperties.getBargainLevel();
             int bLvl = generator.nextInt(100) + 1;
             float k =((float)((float)100.0 - (float)bLvl)/100)*(float)0.6;
             int idx = (100 - bLvl)/25;
             float beta = betaArray[idx];
             myConstants.setConstK(k);
             myConstants.setConstbeta(beta);
             myConstants.setConstBlvl(bLvl);
             break;
          }
       case "Ambitious":
          {
             float [] betaArray = {10.00f, 20.00f, 30.00f, 40.00f};
             String bLvls = myProperties.getBargainLevel();
             int bLvl = generator.nextInt(100) + 1;
             float k =(((float)((float)100.0 - (float)bLvl)/100)*(float)0.4)+(float)0.6;
             int idx = (100 - bLvl)/25;
             float beta = betaArray[idx];
             myConstants.setConstK(k);
             myConstants.setConstbeta(beta);
             myConstants.setConstBlvl(bLvl);
             break;
          }
    }
 break;
    }
  }
}

//-------------------------------------------------------------------------------------------------------------------------
//          Class:                  AuctionGui_22
//          Note:                   Graphical user interface for the simulated electronic marketplace
//-------------------------------------------------------------------------------------------------------------------------

package BiddingAgent;
import jade.gui.GuiEvent;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
```

```
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;
import javax.swing.Icon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.*;

public class AuctionGui_22 extends javax.swing.JFrame implements ActionListener{
   private AuctionAgent myAgent;
   private BidHistoryGui1 myBidHistoryGui1;
   private BidderGui myBidderGui;
   private BidderProperties myBidderProperties;
   Vector<BidderProperties> biddersList = new Vector<BidderProperties>();
   Vector<BidHistory> bidHistory = new Vector<BidHistory>();
   DefaultListModel listModel = new DefaultListModel();
   DefaultTableModel tableModel = new DefaultTableModel();
   AuctionProperties auctionProperties = new AuctionProperties();
   private Icon Icon;

     // Creates new form AuctionGui_22
      public AuctionGui_22(AuctionAgent a1) {
      super();
      myAgent = a1;
      initComponents();
      tableModel = (DefaultTableModel)tblBidHistory.getModel();
   }

     public AuctionGui_22(BidderGui b1) {
      super();
      myBidderGui = b1;
      initComponents();
   }

     public AuctionGui_22(BidHistoryGui1 bh1) {
      super();
      myBidHistoryGui1 = bh1;
      initComponents();
   }

   private AuctionGui_22() {
      throw new UnsupportedOperationException("Not yet implemented");
   }

   @SuppressWarnings("unchecked")
    //Automated code not ahown here
   private void btnAddBiddersActionPerformed(java.awt.event.ActionEvent evt) {

            BidderGui dialog = new BidderGui(this, true, myAgent);
```

```
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
               @Override
               public void windowClosing(java.awt.event.WindowEvent e) {
               }
            });
            int rows = biddersList.size();
            BidderProperties temp = new BidderProperties();
            DefaultTableModel model = dialog.GetTableModel();
            model.getDataVector().clear();

            for(int i=0; i<rows;i++){
               temp = biddersList.get(i);
               String name = temp.getName();
               String type = temp.getType();
               String preference = temp.getPreference();
               String behaviour = temp.getBehaviour();
               String attitude = temp.getAttitude();
               String desire = String.valueOf(temp.getBargainLevel());

               model.addRow(new Object[] {name, type, preference, behaviour, attitude, desire});

            }
            dialog.setVisible(true);
    }

    private void btnStartBiddingActionPerformed(java.awt.event.ActionEvent evt) {
       auctionProperties.setId(txtAuctionId.getText());
       auctionProperties.setMinIncrement( Float.valueOf(txtMinimumIncrement.getText()));
       auctionProperties.setReservePrice(Float.valueOf(txtReservePrice.getText()));
       auctionProperties.setClosingPrice(Float.valueOf(ftxtEstClosingPrice.getText()));
       auctionProperties.setSlots(3);
       auctionProperties.setCount(Integer.valueOf(txtCounter.getText()));
       if(evt.getActionCommand().equals("Start Bidding")){
          GuiEvent ev = new GuiEvent(null, AuctionAgent.Event3);
          ev.addParameter(auctionProperties);
          myAgent.postGuiEvent(ev);
       }
    }

    private void txtTimeLeftActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void txtConditionActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void txtAuctionIdActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void btnSetBidHistoryActionPerformed(java.awt.event.ActionEvent evt) {

