REINFORCEMENT LEARNING IN THE CONTINUOUS DOUBLE AUCTION AND THE TRADING AGENTS COMPETITION

A Thesis Presented

by

JOHN J. BOADWAY

Submitted to the Graduate School
McGill University in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE

February 2003

School of Computer Science

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-88157-1
Our file Notre référence
ISBN: 0-612-88157-1

ACKNOWLEDGMENTS

I would like to take this opportunity to thank Professor Doina Precup, without whose patience and guidance this thesis would not have been completed. Her teaching skills and knowledge have been a source of inspiration and have allowed me to coherently formulate and present my ideas.

ABSTRACT

Electronic commerce plays an important role in many industries in today's economy. The advantages of using electronic means of trading (such as Internet auction servers) range from the easy, convenient and fast use, to the possibility of reaching a much larger market. As a result, the design and analysis of automated bidding agents has become a focus area both in research and in industry. However, many of the automated bidding strategies currently used in industry and academia are either very simple, or based on strong assumptions about the evolution of the market. In this thesis we investigate if reinforcement learning (RL) techniques can be successfully used to build automated trading agents. Market environments are typically non-deterministic and violate the Markovian assumptions needed to prove convergence of RL algorithms. Hence, the success of RL techniques has to be investigated empirically.

We present two case studies in which we develop RL agents for participating in auctions. The first case study focuses on the continuous double auction (CDA), a market mechanism used in many electronic trading venues. There are currently several automated bidding strategies for participating in the CDA, geared both toward personal profit and toward increasing the efficiency of the entire market. Most of these strategies use either static model-based prediction methods or simple heuristic techniques. We use model-free reinforcement learning to construct a bidding strategy for the CDA and empirically evaluates its performance against other well-known automated strategies. The learned strategy consistently increases the efficiency of the market, and it compares favorably to the other strategies in terms of profit as well. The second case study deals with the larger but related problem of interdependent electronic auctions. We describe an RL agent for the trading agent competition (TAC), and analyze its performance. This competition features multiple dependent auctions, and hence provides a much harder test-bed. The empirical results did not show the same success for the RL strategy as in the CDA environment. We attribute this problem to the difficulty of dealing with dependent auctions, in which the optimal strategy in one auction depends on the state of the other auctions as well.

RESUME

Le commerce eléctronique joue un role très important dans l'economie moderne. Les avantages des moyens de commerce eléctronique (comme les engins d'nchères) vont de l'utilisation facile et rapide, jusqu'à la possibilité d'avoir une clientèle plus nombreuse. La programmation des agents de commerce eléctornique automatiques est de plus en plus importante pour l'industrie, ainsi que pour les millieux de recherche. Mais la plupart des strategies employées maintenant sont trè simples, ou sont basés sur des hypothèses restrictives sur l'environnement de ces agents. Dans cette thèse nous recherchons l'emploi de l'apprentissage par renforcement (RL) construire automatiquement des agents de commerce eléctronique. Les marchés modernes n'obeissent pas les conditions théoretiques nécessaires pour la convergence des algorithmes RL. Nous présentons un étude experimental de l'usage des algorithmes RL.

L'étude comprend deux domaines reliés aux marchés d'enchères. Dans le premier domaine (CDA), nos agents ont toujours un effect positif sur la qualité du marché, et en même temps réalisent un profit plus gros que les autres agents automatiques. Le deuxième domain comprend des enchères simmultanées. Içi nous n'avons pas observé le même succès que dand le premier domain.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Electronic commerce plays an increasingly important role in today's economy. The advantages of employing electronic means of trading (such as Internet auction servers) range from the easy, convenient and fast use, to the possibility of reaching a much larger market. The design of software agents that can perform economic tasks on behalf of individuals or companies in different electronic markets is currently receiving a lot of interest in the research community, as well as in industry (see, e.g., [25], [19]). Much of this work is focused on designing and analyzing automated bidding agents for different kinds of auctions.

Many of the agents currently used in industry and academia are based on very simple heuristic strategies. For example, the eBay server provides each customer with an automated fixed markup strategy with a time heuristic, which increases (or decreases) a bid by an amount dependent on the time spent since the beginning of the auction. Existing empirical studies (e.g.,[28]) suggest that significant improvements can be obtained over such simple strategies. Another category of strategies are based on attempts to build a statistical model of the market or the other participants (e.g., [22], [14]). However, this is a very challenging task, since markets are very dynamic and often unpredictable. Hence, it is hard to build accurate models. Moreover, the models have to be based on assumptions about the market dynamics, which are often violated.

In this thesis, we explore the use of reinforcement learning (RL) as a tool for constructing agents for the continuous double auction. Reinforcement learning (RL) is a popular approach for learning from interaction with a stochastic, unknown environment [26]. RL algorithms have been used successfully in the past for many complex applications, ranging from constructing the world's best backgammon player to robotic control,, and from solving combinatorial optimization problems to elevator dispatching.

Auctions pose an interesting challenge to the development of RL agents. On one hand, it is easy to formulate the problem of constructing a bidding agent in the RL framework. It is also natural to expect that model-free RL algorithms would have the capacity to adapt quickly to fluctuations and trends in the market. On the other hand, markets and auctions often violate the Markovian assumptions underlying the theory of RL. Hence, it is not clear if RL agents would converge to a good strategy, converge to a bad strategy, or simply produce oscillatory or divergent behavior.

The goal of this thesis is to investigate the potential of reinforcement learning strategies in practical market environments. We present two case studies, focusing on markets with different auction mechanisms: the Continuous Double Auction (CDA) and the Trading Agents Competition (TAC). The two case studies involve developing simple yet innovative reinforcement learning strategies and comparing them against well-known existing strategies in competitive markets. In the CDA case study, we present an extensive empirical analysis of the learned strategies in markets with different structures. In the TAC case study, we mainly review the results of our participation in the 2001 international trading agents competition, held in Tampa, Florida. In this environment, systematic experimentation is much harder to provide, due to the complexity of the environment.

In both of the case studies the reinforcement learning strategy offers a new and completely different approach as compared to current and previous agent designs. This model free approach does not take into account the opponents, and hence allows a single strategy to compete against potentially many different strategies at different times. This adaptability is a major benefit of reinforcement learning, compared to other design approaches. The empirical results of this thesis show that reinforcement learning strategies have the capacity to perform on par or better than other state-of-the-art automated bidding strategies.

The thesis is structured as follows. Chapter 2 discusses related work along two different lines: bidding agents based on opponent modeling, and the use of model-free learning algorithms (algorithms which do not attempt to model other agents or the environment) for developing agents in multi-agent environments. Chapter 3 describes the continuous double auction, the reinforcement learning agent we designed for this task, several strategies that have been proposed in the research literature for this type of auction, and the empirical results obtained by our agent against a variety of other strategies. Chapter 4 presents the rules of the international trading agents competition, several agent designs used by competing teams, the design of the agent that we used in the 2001 contest, and an analysis of the results of our participation. Chapter 5 summarizes the conclusions of the thesis and presents avenues for future work.

CHAPTER 2

RELATED WORK

The work outlined and discussed in this section can be divided into two general categories: research that deals with modeling opponent agents in an auction environment, and research investigating model-free autonomous learning agents in multi-agent systems. The related research directly related to the CDA and TAC environments is summarizes in the corresponding chapters.

2.1 Building opponent models

One of the main approaches in building adaptive bidding agents is to use past market data in order to build a model of the market, and possibly of the other participating agents. Then, the model is used to simulate what will happen in the future, and the bidding strategy is based on the result of this simulation.

Hu and Wellman [16] investigate how an agent's performance is affected by its level of opponent modeling. A *0-level agent* does not model the underlying behavior of the other agents, but does model their actions by analyzing the history data of those actions. A *1-level agent* attempts to model the policy functions of the other agents, while assuming the other agents take actions based only upon their state (i.e., assuming they are 0-level agents). Inductively, an *i-level agent* assumes that all other agents are $(i-1)$-level agents. Their experiments using a continuous double auction simulator (similar to the simulator used in Chapter 3 here) suggest that the 0-level agents outperform the 1-level and 2-level agents. But they also show that if an i-level agent's assumptions are correct (if its model is flawless) then this agent outperforms the $0 - level$ agent. Due to the uncertainty involved in real on-line auctions, it is very unlikely that one could develop and maintain an approximately correct model. For example, in an environment like a real on-line CDA, opposing agents may enter or leave the auction at any time, unknown to the other agents. This causes rapid change in the model. Even attempting to model adaptive agents in their environment was not shown to be successful. Thus, the results in [16] serve to support the use of 0-level agents in highly dynamic and uncertain markets.

While [16] shows that an agent using the correct assumptions about the other agents (the correct model level) may be beneficial in the CDA, Vidal and Durfee [18] formally investigate this relationship in general multi agent systems (MAS). They use an oracle, M, for each agent in the system that returns the best action that that agent can take in its current state. They show that M may be constantly changing, thus making the learning difficult for agents of any modeling level. They introduce the notion of convergence and prove that once a MAS has converged, all models deeper than a 0-level model become useless. In the simulated CDA experiments outlined in the next chapter, we observed convergence in most of the environments. The above convergence theorem can thus be viewed as evidence that an RL agent (which is 0-level) may be more efficient than agents that use a deeper level of modeling.

Preist et al. [4] use a 1-level agent to act in multiple concurrent auctions, and show how this may be prosperous. Their agent builds a model using a belief function $B(x, q)$ that represents the probability that x bidders value the good under consideration at an amount greater than q, and then suggests bids to the user based upon this model. The main contribution of their paper is to show that auction markets on the Internet are inefficient (i.e., many trades are completed away from the equilibrium price), and thus many participants do not realize their full potential. These inefficiencies may result from individuals who are not capable of participating in large numbers of auctions, and therefore decrease seller competition. In their experiments, in which there are a number of different auctions and participants, the efficiencies of each auction increase when more bidders use their bidding agent, which allows participation in many auction protocols simultaneously. When more bidders use their agent, each bidder participates in a greater number of auctions, thus increasing the competition and efficiency in each market. Their agent was not tested against agents that do not explicitly model the auctions; that would be an interesting extension to this work.

An example of an agent that uses a model for a single auction is provided by Gimenez-Funes et al. [9]. They discuss an agent that uses both probabilistic (market-based) and possibilistic (opponent based) information in order to take a decision. Their agent description is general enough so that it will apply to a wide variety of auction types, although no experimental analysis is provided. Possibility-based agents were designed for both TAC [15] and the CDA [14], and in both cases they were quite successful.

Zeng and Sycara [7] apply Bayesian learning successfully to an environment similar to the CDA. Their strategy is similar to that of Preist et al. [4], except that it applies only to one particular auction at any time.

A connecting assumption in the papers that propose model based approaches to market environments is that it is necessary for a rational agent to model the behavior of the other agents. In all of the above papers, the experiments were successful when markets consisted of fixed strategies or similar model-based strategies. Model free learning strategies, such as RL, were not tested. Vidal and Durfeee [18] and Hu and Wellman [16] discussed the theoretical aspects of multi-agent systems, and in both cases 0-level models (which can be constructed using RL) were shown to have potential in uncertain, non-deterministic environments like the CDA and other auctions. Research that involves learning agents and 0-level agents will now be discussed and reviewed in detail.

2.2 Model-free learning agents in market environments

A number of recent papers analyze learning in multi-agent systems. The papers discussed here involve either auction markets, or cooperative learning agents in a MAS.