        BidHistoryGui1 dialog = new BidHistoryGui1(this, true, myAgent);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
             @Override
               public void windowClosing(java.awt.event.WindowEvent e) {
               }
            });
            int rows = tableModel.getRowCount();
            javax.swing.table.DefaultTableModel model = dialog.GetTableModel();
              for (int i = 0; i<rows; i++) {
            String name = (String) tableModel.getValueAt(i,0);
            String amt = (String)(tableModel.getValueAt(i,1));
            String time = ((String)tableModel.getValueAt(i,2));
            model.addRow(new Object[]{name, amt, time});
         }
           dialog.setVisible(true);
```

```
}

private void menuSaveActionPerformed(java.awt.event.ActionEvent evt) {
}

private void manuSaveActionPerformed(java.awt.event.ActionEvent evt) {
        try {

                DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
                DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

                // Root elements
                Document doc = docBuilder.newDocument();
                Element rootElement = doc.createElement("auction");
                doc.appendChild(rootElement);

                // Staff elements
                Element properties = doc.createElement("Properties");
                rootElement.appendChild(properties);

                // Set attribute to staff element
                Attr id = doc.createAttribute("id");
                id.setValue(txtAuctionId.getText());
                properties.setAttributeNode(id);

                Attr rPrice = doc.createAttribute("r_price");
                rPrice.setValue(txtReservePrice.getText());
                properties.setAttributeNode(rPrice);

                 Attr mInc = doc.createAttribute("m_inc");
                mInc.setValue(txtMinimumIncrement.getText());
                properties.setAttributeNode(mInc);

                 Attr cPrice = doc.createAttribute("c_price");
                cPrice.setValue(ftxtEstClosingPrice.getText());
                properties.setAttributeNode(cPrice);

                 Attr endTime = doc.createAttribute("e_time");
                endTime.setValue(txtTimeLeft.getText());
                properties.setAttributeNode(endTime);

                  Attr currHigherBidder = doc.createAttribute("highest_bidder");
                currHigherBidder.setValue(ftxtCurrHigherBidder.getText());
                properties.setAttributeNode(endTime);

                Attr currBid = doc.createAttribute("curr_Bid");
                currBid.setValue(ftxtCurrMaxBid.getText());
                properties.setAttributeNode(currBid);

                Element item_description = doc.createElement("Item_description");
                rootElement.appendChild(item_description);

                 Attr name = doc.createAttribute("name");
                name.setValue(lblName.getText());
                item_description.setAttributeNode(name);

                Attr condition = doc.createAttribute("con");
                condition.setValue(txtCondition.getText());
                item_description.setAttributeNode(condition);

                Attr location = doc.createAttribute("loc");
                location.setValue(ftxtLocation.getText());
                item_description.setAttributeNode(location);

                Attr postage = doc.createAttribute("post");
```

```
                postage.setValue(ftxtPostage.getText());
                item_description.setAttributeNode(postage);

                 Attr sName = doc.createAttribute("s_name");
                sName.setValue(ftxtSeller.getText());
                item_description.setAttributeNode(sName);

                Attr sRating = doc.createAttribute("s_rating");
                sRating.setValue(ftxtSellerRating.getText());
                item_description.setAttributeNode(sRating);

                Element history = doc.createElement("history");
                rootElement.appendChild(history);

for(int i=0; i<bidHistory.size(); i++){

   Element BH = doc.createElement("BH");
   history.appendChild(BH);

   Attr hName = doc.createAttribute("hname");
   hName.setValue(bidHistory.get(i).getBidderName());
   BH.setAttributeNode(hName);

   Attr amt = doc.createAttribute("amt");
   amt.setValue(String.valueOf(bidHistory.get(i).getBidAmount()));
   BH.setAttributeNode(amt);

   Attr time = doc.createAttribute("time");
   time.setValue(String.valueOf(bidHistory.get(i).getTime()));
   BH.setAttributeNode(time);
}

Element bidders = doc.createElement("bidders");
                rootElement.appendChild(bidders);

for(int i=0; i<biddersList.size(); i++){

   Element BD = doc.createElement("BD");
   bidders.appendChild(BD);

   Attr bName = doc.createAttribute("name");
   bName.setValue(biddersList.get(i).getName());
   BD.setAttributeNode(bName);

   Attr type = doc.createAttribute("type");
   type.setValue(biddersList.get(i).getType());
   BD.setAttributeNode(type);

   Attr pref = doc.createAttribute("pref");
   pref.setValue(biddersList.get(i).getPreference());
   BD.setAttributeNode(pref);

   Attr behv = doc.createAttribute("behv");
   behv.setValue(biddersList.get(i).getBehaviour());
   BD.setAttributeNode(behv);

   Attr att = doc.createAttribute("att");
   att.setValue(biddersList.get(i).getAttitude());
   BD.setAttributeNode(att);

   Attr bLvl = doc.createAttribute("bLvl");
   bLvl.setValue(String.valueOf(biddersList.get(i).getBargainLevel()));
   BD.setAttributeNode(bLvl);
}
```

```
                JFileChooser chooser = new JFileChooser();
        File infile = null;
        if (chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION)
        {
           infile = chooser.getSelectedFile();
        }
        // Write the content into xml file
                TransformerFactory transformerFactory = TransformerFactory.newInstance();
                Transformer transformer = transformerFactory.newTransformer();
                DOMSource source = new DOMSource(doc);
                StreamResult result = new StreamResult(infile);
                transformer.transform(source, result);
        System.out.println("File saved!");
           } catch (ParserConfigurationException pce) {
                pce.printStackTrace();
           } catch (TransformerException tfe) {
                tfe.printStackTrace();
          }
}

private void menuLoadActionPerformed(java.awt.event.ActionEvent evt) {
  try {
        JFileChooser chooser = new JFileChooser();
        File infile = null;
        FileReader reader = null;
        if (chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION)
        {
           infile = chooser.getSelectedFile();
           reader = new FileReader(infile);
        }
                DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
                DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
                Document doc = dBuilder.parse(infile);
                doc.getDocumentElement().normalize();
                System.out.println("Root element :" + doc.getDocumentElement().getNodeName());

        NodeList properties = doc.getElementsByTagName("Properties");
        System.out.println("-----------------------");
        System.out.println( properties.item(0).getAttributes().getNamedItem("id"));
        System.out.println( properties.item(0).getAttributes().getNamedItem("r_price"));
        System.out.println( properties.item(0).getAttributes().getNamedItem("m_inc"));
        System.out.println( properties.item(0).getAttributes().getNamedItem("c_price"));

        txtAuctionId.setText(properties.item(0).getAttributes().getNamedItem("id").getNodeValue());
        txtReservePrice.setText(properties.item(0).getAttributes().getNamedItem("r_price").getNodeValue());
        txtMinimumIncrement.setText(properties.item(0).getAttributes().getNamedItem("m_inc").getNodeValue());
        ftxtEstClosingPrice.setText(properties.item(0).getAttributes().getNamedItem("c_price").getNodeValue());
        txtTimeLeft.setText(properties.item(0).getAttributes().getNamedItem("e_time").getNodeValue());

        //Populate auctionProperties
        String id =properties.item(0).getAttributes().getNamedItem("id").getNodeValue();
        String rPrice =properties.item(0).getAttributes().getNamedItem("r_price").getNodeValue();
        String mInc =properties.item(0).getAttributes().getNamedItem("m_inc").getNodeValue();
        String cPrice =properties.item(0).getAttributes().getNamedItem("c_price").getNodeValue();

        auctionProperties.setId(id);
        auctionProperties.setReservePrice(Float.parseFloat(rPrice));
        auctionProperties.setMinIncrement(Float.parseFloat(mInc));
        auctionProperties.setClosingPrice(Float.parseFloat(cPrice));