For example, Buffet et al. [20] propose an incremental decentralized version of reinforcement learning that can be used when a number of agents are learning cooperatively to achieve a goal. The incremental learning, in which the task gradually becomes more difficult, proves successful compared to non-incremental techniques. Unfortunately, on-line auctions, with all of the uncertainties attached to the other agent strategies and reservation values, do not easily allow for incremental learning. Buffet et al. also underline the important problems of multi-agent cooperative systems: combinatorial explosion, hidden global state, and the credit assignment problem. All of theses issues have to be tackled in the design of our TAC agent, and are discussed in more detail in Chapter 4.

The use of reinforcement learning for pricing agents has been investigated by Tesauro and Kephart in [12] and [13]. Two Q-learning agents compete in simulated markets by setting the price of an abstract good. This is similar to using RL from the buyers point of view in the CDA, in that the problem is "non-stationary and history-dependent" when other adaptive agents are affecting the environment. Just like our experiments in the CDA, their analysis is meant to be mainly a proof of concept. The environment setting are slightly simplified and unrealistic in that the state space (i.e., buyers profit functions) was fully known to both Q-learning agents, and a well-defined ordering was used in the price setting and bidding. One promising conclusion of their experiments is that the use of Q-learning helps to eliminate price wars.

Oliveira et al. [10] also use a variation of Q-learning for selling in competition in a first-price sealed-bid auction. Their Q-learning strategy performed well when tested against a fixed strategy. Again, this work is only a proof of concept because their strategy was only tested against itself and fixed strategy. They also endorse using 0-level agents in uncertain auction environments.

Another type of model-free learning, genetic algorithms, has been applied to a type of market protocol. Oliver [17] uses genetic algorithms in simulated negotiations and shows that their performance is on par with human counterparts. It would be of interest to see this type of learning applied to the CDA, where people have been outperformed by autonomous agents.

Model-free bidding has also been used to construct buyers in multi-auction systems. Anthony et al. [21] introduce a bidding agent that acts on behalf of the consumer by observing a number of auctions (with different protocols) and bidding appropriately. The agent uses a number of bidding constraints (time left in each auction, number of auctions, willingness to bargain, and desperateness). Associated with each constraint is a bidding tactic, and these tactics are combined with weights to realize the agents final strategy. Although these weights can be learned, the agent presented in the paper is essentially just a successful heuristic.

CHAPTER 3
THE CONTINUOUS DOUBLE AUCTION

3.1 Introduction to the Continuous Double Auction

The continuous double auction (CDA) is used in many major trading venues such as the New York Stock Exchange and the Chicago mercantile exchange [11], and it is becoming the protocol of choice for many Internet trading sites such as eBay, Amazon, etc. The reason for the prevalence of the CDA on the Internet as opposed to other types of auction protocols is probably its simplicity of use and analysis. Anyone may participate in any CDA at any time by placing a bid at any price for a specified quantity of a good. Bids and asks are broadcast to all participants. Whenever there are open bids and asks that are compatible in terms of the price and quantity to be transacted, a trade is executed immediately and announced to everyone. The participants make their decision based only on their own utility function and on the information about the bids, asks and transactions that have taken place. The analysis is free of confusion because the buyer and seller act in similar manner, and both aim for a beneficial transaction. Economic studies show that CDAs can achieve very high efficiency, and can respond rapidly to changing market conditions [29].

Many Internet CDA servers constrain the auctions to run for a specified length of time, and the auction start time may not be known in advance to the user. Thus, deploying an automated strategy (or agent) greatly simplifies the user's task of bidding for a given good. Moreover, empirical studies (e.g., [6]) showed that some automated strategies for the CDA actually outperform humans. So not only do the users simplify the task of bidding by using an automated strategy, but they may also be increasing the efficiency of the entire market, as well as make a better profit.

Currently there is no recognized standard or benchmark for analyzing automated strategies for on-line auctions. Many of the automated strategies used are not very sophisticated. For example, when a consumer uses eBay, the server automatically deploys an automated fixed mark-up strategy with a time heuristic, which increases (or decreases) the bid by an amount dependent on the time spent since the beginning of the auction. Existing empirical studies (e.g.,[28]) suggest that significant improvements can be obtained over such simple strategies.. From a practical standpoint there is much room for ideas and improvements concerning agent strategies.

The CDA offers an interesting environment for reinforcement learning (RL) agents. The CDA is a non-deterministic environment, in which agents can come and go, or they can change over time. Theoretically, RL has not been shown to converge in such non-deterministic environments. However, RL has been applied with some success to other less common types of auctions [10], and it has been used in many large-scale practical applications [26]. Favorable results in the CDA environment may bolster the belief that RL converges, and can produce good performance, in certain types of non-deterministic environments. Our goal in the experiments reported here is to show that RL can provide a simple, adaptable strategy that can benefit the entire market, as well as the individual participants in the market.

A number of competing strategies for the CDA have been introduced and compared in the research literature but there is yet no formal framework or testing benchmark for CDA markets. In the analysis that follows, we adopt the most prominent experiment design and testing methods from related research, and we compare our RL agent against several agent designs that have been proposed before.

3.2 The Continuous Double Auction Simulator

The CDA is an auction where there are an unrestricted number of buyers and sellers. At any time the buyers may broadcast to all other buyers and sellers a price for which they are willing to buy a specified quantity of a good or service. Similarly, the sellers may, at any time, broadcast to all other agents a price at which they are willing to sell a specified quantity of a good or service. If, at any time, a seller, named A, has issued a price that is less than or equal to the price that a buyer, say B, has issued, then A will transact the specified good or service with B. The price at which A and B transact the good or service is equal to the earlier broadcast price between A and B. The auction has a specified start time but not necessarily a specified end time. Sellers are not restricted to remain sellers and buyers are not restricted to remain buyers.

The simulator used for the experiments in this study is a discrete-time simulator based upon the description found in [28]. The simulator is given a predefined number of buyers, number of sellers and total time for the auction. The sellers remain sellers and the buyers remain buyers for the duration of the auction. At the start of each auction the agents are given a number (also predefined) of reservation values within a static range. These reservation values represent the value (i.e., personal worth) to that agent of the abstract good or service under consideration and are ordered from smallest to largest for sellers and from largest to smallest for buyers. Each agent may only have one bid in the auction at any one time. Once a bid is in the auction it may be updated but it cannot be removed unless a transaction occurs with that bid.

The simulator operates in time steps. At each time step each agent (buyer or seller) has a 25% probability of being active. If an agent is active then it may alter its current bid, otherwise it is unable to do anything. If the agent alters its current bid it may only raise its bid in the case where the agent is a buyer or lower its bid in the case where the agent is a seller. Note that the agents bid on behalf of one reservation value at a time with an ordering as explained above. The probability of being active was chosen as 0.25 to coincide with the work of Das and Tesauro [28], who found that this best emulates their continuous-time CDA environment MAGENTA.

After each time step the sellers' bids and buyers' bids are examined. If there is a seller bid that is less than or equal to a buyer bid, then the buyer and seller who own those bids undergo a transaction at the price of the earlier submitted of the two bids. If more that one buyer has a bid that is higher than a seller's bid, then the earlier submitted of the buyer bids is transacted. If the buyer bids were submitted at the same time step, then the buyer who trades with the seller is chosen randomly.

If a transaction takes place between seller A and buyer B then the bids from A and B are removed from the auction and the reservation values for A and B that were under consideration for the transaction are removed from the reservation lists of A and B. Once a reservation is removed from an agent's list, its next reservation value (under the specified ordering) is considered until their next transaction occurs. The process continues until the auction is complete, or until there are no more reservation values for that agent.

There are a number of metrics for analyzing the quality of the agents and of the market as a whole. Individual qualities can be assessed by the surplus obtained by the buyers and sellers. For the buyers, the surplus for each transaction is the buyer's reservation value less the transaction price. Symmetrically, for the seller the surplus for each transaction is the transacted price less the seller's reservation value. The most important metrics for the market as a whole are concerned with the total number of transactions and the number of transactions going to the bidder who values that good or service the most. The market efficiency measure used in [28] is the ratio of total surplus to theoretical surplus. The theoretical surplus is the total surplus that would have been amassed had all transactions occurred at the theoretical equilibrium. This is the measure of market efficiency that we will be using in the experiments described below. Note that other notions of market efficiency have also been proposed in the economics literature. For instance, Vickrey's measure of efficiency is defined as follows. If each of the i agents participating in a market has a reservation value, and these values are ordered from 1 to i, then an auction is efficient if it awards the cheapest offered item(s) to the agent(s) that values it the highest. So, in an efficient auction, if five items are offered, then the agents with the five highest worth values for that item will each buy that item [30]. We do not know of any study comparing the different measures of market efficiency. Our choice here was mainly based on computational considerations.

3.3 Reinforcement Learning Strategy

The goal of using reinforcement learning is to create an agent that can adapt to market volatility, without explicitly modeling other agent strategies or the market environment. Most current adaptive agent strategies use a model of the environment or model other agents over time. But in a realistic market, sellers. buyers and market dynamics will be continually changing. So our assumption is that using a strategy that does not use explicit models or assumptions about the market, but still retains adaptability, would be ideal. In attempting to achieve this goal we also desire to find an agent that creates an efficient market (as opposed to having one agent obtain minor gains at a greater cost to the rest of the population). In doing this we increase the opportunities for using CDAs as a fair and equitable auction protocol. It is not how realistic the environment is that is under scrutiny, but the concept of the RL strategy.

The agent design adopted for this analysis uses the Reinforcement Learning (RL) paradigm, in particular the Sarsa(λ) algorithm with eligibility traces [26]. In this paradigm, we must define a set of actions, a set of states, and a reward scheme for our agent. The rewards are associated with state-action pairs. The agent interacts with its environment at discrete time steps. At each time step t, the agent is in some state s_t and chooses an action a_t. One time step later, the agent gets a reward r_{t+1} and transitions to a new state s_{t+1}. The transition to the next state is governed by a probability distribution $P(s'|s,a) = Pr(s_{t+1} = s'|s_t = s, a_t = a)$ in each state. The goal of the agent is to learn a policy, which is a mapping from states to actions, $\pi : S \to A$, which yields a lot of reward in the long run. The policy fully specifies what action to choose in each state. In order to choose actions, the RL agent maintains an action value function, $Q(s,a)$, which approximates the total reward that can be obtained if the agent starts in state s, takes action a, and chooses optimal actions afterward:

$$Q(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a'),$$

where $\gamma \in [0,1]$ is a discount factor. Discounting is used to express the fact that rewards received right away are more valuable than rewards received later in the future.

In order to learn estimates of the action-value function, $Q(s,a)$, we use the Sarsa(λ) algorithm. This choice is based on the fact that this algorithm is known to perform well (empirically) in non-Markovian environments, even though no theoretical guarantees are provided. Since the CDA environment is also non-Markovian, we anticipated that this algorithm might perform better than other RL techniques. The Sarsa(λ) algorithm updates estimates of the action values after every transition observed in the environment. More specifically, the action value $Q(s,a)$ is updated toward the value of the reward and next state observed, using the following update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a),$$

where δ is the temporal-difference error:

$$\delta = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1} - Q(s_t, a_t),$$

e is the eligibility trace parameter and obeys the following rule:

$$e(s_t, a_t) = e(s_t, a_t) + 1$$

$$e(s,a) = \gamma\lambda e(s,a), \forall s, a$$

and $\alpha \in [0,1]$ is the learning rate parameter. In the experiments reported below, the parameters associated with the RL agent are as follows: $\alpha = 0.1$, $\lambda = 0.1$, and $\gamma = 0.9$. These parameters have not been optimized in any way, because we wanted to see if a fairly standard RL algorithm could produce good strategies, without involving much tweaking.