        NodeList ItmDesc = doc.getElementsByTagName("Item_description");

        lblName.setText(ItmDesc.item(0).getAttributes().getNamedItem("name").getNodeValue());
        ftxtLocation.setText(ItmDesc.item(0).getAttributes().getNamedItem("loc").getNodeValue());
        txtCondition.setText(ItmDesc.item(0).getAttributes().getNamedItem("con").getNodeValue());
```

```
            ftxtPostage.setText(ItmDesc.item(0).getAttributes().getNamedItem("post").getNodeValue());
            ftxtSeller.setText(ItmDesc.item(0).getAttributes().getNamedItem("s_name").getNodeValue());
            ftxtSellerRating.setText(ItmDesc.item(0).getAttributes().getNamedItem("s_rating").getNodeValue());

            NodeList history = doc.getElementsByTagName("history");
            NodeList bh = history.item(0).getChildNodes();

            tableModel.getDataVector().clear();
            for(int i=0;i<bh.getLength(); i++){
              BidHistory temp = new BidHistory();
              String hname = bh.item(i).getAttributes().getNamedItem("hname").getNodeValue();
              String amt = bh.item(i).getAttributes().getNamedItem("amt").getNodeValue();
              String time = bh.item(i).getAttributes().getNamedItem("time").getNodeValue();
              temp.setBidderName(hname);
              temp.setBidAmount(Float.parseFloat(amt));
              temp.setBidTime(Float.parseFloat(time));
              bidHistory.add(temp);
              tableModel.addRow(new Object[]{hname, amt, time});
            }
            currMaxBidSetText();

            GuiEvent ev = new GuiEvent(null, AuctionAgent.Event2);
            ev.addParameter(bidHistory);
            myAgent.postGuiEvent(ev);
            NodeList bidders = doc.getElementsByTagName("bidders");
            NodeList bd = bidders.item(0).getChildNodes();
            listModel.clear();
            biddersList.clear();

            for(int i=0;i<bd.getLength(); i++){
              BidderProperties temp = new BidderProperties();
              String name = bd.item(i).getAttributes().getNamedItem("name").getNodeValue();
              String att = bd.item(i).getAttributes().getNamedItem("att").getNodeValue();
              String bLvl = bd.item(i).getAttributes().getNamedItem("bLvl").getNodeValue();
              String pref = bd.item(i).getAttributes().getNamedItem("pref").getNodeValue();
              String type = bd.item(i).getAttributes().getNamedItem("type").getNodeValue();
              String behv = bd.item(i).getAttributes().getNamedItem("behv").getNodeValue();
              //populate the bidder properties
              temp.setName(name);
              temp.setAttitude(att);
              temp.setBargainLevel(bLvl);
              temp.setPreference(pref);
              temp.setType(type);
              temp.setBehaviour(behv);

              biddersList.add(temp);
              listModel.addElement(name);
            }
             currHigherBidderSetText();

              GuiEvent ev1 = new GuiEvent(null, AuctionAgent.Event1);
              ev1.addParameter(biddersList);
              myAgent.postGuiEvent(ev1);
              } catch (Exception e) {
                      e.printStackTrace();
        }
    }

    private void ftxtLocationActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void ftxtSellerActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void txtReservePriceActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}

private void txtMinimumIncrementActionPerformed(java.awt.event.ActionEvent evt) {
}

private void ftxtPostageActionPerformed(java.awt.event.ActionEvent evt) {
}

private void txtCounterActionPerformed(java.awt.event.ActionEvent evt) {
}
// Variables declaration - do not modify
private javax.swing.JButton btnAddBidders;
private javax.swing.JButton btnSetBidHistory;
private javax.swing.JButton btnStartBidding;
private javax.swing.JFileChooser db;
private javax.swing.JFormattedTextField ftxtCurrHigherBidder;
private javax.swing.JFormattedTextField ftxtCurrMaxBid;
private javax.swing.JFormattedTextField ftxtEstClosingPrice;
private javax.swing.JFormattedTextField ftxtLocation;
private javax.swing.JFormattedTextField ftxtPostage;
private javax.swing.JFormattedTextField ftxtSeller;
private javax.swing.JFormattedTextField ftxtSellerRating;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel9;
private javax.swing.JList jList1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem6;
private javax.swing.JMenuItem jMenuItem7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel6;
private javax.swing.JPanel jPanel7;
private javax.swing.JRadioButtonMenuItem jRadioButtonMenuItem1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JLabel lblImage;
private javax.swing.JLabel lblName;
private javax.swing.JMenuItem manuSave;
private javax.swing.JMenuItem menuLoad;
private javax.swing.JMenu menuSave;
private javax.swing.JTable tblBidHistory;
private javax.swing.JFormattedTextField txtAuctionId;
private javax.swing.JFormattedTextField txtCondition;
private javax.swing.JFormattedTextField txtCounter;
```

```
    private javax.swing.JFormattedTextField txtMinimumIncrement;
    private javax.swing.JFormattedTextField txtReservePrice;
    private javax.swing.JTextField txtTimeLeft;
    // End of variables declaration


    void addBidderList(Vector<BidderProperties> bidders) {
        biddersList = bidders;
    }

    void addbidsHistory(Vector<BidHistory> bidsHistory) {
       bidHistory = bidsHistory;
    }

    private String getTagValue(String sTag, Element eElement) {
        NodeList nlList = eElement.getElementsByTagName(sTag).item(0).getChildNodes();
        Node nValue = (Node) nlList.item(0);
                return nValue.getNodeValue();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
    }

    void currMaxBidSetText() {
        float currMaxBid = bidHistory.lastElement().getBidAmount();
        ftxtCurrMaxBid.setText(String.valueOf(currMaxBid));
    }

    void currHigherBidderSetText() {

        String currHigherBidder = bidHistory.lastElement().getBidderName();
        ftxtCurrHigherBidder.setText(currHigherBidder);
    }
}

//---------------------------------------------------------------------------------------------------------------------------
//          Class:                  BidderGui
//          Note:                   Graphical user interface for the bidder agents
//---------------------------------------------------------------------------------------------------------------------------

package BiddingAgent;
import jade.gui.GuiEvent;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.table.DefaultTableModel;

public class BidderGui extends javax.swing.JDialog implements ActionListener{

// Creates new bidder form BidderGui
    private AuctionAgent myAuctionAgent;
    private AuctionGui_22 myAuctionGui_22;
    String bidderType;
    String biddingPreference;
    String biddingAttitude;
   private BidderProperties myBidderProperties;
   private BidderConstants myBidderConstants;
    Vector<BidderProperties> biddersList = new Vector<BidderProperties>();
    Vector<String> test = new Vector<String>();
    Random generator = new Random();

    public BidderGui(java.awt.Frame parent, boolean modal, AuctionAgent a1) {
```

```
    super(parent, modal);
     myAuctionGui_22 = (AuctionGui_22) parent;
     myAuctionAgent = a1;
    initComponents();
}
public BidderGui(AuctionAgent a2) {
     super();
    myAuctionAgent = a2;
    initComponents();
}

@SuppressWarnings("unchecked")
//Automated code not shown here

private void bidderTypeComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    bidderType = (String) bidderTypeComboBox.getSelectedItem();
}

private void biddingPreferencesComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    biddingPreference = (String) biddingPreferencesComboBox.getSelectedItem();
}

private void biddingAttitudeComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    biddingAttitude = (String) biddingAttitudeComboBox.getSelectedItem();
}