The state space for the agent participating in the CDA consists of two variables:

- the difference between the agent's current bid and the agent's reservation value

- the difference between the best offer and the agent's reservation value

So if MAX is the maximum possible bidding value and MIN is the minimum possible bidding value then the state space has size (MAX-MIN) x (MAX-MIN). Notice that no consideration of the number or type of other agents is taken into account. In the simulator we use reservation values between 100 and 200, and we limit the bids to maximum 400.

The set of actions for the agent in this CDA is of size three. These actions are as follows:

- If the agent has an outstanding bid than do not change this bid, otherwise do not submit a bid

- In the case where the agent is a buyer, increase the bid to the midway point between its current value and its reservation value. The case for the seller is similar

- Set the bid to the agent's reservation value in an attempt to complete a transaction

The state space (as outlined above) is small enough that a lookup table can be used to store all the action values in separate cells. In the future it may be necessary to use a function approximation technique in order to accommodate large state spaces.

No reward is given unless the agent participates in a transaction. When the agent does participate in a transaction, the reward given is equal to the surplus gained in that transaction.

During learning, the agent must explore all action in order to find out which ones are best. Our agent uses an ϵ-greedy policy, i.e., it chooses a random action with probability ϵ, and the current greedy action (with respect to the action-value function) with probability $1-\epsilon$. We use $\epsilon = 0.4$ in our experiments, in order to ensure enough exploration. When that agent chooses a random action, it is biased toward choosing to not change its bid with a probability of 0.6.

The agent design appears simplified because the main theme of this research is to explore the potential success of RL in these environments and not necessarily to optimize performance. It should also be noted that the parameters are by no means fine tuned for any of the participating strategies.

3.4 Competing Agents

In order to sufficiently and realistically test the RL agent's capabilities various other agents must be used as competition. [14] has suggested a possible benchmark of competing agent strategies for the types of experiments undertaken in this study. Three out of the four agents from this benchmark are used in the experiments discussed below. The three agents are the Zero Intelligence (ZI) agent, the Fixed Mark-up (FM) agent and Gjerstad and Dickhaut's (GD) agent. As well as these benchmark agents the Snipe [24] will also be used as in [27]and [28]. The specific parameter settings for these agents as used in the experiments will be discussed in experiment discussion section.

3.4.1 Zero Intelligence

The strategy of the ZI agent is to bid an amount that is uniformly selected between its reservation value and the maximum allowable bid for a seller and the minimum allowable bid for a buyer [2]. The idea here is that the agent be using the least amount of information and adaptability.

3.4.2 Fixed Mark-Up

A FM agent bids its current outstanding bid plus (or minus in the case of a seller) some pre-defined mark-up value on every time step that it is active [5]. It is similar to automated agents currently used on EBay save for the time between bids. For the purpose or the experiments in this study a mark up value of two was found to be adequate and successful.

3.4.3 Gjerstad-Dickhaut

This agent strategy requires more explanation, as it is the first of the competing agents with a notion of memory and adaptation. The GD agent uses a window of the history of bids and transactions to calculate a belief function [11]. For these experiments the window is of the last 25 time steps. The GD agent bids the price which maximizes the belief function, ie. the bid with the highest transaction probability. If a point in the belief function has not been seen before than a cubic spline interpolation is performed to obtain the belief value. Formally, the belief function, $f(p)$ is the following:

$$f(p) = \frac{AAG(p) + BG(p)}{AAG(p) + BG(p) + UAL(p)}$$

where $AAG(p)$ is the number of transacted seller bids in the history window with a price $\geq p$, $BG(p)$ is the number of buyer bids in the history window with a price $\geq p$, and $UAL(p)$ is the number of non-transacted seller bids in the history window with price $\leq p$ [14].

3.4.4 Snipe

The snipe agent is a heuristic based agent that bids if one of the following occurs:

- The auction is nearly over
- There is an extremely profitable deal
- There is a deal that would be more profitable than one undertaken in the last auction

Specifically, if the auction period comes within t percent of the end, and there is a non-profit-losing bid, or if the profit for transacting would be greater than any profit seen in the last period, or if the profit for transacting would be at least x percent of the maximum theoretical profit, then the agent will bid the required amount in order to complete the transaction. This is an adaptation of the snipe agent [24] as used in [28].

3.5 Experiment Design

The ideal way to evaluate the RL strategy would be to compare it to every other strategy in every market type (such as different numbers of buyers and sellers and altering the supply and demand). Due to the nature of the CDA environment (ie. unlimited numbers of buyers and sellers) and to the number of strategies continuously being introduced (see summary below), this brute force testing method becomes intractable. Apart from the experiments in [14], in which supply and demand in the CDA are varied and the results evaluated, other studies of agents in the CDA are of the form found in [28]. The types of experiments used in this analysis are also adopted from [28]. They include three different market environments that equally compare different types of agents in both mixed and homogeneous environments, as explained below.

The "homogeneous" environment is a market with three sellers and three buyers, all using the same strategy. This is a standard market experiment found in [14], [28], and [6] in which the efficiency of the agent population is measured. In some sense the goal of this field of research is to find an agent that creates an efficient market (as opposed to having one agent obtain minor gains at a greater cost to the rest of the population). In doing this we increase the opportunities for using CDAs as a fair and equitable auction protocol. Again it is not how realistic the environment is that is under scrutiny, but the concept of the RL strategy.

The "one-in-many" experiments are mixed market environments consisting of three sellers using strategy A, two buyers using strategy A, and one buyer using strategy B. This tests the benefit of using strategy B when all other agents use strategy A (ie. the incentive to deviate from A to B).

The "half-half" (or balanced) market is another mixed market that contains three sellers of type A and three buyers of type B. This is believed to be the fairest way to test two strategies against one another [28].

3.6 Experiment Results

Each experiment analyzed below is a CDA market with three buyers and three sellers. If the experiment does not involve an RL agent then the results are averaged over one thousand runs, which is known to be statistically accurate [14]. Each run consists of five periods of three hundred time steps. At the beginning of the first period of each run, each agent is randomly assigned ten reservation values between one hundred and two hundred. At the start of the other four periods of the run the agent is assigned these same limit values. In essence each experiment is in fact five repetitions of the market using the same initial values. This allows for greater market prediction and adaptation and follows the work of [28].

In the case when at least one of the agents is an RL agent then five hundred consecutive tests are performed, over which the RL agent learns. Each test consists of ten runs using an ϵ-greedy policy (see RL explanation) followed by sixty runs using a pure greedy policy. These runs are the same as the runs in the experiments where no RL agents are used, as explained above. This process of running five hundred tests is repeated five times, resetting the learned action-values to zero at the beginning of each repetition.

Note that the GD and Snipe strategies are not optimized and are used only to show the potential for an RL strategy in this type of market.

Over the three types of experiments discussed below two types of measurements are taken, efficiency and surplus. Efficiency is a measurement of the performance of the entire market and is defined as the ratio of total actual surplus to the surplus that would be obtained had every transaction occurred at the expected equilibrium price. The actual surplus is a measurement of the individual agents in comparison to the other agents in the market.

3.6.1 Homogeneous

These experiments consist of markets of three buyers and three sellers all of the same type. These results can be seen as a comparison for the later experiments where one of the six agents is of a different type.

The experiments that do not include an RL agent are summarized in table 3.1, which shows the market efficiencies and average surpluses. The FM, GD, and ZI agents all have very similar results resulting in an efficiency of approximately 0.80, whereas the Snipe agent has a much lower efficiency of about 0.73. This difference is to be expected and is found in other studies (see [28]). It should be noted here that the efficiency values are all generally lower in these experiments than in others of a similar nature. The reasons are first because only six trading agents are used, where other studies have used twenty and sixteen, and because the agents have not been completely optimized. The parameters used for the ZI, GD, and Snipe agents are similar to those in [28] and are discussed in the competing agents section. The fact that there are only six trading agents has an important impact on the efficiency because with less traders there are less bids and thus there is less competition, and so the efficiency suffers as a result.

Table 3.1. Homogeneous

Type of Market	Market Efficiency	Average Surplus
FM	80.77	606.2
GD	79.18	593.5
SNIPE	72.60	539.7
ZI	81.06	606.5

Figures 3.1 and 3.2 show the market efficiency and agent surplus vs. test number when six RL agents are used. These results are very promising, in that there is indisputable improvement over the course of the first 125 tests. In fact, over this time the efficiency increases from as low as 0.60 to higher than any of the other four homogeneous markets. Looking at the surplus graph (figure 3.2), one can note a similar learning pattern. Over the first 125 runs the RL agent increases its surplus from 400 to around 600 which is at least as much as the agents in the other homogeneous markets. In this market we see that all of the RL agents are learning a policy that at least increases their profits. The pattern of surplus improvement is similar for all RL agents in this environment (the sellers are not shown for clarity), so it can be hypothesized that each agent is learning a similar policy. One interesting area for future investigation is to analyze and compare the strategies for each RL agent in this market.

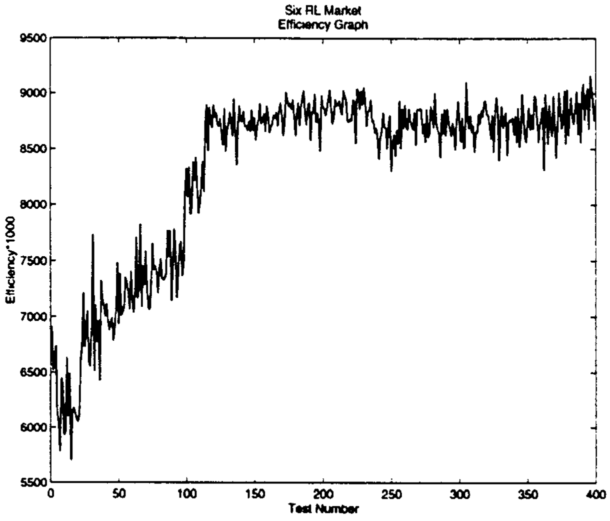

Figure 3.1. Homogeneous market of RL agents

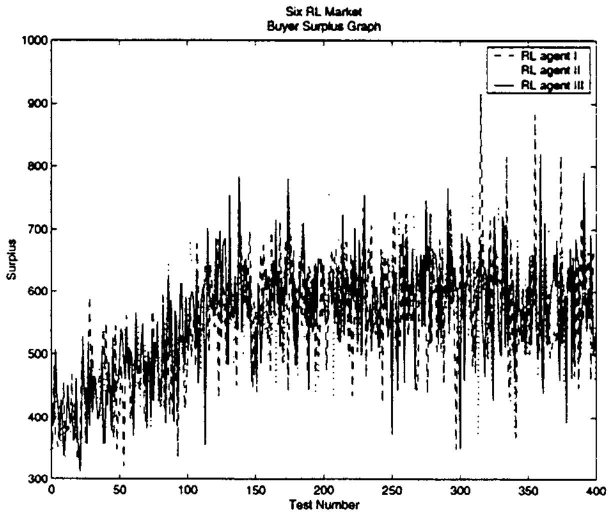

Figure 3.2. Homogeneous market of RL agents

3.6.2 One-in-Many

These experiment consist of a buyer of one type and two buyers and three sellers of another type. The experiments that do not consist of an RL agent are summarized in tables 3.2, 3.3, 3.4, and 3.5. Table 3.2 consists of the experiments that have five FM agents. The rows show the statistics when the single buyer is of the type stated. For example, the final row shows, from left to right, the efficiency of the market when there is one Snipe agent and five FM agents, the average surplus of the FM seller agents, the average surplus of the FM buyer agents, and the average surplus of the Snipe buyer agent. Table 3.3, 3.4, and 3.5 are similar except that the five agents are ZI, GD, and Snipe respectively. These tables show the incentive of a single trading agent to deviate from a market consisting of a homogeneous set of trading agents.