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel)tblBidders.getModel();
    model.addRow(new Object []{ftxtBidderName.getText(),bidderTypeComboBox.getSelectedItem(),
    biddingPreferencesComboBox.getSelectedItem(),bidderBehaviourComboBox.getSelectedItem(),
    biddingAttitudeComboBox.getSelectedItem(),txtBargainLevel.getText()});
}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel)tblBidders.getModel();
    int rows = model.getRowCount();
    myAuctionGui_22.biddersList.clear();
    myAuctionGui_22.listModel.clear();
    biddersList.clear();
    for(int i=0; i<rows; i++){
        BidderProperties temp = new BidderProperties();
        temp.setName((String)model.getValueAt(i,0));
        myAuctionGui_22.listModel.addElement(temp.getName());
        temp.setType((String)model.getValueAt(i,1));
        temp.setPreference((String)model.getValueAt(i,2));
        temp.setBehaviour((String)model.getValueAt(i,3));
        temp.setAttitude((String)model.getValueAt(i,4));
        temp.setBargainLevel((String)model.getValueAt(i,5));
        biddersList.add(temp);
    }
    myAuctionGui_22.addBidderList(biddersList);
        GuiEvent ev = new GuiEvent(null, AuctionAgent.Event1);
        ev.addParameter(biddersList);
            myAuctionAgent.postGuiEvent(ev);
    dispose();
 }

private void btnRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    int i = tblBidders.getSelectedRow();
    javax.swing.table.DefaultTableModel model = (javax.swing.table.DefaultTableModel)tblBidders.getModel();
    model.removeRow(i);
```

```
    }

    private void conditionTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void ftxtCurMaxBidActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void ftxtCurBidActionPerformed(java.awt.event.ActionEvent evt) {
    }

    private void bidderBehaviourComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
   }

    private void bargainLevelComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
       if(bargainLevelComboBox.getSelectedItem()== "Random")
       {
            int roll = generator.nextInt(100) + 1;
          txtBargainLevel.setText(String.valueOf(roll));
          txtBargainLevel.setEditable(false);
       }
       if(bargainLevelComboBox.getSelectedItem()== "Manual")
       {
          txtBargainLevel.setEditable(true);
       }
    }
    // Variables declaration - do not modify
    private javax.swing.JComboBox bargainLevelComboBox;
    private javax.swing.JComboBox bidderBehaviourComboBox;
    private javax.swing.JComboBox bidderTypeComboBox;
    private javax.swing.JComboBox biddingAttitudeComboBox;
    private javax.swing.JComboBox biddingPreferencesComboBox;
    private javax.swing.JButton btnAdd;
    private javax.swing.JButton btnExit;
    private javax.swing.JButton btnRemove;
    private javax.swing.JFormattedTextField conditionTextField;
    private javax.swing.JFormattedTextField ftxtBidderName;
    private javax.swing.JFormattedTextField ftxtCurBid;
    private javax.swing.JFormattedTextField ftxtCurHigherBidder;
    private javax.swing.JFormattedTextField ftxtCurMaxBid;
    private javax.swing.JFormattedTextField ftxtEstClosingPrice;
    private javax.swing.JFormattedTextField ftxtLocation;
    private javax.swing.JFormattedTextField ftxtPostage;
    private javax.swing.JFormattedTextField ftxtSeller;
    private javax.swing.JFormattedTextField ftxtSellerRating1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
```

```
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTable tblBidders;
    private javax.swing.JTextField txtBargainLevel;
    // End of variables declaration

    DefaultTableModel GetTableModel() {
       return((DefaultTableModel)tblBidders.getModel());
             }
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("Save")){
       System.out.println("-----------MessageAuction from Gui--------");
          GuiEvent ev = new GuiEvent(null, AuctionAgent.Event1);
         System.out.println("-----------MessageAuction2 from Gui--------");
          ev.addParameter(biddersList);
          myAuctionAgent.postGuiEvent(ev);
        }
    }
}