Table 3.2. Five FM.

Lone Agent	total Market Efficiency	Avg FM Seller Surplus	Avg FM Buyer Surplus	Lone Buyer Surplus
ZI	80.65	751.6	511.0	332.0
GD	80.43	814.6	510.5	165.0
Snipe	78.94	493.0	735.0	577.0

Table 3.3. Five ZI.

Lone Agent	total Market Efficiency	Avg ZI Seller Surplus	Avg ZI Buyer Surplus	Lone Buyer Surplus
FM	80.49	398.3	829.5	790.0
GD	80.43	811.6	511.0	161.0
Snipe	79.29	346.3	915.0	676.0

It is interesting to note that in all cases when the single agent is a Snipe agent, the buyers surpluses are much higher than the sellers, furthermore the surplus of the Snipe buyer is less than that of the other buyers. In fact the Snipe agent performs better when it is alone than when it is in a market with other Snipe agents. Although

Table 3.4. Five GD.

Lone Agent	total Market Efficiency	Avg GD Seller Surplus	Avg GD Buyer Surplus	Lone Buyer Surplus
FM	79.09	407.3	742.0	843.0
ZI	79.66	493.6	625.0	831.0
Snipe	77.36	362.2	877.0	625.0

Table 3.5. Five Snipe.

Lone Agent	total Market Efficiency	Avg Snipe Seller Surplus	Avg Snipe Buyer Surplus	Lone Buyer Surplus
FM	71.03	546.6	589.0	372.0
ZI	73.48	633.0	460.5	460.0
GD	73.32	745.2	375.5	271.0

when one single Snipe agent is introduced the total market efficiency decreases. Also, when there are five Snipe agents in the market the single other agent performs no better (and in some cases much worse) than the five Snipe agents, yet the total market efficiency remains relatively unchanged as compared to the homogeneous markets (ie. without having deviated to a Snipe agent).

In the cases when one GD buyer is introduced the market efficiency does not suffer, and in fact increases when there are five Snipe agents. But, the surplus of the buyers drastically decreases and is especially low for the GD agent. This may be a sign of strong competition among the buyer agents at the introduction of a GD agent or it may be that there is not enough trading agents to maintain a high enough level of market activity to enable the GD belief state predictor to be accurate. Also the history window size of the GD agent has not been optimized for this type of market.

Whether there is one ZI agent or five ZI agents the market efficiencies do not significantly change, but the single ZI agent does obtain the lowest surplus when the other agents are FM, and the highest surplus when other agents are GD. Similarly when the single agent is a GD agent and there are five ZI agents the GD agent performs very poorly (at no cost to the market efficiency). When the single agent is an FM agent and there are five ZI agents the ZI buyers do better than the FM buyer but the ZI sellers perform much worse.

The five Snipe agent markets present an interesting breakdown. The market efficiency remains relatively low no matter what the other agent is, but the other agent does not receive as much surplus as the Snipe agent would have if it were a homogeneous market. This is an example of a parasitic agent; one that profits at the expense of the entire market.

From these experiments under similar environment settings and agent parameters it is beneficial to deviate to an FM agent or a Snipe agent when the market consists entirely of GD or ZI agents. It is also beneficial to deviate to a ZI agent when the entire market consists of GD agents. Otherwise the incentive to deviate is either neutral or negative.

When there is an RL agent involved the analysis changes slightly because the RL agent uses a number of trials to learn a policy. Figures 3.3-3.18 summarize the efficiencies and surpluses of the RL agents and other agents in each market where one RL agent is present. In all cases except when there is five FM agents the market efficiencies improve. When there is one RL agent and five Snipe agents the market efficiency begins around 0.79 and after less than fifty runs, fluctuates around 0.84. Similarly the surplus of the RL buyer is much lower than that of the Snipe buyers, but by the one hundredth run is at least equivalent. The relatively simple bidding policy of the Snipe agent may serve to simplify the apparent RL learning process in this type of market.

The RL agent also undergoes an improvement phase over the first one hundred runs when the market consists of five GD agents. In this case the market efficiency begins around 0.91 and seems to stabilize at over 0.92. The surplus also shows strong signs that the RL agent is learning a more profitable policy. At the beginning of the tests the RL agents surplus is less then 600, whereas the GD agent's surplus is close to 800. Steady improvement of the RL agent's policy results in the surplus of the RL agent being greater than that of the GD agent by around the one hundredth test.

In all one-in-many markets, where the lone agent is an RL agent, the market efficiency ends up higher than it was with the homogeneous market. Thus, there is incentive to deviate to a trained RL agent in an otherwise homogeneous market of any agent type. This leads to an important area of future work and experimentation: testing the performance of RL agents in one type of market that are trained in a different type of market. Similarly, testing an RL agents performance in a market that is continuously changing (in terms of opponent type) would be interesting and conclusive.

In the case when there is one RL buyer and five FM agents the learning is not so obvious. The efficiency does not seem to significantly alter, but it does begin at a high level of over 0.90. The surplus graph (figure 3.6) does show signs of learning, as the RL agents surplus increases to to the level of the FM agents surplus after seventy-five tests. At this point the buyers surpluses seem to take a slight down turn, possibly an indication that the market is becoming more competitive from the buyer's standpoint - more evidence that the RL agent is learning.

In all cases the total market efficiency is higher when there is and RL agent, and in some cases even prior to learning a more powerful policy. Before the RL agent has learned any policy, the strategy can be seen as a randomized fixed mark-up strategy, where the mark up value is chosen randomly between three choices. This leads us to believe that the relatively worse performance of the RL agent in a market with FM agents (as opposed to other types of agents) is due to the generally smaller mark-up value for the FM agent. Thus, it may be the case that the RL agent does not learn to wait until a more suitable deal appears. This problem may be tackled by adding

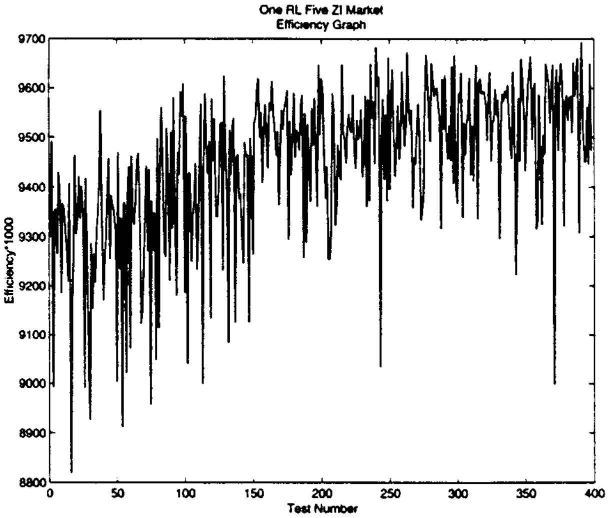

Figure 3.3. One RL Five ZI Efficiency

another action to the RL agent that causes the RL agent to wait a number of time steps or by optimizing the RL agents learning parameters.

When there are five RL agents in the market and one Snipe agent the market shows drastic signs of improvement as the efficiency begins around 0.70 and after two hundred runs is close to 0.90. The graph (figure 3.10) showing the surplus of this market shows signs that as the RL agent learns, it benefits the entire market. This is because the Snipe buyers surplus increases at the same rate as the RL agents surplus increases, although the Snipe agents surplus is always slightly lower than the RL agents.

The market with one GD and five RL agents shows similar signs of drastic efficiency improvement. Over three hundred runs the efficiency begins around 0.65 and ends around 0.85. The market with one FM agent or one ZI agent and five RL agents also has an increasing efficiency trend, although not as drastic as the previous two markets.

An important point of further research at this point would be to test the learning capabilities of the RL agent when the market is changing. Realistically the opponents may constantly vary and adapt to changing conditions. Also the supply and demand

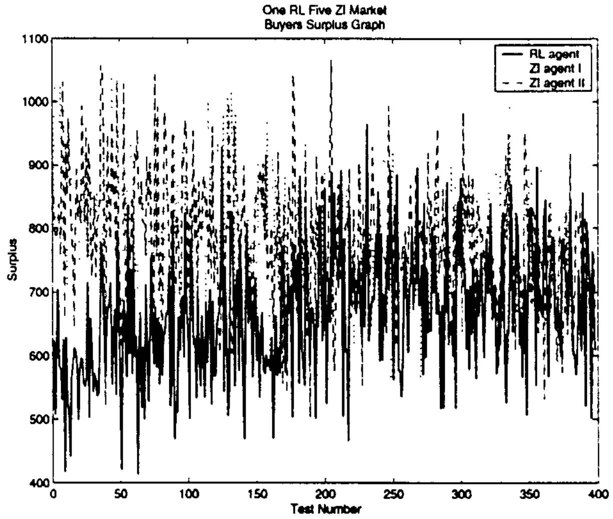

Figure 3.4. One RL Five ZI Surplus

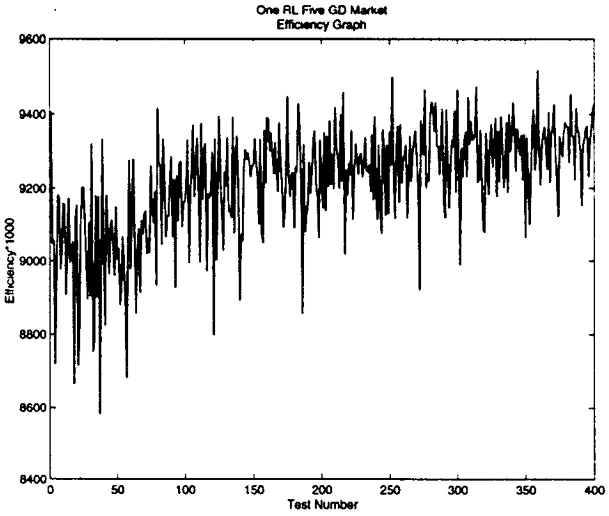

Figure 3.5. One RL Five GD Efficiency

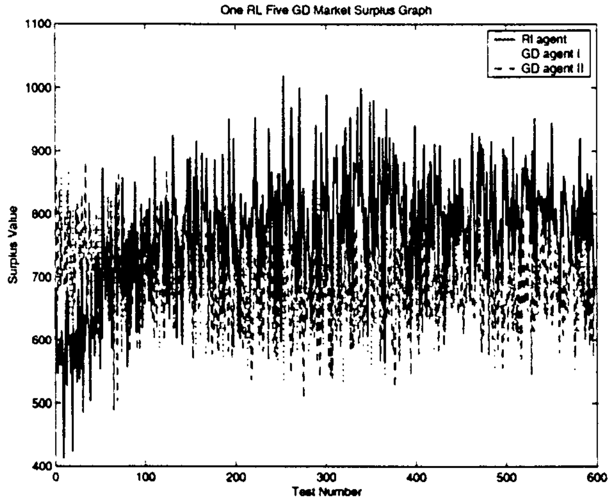

Figure 3.6. One RL Five GD Surplus

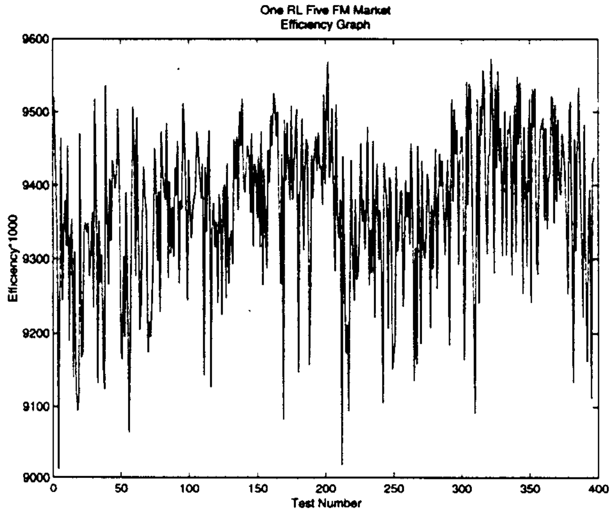

Figure 3.7. One RL Five FM Efficiency

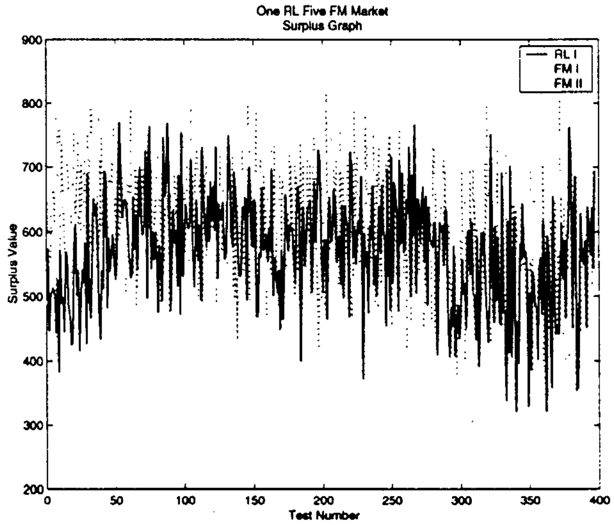

Figure 3.8. One RL Five FM Surplus

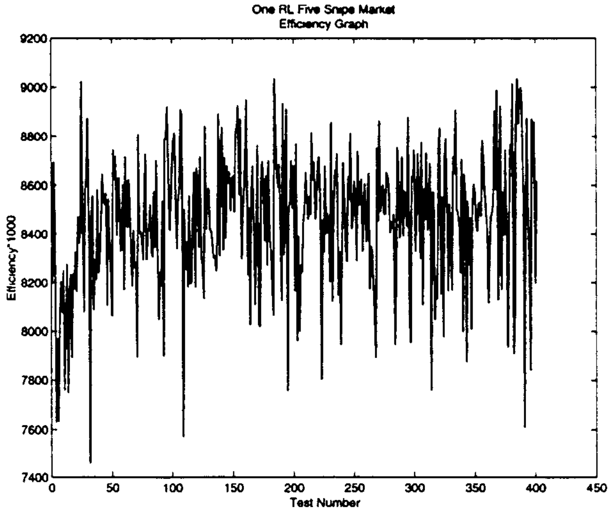

Figure 3.9. One RL Five Snipe Efficiency

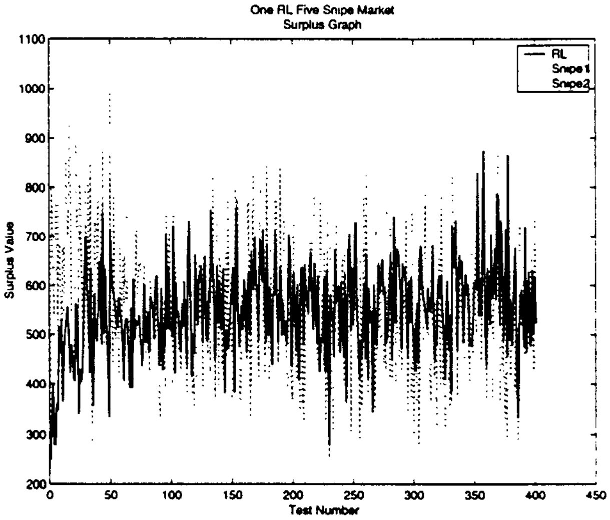

Figure 3.10. One RL Five Snipe Surplus

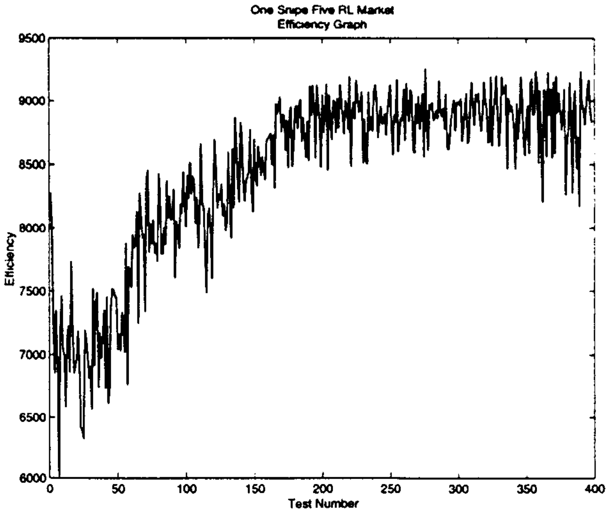

Figure 3.11. One Snipe Five RL Efficiency

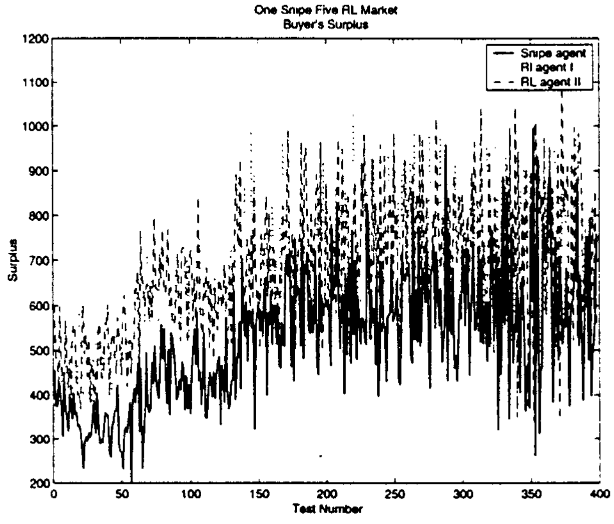

Figure 3.12. One Snipe Five RL Surplus

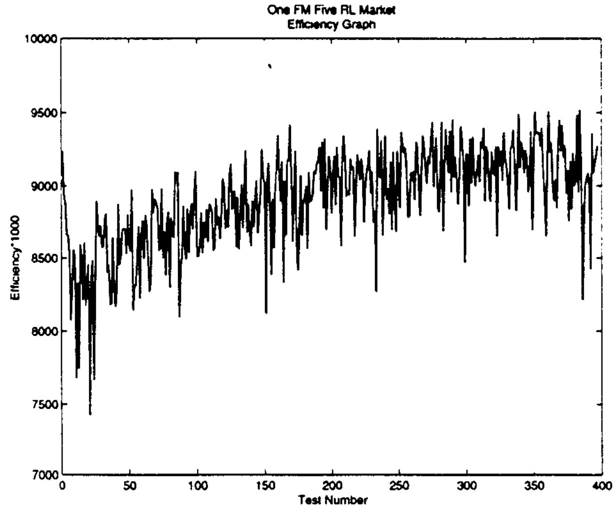

Figure 3.13. One FM Five RL Efficiency

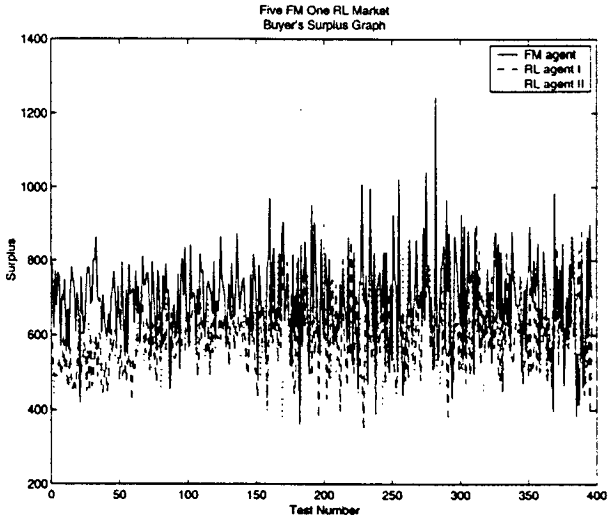

Figure 3.14. One FM Five RL Surplus

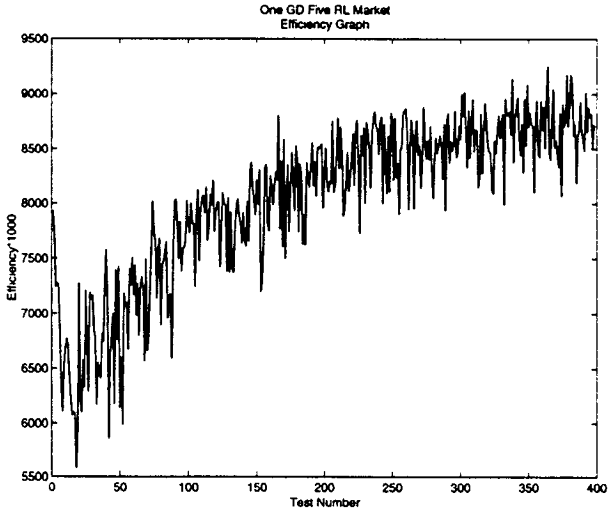

Figure 3.15. One GD Five RL Efficiency

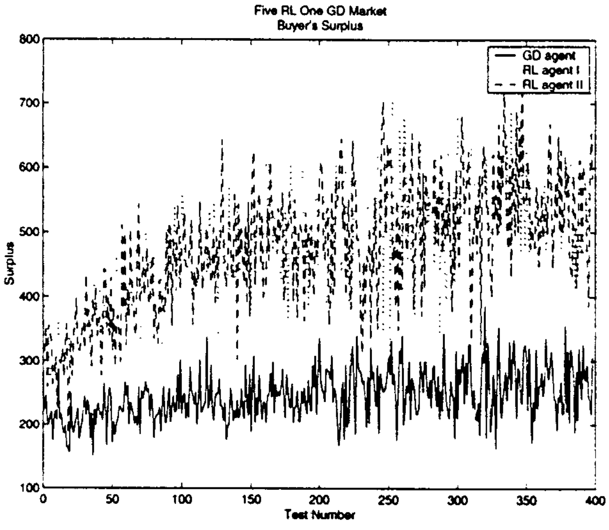

Figure 3.16. One GD Five RL Surplus

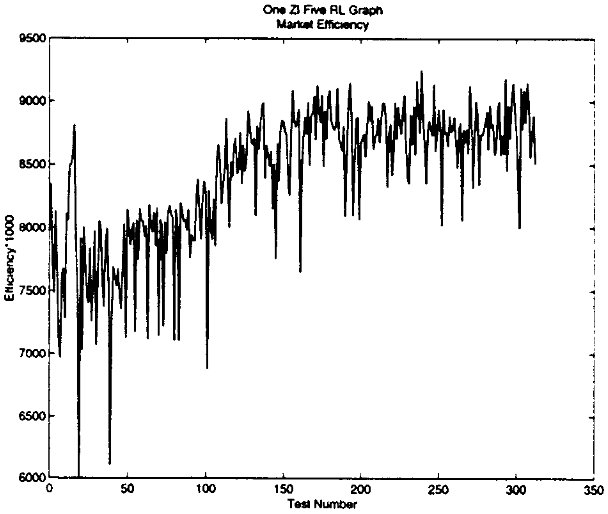

Figure 3.17. One ZI Five RL Efficiency

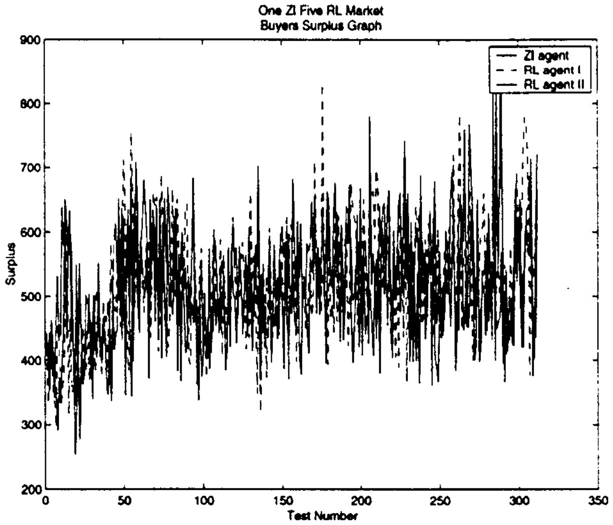

Figure 3.18. One ZI Five RL Surplus

rates will not be constant, as they are in these experiments. It remains to be shown how heavily the learning capacity of the RL agents depend on the other agents, on the market mechanism, or on the economic dynamics of the market. Because the RL agents learning does not depend directly on a model of the market or on modeling opponent strategies, the expected performance of RL strategies in changing markets is promising.

3.6.3 Half and Half

These experiments consist of three buyers of one type and three sellers of another type. The experiments that do not contain any RL agents are summarized in tables 3.6, 3.7, 3.8, and 3.9. Table 3.6 contains experiment results when three snipe agents are included, the columns indicate (from left to right) market efficiency, the average snipe surplus, and the average surplus of the other agent. Table 3.7, 3.8, and 3.9 are similar, and summarize markets of three FM, three ZI, and three GD.