//-----------------------------------------------------------------------------------------------------------------------
//                                      Classes for handling bidders and auctions
//-----------------------------------------------------------------------------------------------------------------------


package BiddingAgent;
import java.util.ArrayList;

public class BidderProperties {
    private String name;
    private String type;
    private String preference;
    private String attitude;
    private String bargainLevel;
    private String behaviour;
    public void setName(String name) {
      this.name = name;
    }

    public String getName() {
      return name;
    }

    public void setType(String type) {
      this.type = type;
    }

    public String getType() {
      return type;
    }

    public void setPreference(String preference) {
      this.preference = preference;
    }

    public String getPreference() {
      return preference;
    }

    public void setAttitude(String attitude) {
```

```
        this.attitude = attitude;
    }

    public String getAttitude() {
        return attitude;
    }

    public void setBargainLevel(String bargainLevel) {
        this.bargainLevel = bargainLevel;
    }

    public String getBargainLevel() {
        return bargainLevel;
    }

    public void setBehaviour(String behaviour) {
        this.behaviour = behaviour;
    }

    public String getBehaviour() {
        return behaviour;
    }
}

class BidderConstants{
    private float const_K;
    private float const_beta;
    private int const_blvl;

    public float getConstK(){
        return(const_K);
    }

    public void setConstK(float val){
        const_K = val;
    }

    public float getConstbeta(){
    return(const_beta);
    }

    public void setConstbeta(float val){
        const_beta = val;
    }

    public int getConstBlvl(){
        return(const_blvl);
    }


    public void setConstBlvl(int val){
        const_blvl = val;
    }
}

class BidderBids{
    private String name;
    private boolean bidPlaced;
    private String const_K;
    private String const_beta;
    private ArrayList<String> bids = new ArrayList();
        public void setBidPlaced(boolean fl){
        bidPlaced = fl;
    }
```

```
    public boolean getBidPlaced(){
        return bidPlaced;
    }

    public void setName(String nm){
        name = nm;
    }

    public String getName(){
        return name;
    }

    public void setK(String nm){
        const_K = nm;
    }

    public String getK(){
        return const_K;
    }
    public void setBeta(String nm){
        const_beta = nm;
    }

    public String getBeta(){
        return const_beta;
    }

    public void addBid( String bd){
        String temp = new String();
        temp = bd;
        bids.add(temp);
    }

    public int getBidsSz(){
        return bids.size();
    }

    public void clearBids(){
        bids.clear();
    }

    public String getBid(int idx){
        if( idx > bids.size())
            return "";
        else
            return bids.get(idx);
    }
}

package BiddingAgent;
import java.util.Vector;

public class AuctionProperties {
    private String    id;
    private float   reservePrice;
    private float   minIncrement;
    private float   predictedClosingPrice;
    private int     slots;
    private int count;

    public void setId(String i) {
       id = i;
    }

    public String getId() {
```

```
        return id;
    }

    public void setSlots(int i) {
        slots = i;
    }

    public int getSlots() {
        return slots;
    }

     public void setCount(int i) {
        count = i;
    }

    public int getCount() {
        return count;
    }

    public void setReservePrice(float Price) {
        reservePrice = Price;
    }

    public float getReservePrice() {
        return reservePrice;
    }

    public void setMinIncrement(float Increment) {
        minIncrement = Increment;
    }

    public float getMinIncrement() {
        return minIncrement;
    }

    public void setClosingPrice(float price) {
        predictedClosingPrice = price;
    }

    public float getClosingPrice() {
        return predictedClosingPrice;
    }

}

class ItemDescription {
    String name;
    private String condition;
    private String location;

     public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setCondition(String condition) {
        this.condition = condition;
    }

    public String getCondition() {
        return condition;
    }
```

```
    public void setLocation(String location) {
      this.location = location;
    }

    public String getLocation() {
      return location;
    }
}

 class BidHistory {
      private String bidderName;
      private float bidAmount;
      private float bidTime;

    public void setBidderName(String bidderName) {
      this.bidderName = bidderName;
    }

    public String getBidderName() {
      return bidderName;
    }

    public void setBidAmount(float bidAmount) {
      this.bidAmount = bidAmount;
    }

    public float getBidAmount() {
      return bidAmount;
    }

     public void setBidTime(float bidTime) {
      this.bidTime = bidTime;
    }

    public float getTime() {
      return bidTime;
   }
 }
```