Whenever the Snipe agent is involved it outperforms the other type of agent in terms of surplus, and when the other agents are the FM type then the efficiency is lower than when either type of agent is in the homogeneous case. Even though the

Table 3.6. Three Snipe.

	total	Avg Snipe	Avg Other
	Market	Surplus	Surplus
Other Agent Type	Efficiency	Surplus	Surplus
FM	59.64	509.3	385.3
ZI	74.67	755.0	371.3
GD	75.68	994.3	132.3

Table 3.7. Three FM

	total	Avg FM	Avg Other
	Market	Surplus	Surplus
Other Agent Type	Efficiency	Surplus	Surplus
Snipe	59.64	385.3	509.3
ZI	82.28	1000.3	219.0
GD	80.83	608.7	506.7

Table 3.8. Three ZI

	total	Avg ZI	Avg Other
	Market	Surplus	Surplus
Other Agent Type	Efficiency	Surplus	Surplus
Snipe	74.66	371.3	755.0
FM	82.28	219.0	1000.3
GD	81.04	937.0	287.7

Table 3.9. Three GD

	total	Avg GD	Avg Other
	Market	Surplus	Surplus
Other Agent Type	Efficiency	Surplus	Surplus
Snipe	75.68	132.3	994.3
FM	80.83	506.7	608.7
GD	81.04	287.7	937.0

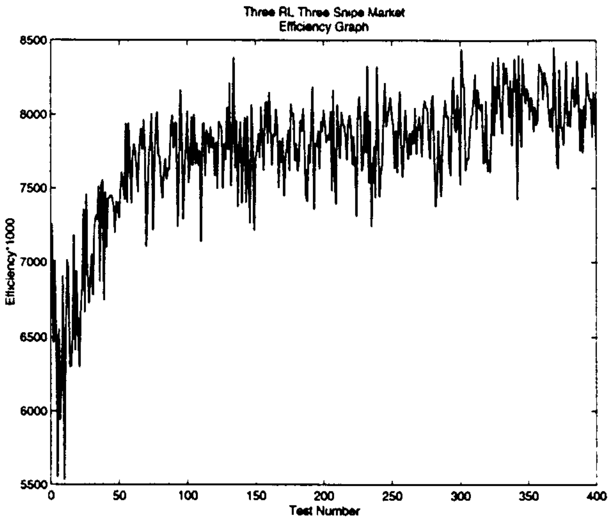

Figure 3.19. Three RL Three Snipe Efficiency

homogeneous efficiency of the Snipe agent is the lowest, their performance is greater than the other agents when they are half of the participants.

The GD agents gain a lower surplus than the three Snipe and the three ZI agents, but are comparably equal to the three FM agents when in a market together. The FM agents gain a higher surplus than the ZI agents when in a market together but have a similar market efficiency as when there are six FM or six ZI agents.

In each case when the RL agents are involved , except for when there are three FM agents, the efficiency increases by at least fifteen percent. This is a sign that the agent is profitably altering its policy. In the case when there are three FM and three RL agents the efficiency increases slowly to around 0.96. The RL agent gains a much higher surplus in every case, again except for when there are three FM agents. When there are three FM agents the average surplus for the RL agent increases from 400 to 450 and the average surplus for the FM agents remains around 800. These results are summarized in figures 3.19-3.26.

Judging by the results in the half and half experiments, after a short period of learning the RL agents are more beneficial to the market when more than one RL agent is present than are any other type of agent. With the exception of FM agents

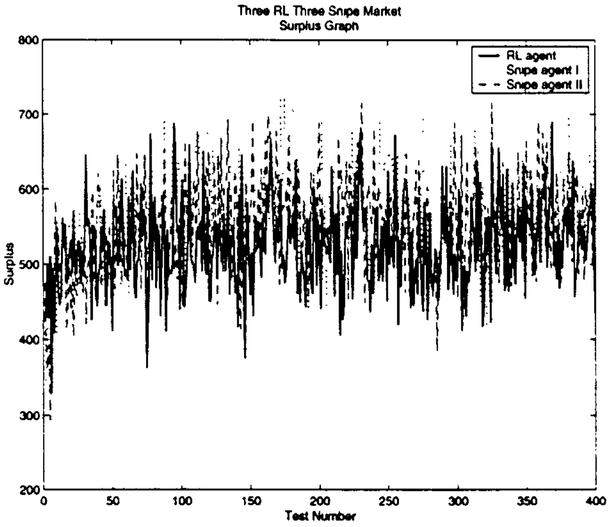

Figure 3.20. Three RL Three Snipe Surplus

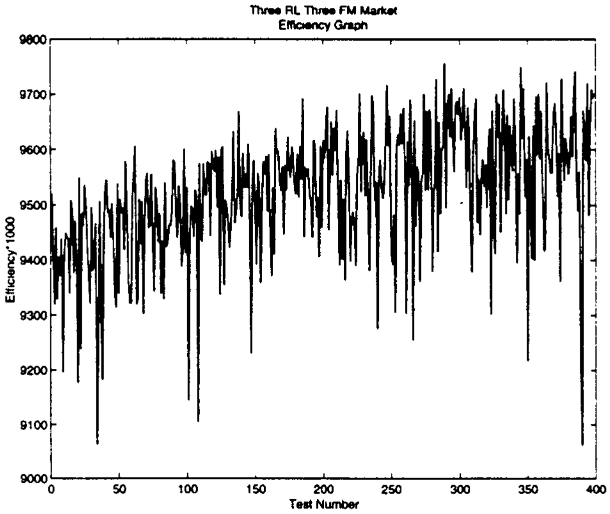

Figure 3.21. Three RL Three FM Efficiency

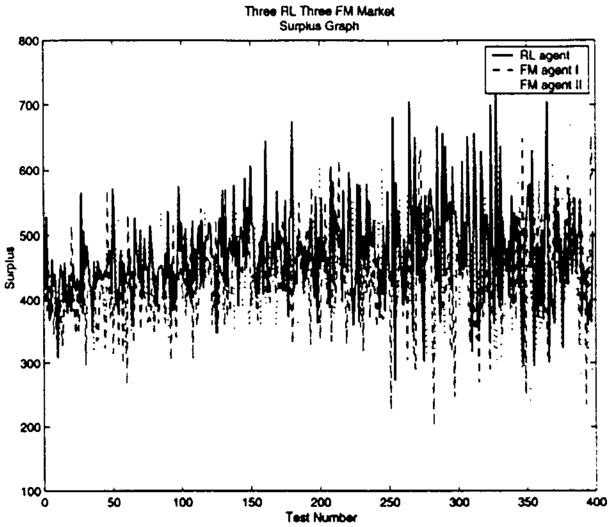

Figure 3.22. Three RL Three FM Surplus

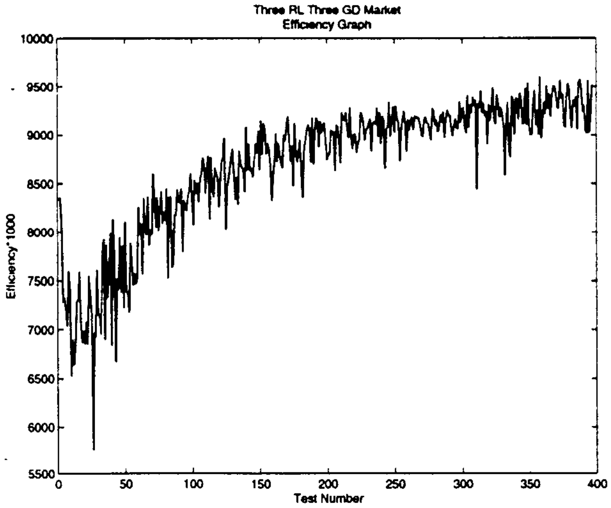

Figure 3.23. Three RL Three GD Efficiency

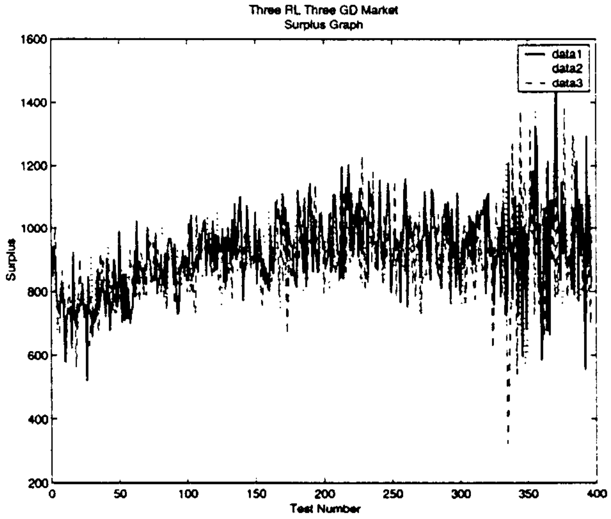

Figure 3.24. Three RL Three GD Surplus

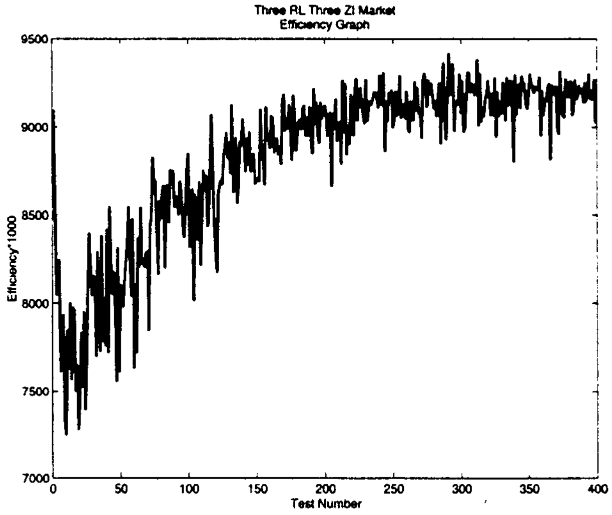

Figure 3.25. Three RL Three ZI Efficiency

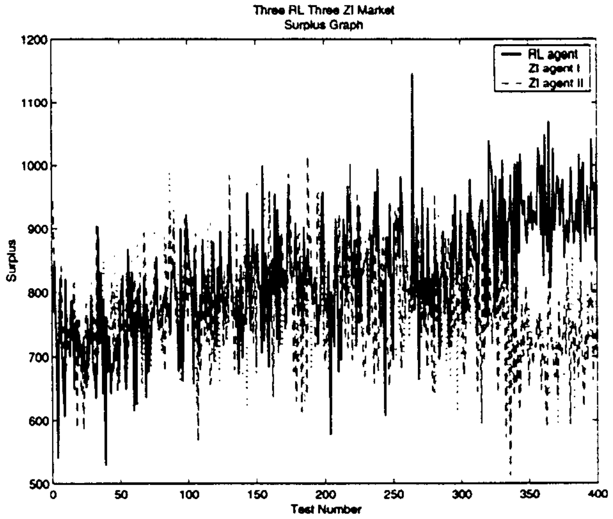

Figure 3.26. Three RL Three ZI Surplus

the RL agents also gain more surplus than other agents when used in a group with them.

3.7 History

Research in experimental economics in the CDA market began with Smith's experiments involving humans [29]. The results found in [29] were used as a comparison for the "zero-intelligence" (ZI) automated agents introduced in [2]. In [2] the ZI agents were used as an attempt to show that the market structure is largely responsible for the high levels of allocative efficiency, and they show that the results in the human experiments from [29] are matched using the ZI strategy. But their claim that "....trader's attempts to maximize their profits, or even their ability to remember or learn about events of the market, are not necessary for convergence [of the transaction price to the theoretical equilibrium price]" are shown, quantitatively, to be untrue in [3].

Building upon these results [3] introduce the "zero-intelligence-plus" ZIP agent. This is the first example of an adaptive automated strategy. The ZIP agent maintains a profit margin variable that it uses (together with the agent's reservation value) to obtain its current bidding (or selling) price, p. After each transaction in the auction the ZIP agent alters p in the direction of the trade price (unless the profit margin decreases). For each outstanding bid in the auction the ZIP agent alters p in the direction of surpassing the outstanding bid toward a transaction price. Sellers and buyers act symmetrically. [3] show quantitatively that homogeneous markets of ZIP agents are much closer to human data in similar markets than are ZI agents.

From this point in the research time-line, the emphasis is shifted from automated strategies that attempt to match human capabilities in the CDA to strategies that realize greater efficiencies and stronger results than their human counterparts. This was started by the results in [6] that show that some automated strategies (including GD and ZIP) are at least as capable as humans in controlled experiments (similar to the experiments used in this study).

A number of such competing automated strategies have been introduced and analyzed in the literature, those that have not been discussed above will be outlined here.

[27] adapt dynamic programming methods (DP) to a "strategic sequential bidding algorithm". States are represented by an agent's holdings, and the transition model is calculated from the market history and a forecast of the future changes. It is an extension of the GD agent in that it uses a similar belief function to build its model, though the computational overhead is unclear and may be a problem. This DP strategy is compared experimentally to the GD strategy, and is found to be at least as competent, and even superior in some cases.

[14] bases their agent strategy around fuzzy logic (FL). This FL strategy uses heuristic fuzzy rules and a fuzzy reasoning mechanism to decide what bids or asks to place [14]. This is extended to an adaptive version, where the FL agent adapts to the supply and demand levels by altering its aversion to taking risks. This strategy is shown to outperform GD, FM, and ZI strategies in varying supply and demand

settings when there are no abrupt changes in supply and demand. It is also shown to be highly efficient in homogeneous markets. The results are not conclusive when there are abrupt changes in the supply and demand, and is left as future work.

[22] create a bidding strategy based a stochastic model of the environment, called a p-strategy. This p-strategy creates a Markov Chain of the current auction where each state of the chain is an ordered list of the bids. For example the state "bbss" would represent an auction that has two standing buy bids (bb) and two standing sell bids (ss) where the two buy bid amounts are less than the sell bid amounts. Using this model transition probabilities and utility values were discovered enabling automated agent decisions. They found that this strategy was detrimentally resource consuming ($O(pn^{3})$, where p is the number of bids and n is the number of states in the MC), and so an adaptive technique was used to decide when to apply the p-strategy. This adaptive technique learned from experience when to use the p-strategy and when to use a fixed markup (ie constant bid increment) strategy. The goal of the p-strategy was to model the behavior of the other agents without modeling their 'internal reasoning', and [22] attempt to show that "an agent should use a stochastic model of the auction process while ignoring the fact that other agents behave strategically". Their p-strategy worked just as well as the competing strategies it was tested against although there is and admitted large overhead of model building for each auction.

3.8 Conclusion and Future Work

In each of the above series of experiments the RL agent showed strong signs of learning, and in almost every case outperformed its counterparts. Even more importantly the RL agent increased the total market efficiencies, improving the market competition and in some cases the opponents surplus as well.

These experiments form a strong foundation for the potential for the investigation of RL in the CDA. In terms of conclusive evidence of superiority over other bidding algorithms, more work needs to be done. For instance, testing RL agents in markets that do not have fixed buyers and sellers, testing RL agents in markets that have varying supply and demand (as in [14]), and even exploring larger action spaces and function approximators to refine the agents policy. These changes would imply a more realistic CDA set-up and a more advanced RL algorithm, which could then be tested more rigorously. The purpose of testing the RL strategy the CDA market was to prove that it does work, and to show the potential gains to be had by using RL in a more advanced manner and in more advanced markets, as explained above.

There are advantages to using model free learning agents, such as RL, in the CDA. Agents with explicit models have shown to be time consuming and depend on fixed markets (see history section). It is foreseeable that a model free approach, having proved itself worthy of investigation here, will be able to adapt to highly volatile and changing markets. Evidence for this comes from the fact that the agents policies improve quickly, realizing a profitable strategy in fixed markets without taking into account the types or strategies of other agents.

Analysis of the types of policies learned by the RL agents and of the convergence properties may be necessary if RL is to be studied in more realistic and complex markets.

Most importantly, the RL agent must be able to learn off-line until it's policy converges. Otherwise it will be of little use in any practical setting. In the experiments presented above, the number of training runs and test runs were arbitrarily chosen and this does not give us a strong sense of when the agent should optimally begin using a greedy policy. In other words, the convergence properties of the RL agent need to be studied in future work.

CHAPTER 4

THE TRADING AGENTS COMPETITION

4.1 Introduction to the Trading Agents Competition

The trading agents competition took place in October of 2001 in Tampa Florida. It was a market competition in which each competitor was an autonomous bidding agent. The participants included industry and academic professionals from AT&T, Carnegie Mellon University, and Stanford to name a few.

The TAC market is somewhat more complicated than the CDA and this makes it much more difficult for the reinforcement learning agent to learn a successful policy. Furthermore due to the non-deterministic nature of the market this policy may fluctuate. This complicated nature of TAC arises from the fact that there are three concurrent and somewhat interdependent auctions, in which the selling price may randomly change. In fact, the CDA is a sub-problem of TAC, which gives promise to the success of RL in TAC. The reason for using TAC as a case study is to push the capabilities of RL even further and examine the results.

There are many ways of defining the state space, action space, and reward structure for the reinforcement learning agent in the TAC environments. What was used in this case study was deemed to be the simplest and most intuitive. As can be seen from the competition results there is much to be improved upon. But, it should be noted that there is much room for possible improvements.

4.2 The Trading Agents Competition

The trading agents competition (TAC) is a multi-agent, multi-auction game in which there are three different interdependent auctions. The premise of the game is that each agent acts as a travel agent and attempts to purchase a travel package (from TACtown to Tampa) for eight clients over the period of five successive days. The clients are defined by a set of preferences, as explained below.

In each game, lasting twelve minutes, eight agents compete against each other in three auctions for the three elements of the travel package: flight tickets, hotel reservations, and entertainment tickets. A travel package for a single client must include an in-flight, out-flight, and hotels for each day they are in Tampa. There are two types of hotels, Tampa Towers and Shoreline Shanties, and each client may only stay in one type of hotel for the duration of their trip. The entertainment tickets are of three types: alligator wrestling, amusement park, and museum. Each client may use a maximum of one entertainment ticket for each night they are in Tampa, and may only use one of each type of entertainment ticket for the duration of their stay. At the end of the twelve minutes a utility score is calculated for each client based upon their travel package and preferences.

The final score for each agent is the sum of each of the eight clients utility scores, minus the agent's total expenses, minus a penalty score. This is explained in detail below.

4.2.1 The Auctions

4.2.1.1 Flights

There is one auction for each type of flight (in-flight or out-flight) on each night. Note that there is no out-flight on the first day and no in-flight on the last day. The flight auctions are continuous one-sided auctions that close at the end of the game. There is an unlimited amount of flights for each auction. At any time the agent may submit a buy-bid for any number of flights, but the agent is never permitted to submit a sell bid. Once a bid is placed in the auction it remains there until it is withdrawn by the agent or matched. A buy bid is matched if there is a sell bid (the price set by the TACAIR auction server) with a price less than or equal to the buy bid price. If a bid is matched then there is a transaction at the sellers price (the current price of the flight) for the number of flights matched in the buyers bid.

The price of a flight fluctuates according to a stochastic random walk but is restricted to be between 150 and 800 at all times. The initial price of the flights is chosen uniformly between 250 and 400 for each flight auction, and is altered every 24 to 32 seconds by a uniformly chosen amount between -10 and $x(t)$, where t is the time since the game's beginning and $x(t)$ is a linear interpolation between 10 and the final bound x, where x is unknown to the agents. Formally $x(t) = 10 + (t/12:00) * (x - 10)$. x is chosen independently and uniformly in $[10, 90]$ for each flight auction and is not revealed to the agents.

By itself the Flight auction seems more simple than the CDA, thus an RL may perform well in it. It the the interdependence of the three types of auctions that cause the difficulty.

4.2.1.2 Hotels

There is one auction for each day and hotel type (Shoreline Shanties or Tampa Towers), except for the last day when hotels are not needed. The auctions are English ascending multi-unit auctions that close at a time unknown to the agents. At four minutes into the game the first randomly selected hotel auction closes and on each minute thereafter another randomly selected hotel auction closes until the eleventh minute when the last hotel auction closes. When a hotel auction closes, sixteen rooms are sold to the sixteen highest bidders at the price of the sixteenth highest bid. If there are less than sixteen bids then the rooms are sold for 0. Buy bids may be submitted at any time while the auction is not closed. Bids may not be withdrawn but may be updated according to the NYSE rule: a standing bid of amount x may be updated to a new value no less than x. Sell bids may not be submitted.

The random closing time in this auction may be a cause of concern. Using a model-free RL strategy does not leave room for closing time prediction, and this may cause the RL agent to perform poorly on or around each possible closing time. This type of auction will need to be analyzed separately with an RL strategy in order to draw any finer conclusions.

4.2.1.3 Entertainment

At the outset of the game each agent owns twelve tickets chosen in the following way: Four tickets of one type on day 1 or 4, four tickets of one type on day 2 or 3, two tickets of one type (different from the bundle of four) on day 1 or 4, and two tickets of one type on day 2 or 3. For each day (excluding the fifth) and type of ticket (Alligator wrestling, Museum, AmusementPark) there is a CDA in which agents may submit buy or sell bids. The CDA is very similar to the one analyzed in chapter 1, except that the agent may change from being a buyer to a seller.

4.2.2 The Clients

The clients are represented by a randomly chosen set of preferences. These preferences are: preferred arrival day (PA), preferred departure day (PD), hotel preference (HP), and entertainment preferences for each of the entertainment tickets (AW, AP, MU). These preferences are used to calculate the client's utility. At the end of the game the auction server calculates the score of the agent by allocating the goods obtained by the agent to its clients in an optimal way.

The optimal allocation is based upon finding the maximum sum of the utilities for each of the clients. For each client with a feasible travel package, the utility, U, is calculated as follows:

$$U = 1000 - TP + HB + EB$$

where TP is the travel penalty, HB is the hotel bonus, and EB is the entertainment bonus. These are defined below:

$$TP = 100 * (|AD - PD| + |AA - PA|)$$

where AA is the actual arrival date and AD is the actual departure date.

$$HB = TT? * HP$$

where $TT?$ is a Boolean variable set to 1 when the client is staying at Tampa Towers and 0 otherwise.

$$EB = AW? * AW + AP? * AP + MU? * MU$$

where $AW?$ $AP?$ and $MU?$ are all boolean variables indicating ownership of those entertainment tickets.

If a client does not have a feasible package then their utility is 0. A feasible package is one in which there is an in-flight and an out-flight with a hotel room of the same type for each night.

4.3 Related Work

Several research reports have been published concerning TAC and the strategies adopted for the competition. The most notable will be outlined here.

SouthamptonTAC [15] "achieved the highest mean score and the lowest standard deviation in the course of the competition's approximately 600 games." The general idea of the SouthamptonTAC strategy was to divide the 12 minute game into three stages: probing (up to minute four), decisive (minutes five to eleven), and finalization (up to minute twelve). These stages were divided up this way because no auction closes within the first four minutes and thus market history may be built up and the most definite purchases made within this time. The decisive stage is when all of the hotel auctions close and the majority of the transactions occur. The final stage is when the the final desperate bidding occurs. For both the flight and hotel auctions predictors are used. In the case of flights, the predictor calculates the average change in price and uses this to categorize the volatility of the auction into one of four categories. In the case of the hotel predictions, fuzzy reasoning is used taking into account factors such as: the ask price, the counterpart hotel ask price, the counter part hotel closing time (if known), the current time, the rate of change of the hotel price, and the rate of change of the counterpart hotel price. The rules are then used to predict the clearing price. Fuzzy sets are also used for the entertainment tickets as in [14] as discussed in the CDA sections. Many of the ideas presented in their paper are specific to the TAC domain, such as the fuzzy set based predictors and flight predictors. It is difficult to foresee a generalized advancement in the autonomous-bidding multi-auction field coming from this paper.

The 006 agent [8] used marginal cost predictions - calculated from historical data and current prices - in order to calculate the optimal amounts of goods to own in a "constraint solver." The constraint solver proposes travel packages for the clients (using marginal costs) that maximize the total utility. Specifically, constraint programming (CP) over finite domains is used. Once the optimal allocation is calculated, using the CP solver, bids are placed with a maximum amount as given by the marginal costs. In the case of the hotel auctions the maximum amount was bid immediately, and in the case of the entertainment auctions, the bid was increased incrementally up to the maximum. Flights that were most likely to be needed were bid for immediately, otherwise they were bid for after five minutes. The likelihood of needing a flight was determined by running the CP solver with a number of different marginal costs. 006 placed seventh overall in the semifinals but did not make the finals in TAC 2001.

AT&T presented a more complex model and price prediction approach to TAC. In their paper [23] they introduce a general boosting-based algorithm for solving the conditional density estimation problem, and show how it was applied to TAC. The conditional density estimation problem is a supervised learning problem in which "the goal is to estimate the entire distribution of a real-valued label given a description of current conditions, typically in the form of a feature vector." It may be interesting to note that some of the features were the opponent types in the market. For TAC, the algorithm is based upon not only predicting hotel prices, but more exactly upon predicting the entire conditional distribution of the hotel closing price given the current knowledge of the agent (ask price, time remaining, ...). They were able to build up their model by analyzing data from previous games. They used a new learning algorithm based on boosting techniques and logistic regression for solving this predicting problem. Their agent was very successful and placed second the the 2001 competition. AT&T's approach can be seen as a rigorous application of model-based learning techniques, quite the opposite of what is proposed in the RL agent.

The RoxyBot [1] approach in the 2000 competition (similar to the 2001 competition) was to use an A-star search for solving the allocation problem and a beam search for solving the bidding problem. The beam search they used was a heuristic based A-star search over pricelines. Pricelines are defined as a data structure that contains succinct information pertaining to the supply and demand of a given auction. This beam search was used for the hotel auctions. Interestingly a ZIP-based algorithm was used for the entertainment ticket auctions. In the 2000 TAC competition RoxyBot's heuristic based approach placed second behind AT&T.

4.4 Reinforcement Learning Agent

TAC offers an interesting environment to test the model free nature of RL strategies. First of all it is much more complex than the CDA, due to the interdependencies of the auctions and random preferences of the clients. This complexity causes difficulty not only for the RL agent but also for the model based predictor agents (such as [23]) who perform on a comparable level to the heuristic based agents (such as [1] and [15]). Also, there is much less relevant data for the RL agent to train on then in the CDA section. This is clear in the entertainment ticket auctions, which are CDAs, where the RL agent does not show signs of learning or improvement. Unfortunately the entertainment ticket auctions have the least impact on the final score.

The agent strategy can be viewed as two separate problems: allocation and bidding. The allocation problem discerns what goods owned go to which clients and what new goods should be purchased. The bidding problem decides how to bid for the desired goods. When the agent is running, these two problems are continuously resolved successively, beginning with the allocation problem. When the allocation problem is solved, the bidding agent has the information it needs (ie goods desired) in order to solve the bidding problem (ie. place the bids).

These two sub problems are analyzed in more detail below.

4.4.1 The Allocation Problem

The allocation problem is the problem of deciding how to allocate the goods one owns to the clients in order to maximize one's utility. In the first TAC (2000) most participants considered the allocation problem unsolved [1]. But ATTAC used a binary linear programming solver to solve the allocation problem correctly more than 99.9 percent of the time [23]. Other successful techniques, such as an $A*$ search, were also used, [1].

The linear programming solution is adapted for the reinforcement learning agent analyzed here. The linear programming solver used is LP-solve. The equation to be maximized is:

$$Utility - Cost$$

where *Utility* is the sum of the utilities for each of the eight clients, and *Cost* is the total cost of all of the agent's transactions. There are 212 constraining equations necessary for ensuring the integrity of the final solution. There are a number of constants used in the constraining equations giving: the number of goods owned of each type, and the current price of each good. There are 544 variables used in the constraining equations, each representing the quantity of a certain good to purchase (or sell in the case of entertainment tickets) on a certain day in order to maximize $Utility - Cost$.

When the linear programming solver runs, taking insignificant amount of time (much less than one second), the total score equation is maximized, thus giving the optimal number of goods to purchase of each kind at the current price.

4.4.2 The Bidding Problem

The problem of how to bid is tackled using RL. The RL strategy we use for this problem used is based upon the inherently decentralized nature of the TAC competition. There is one separate learner for each type of auction; in-flight, out-flight, Tampa towers, Shoreline shanties, Museum tickets, Alligator wresting tickets, and Amusement park tickets.

As in the CDA experiments each of the learners uses Sarsa(λ) with eligibility traces. In addition to this and because of the larger state space a function approximator (CMACS) with tile encoding was used. There are four tilings and 1000 parameters for each of the learners. Again this has not been rigorously optimized.

4.4.3 In-Flight Learner

The flight auction price history is observed for the first three minutes of the game, in which time an estimate of the minimum upper bound, x, on the final price of the flights is predicted.

The state space consists of nine features that include the following:

- In-flights needed for days one, two, three, and four as given by the output from the linear programming solver.

- The minimum upper bound for the price of the in-flights on days one, two, three, and four, as given by the predictor.

- The current time as given in twenty second intervals.

Note that the number of flights currently owned is not included as it has no bearing on the bidding strategy and is considered in the allocation problem.

The action space consists of sixteen actions. One action decides how to bid for each of the four in-flight auctions. For each auction, the agent may bid aggressively or may decide not to bid. If the agent bids aggressively, the bid price will be the current ask price plus 100 monetary units. This ensures that the agent will purchase the flight ticket. There are sixteen possible combinations of these two bid choices for each of the four in-flight auctions.

The reward, R, for the in-flight learner is the following:

$$R = PB * (THP) - Penalty - UP * (UFP)$$

where the package bonus, $PB = 70$, the unused goods penalty, $UP = 0.45$, the unused flight price (UFP) is the total price spent on unused flights, THP is the total number of hotel packages, and the Penalty is the penalty given for not obtaining in-flights for the client's preferred arrival date. The reward is received at the conclusion of the game.

4.4.4 Out-Flight Learner

The out-flight learner is similar to the in-flight learner except that instead of considering days one, two, three, and four, the learner considers days two, three, four, and five. This is because there are no out-flights on day one.

Both of the flight learners participate in relatively simple auctions, but they're performance is very difficult to analyze. This is because the performance of the entire agent depends more on the combination of goods, and not on the flights alone. One could imagine the flight learners learning a policy that would refrain from bidding while the flight price isn't expected to fluctuate, and bid otherwise. Only this would require much training or specific changes to the action space that would allow for prolonged waiting actions.

4.4.5 Tampa Towers Learner

The state space for the Tampa Towers learner consists of seventeen feature variables:

- The number of Tampa Towers desired for each of the four nights as given by the linear programming solver.

- The number of Shoreline shanties desired for each of the four nights as given by the linear programming solver.

- The current ask price for each day for the Tampa Towers hotels. If the day is two or three an absolute constant value of 100 monetary units are added to the price. This is to bias the linear programming solver to choose hotels on the first and last day. The demand for hotels on those days is on average less than days 2 and 3 and thus their price is lower.

- The current ask price for each day for the Shoreline Shanties hotels. If the day is two or three an absolute constant value of 50 monetary units are added to the price.

There are sixteen actions for the Tampa Towers learner. They are combinations of bidding aggressively and weakly for each of the four nights. Once an action is taken the agent bids in all four auctions. Bidding aggressively increases the current bid by 100 monetary units, and bidding weakly does not alter the current bid.

The reward, R for the Tampa Towers learner obtained at the end of the game and is given by:

$$R = PB * TP + HB - UP * UTP$$

where HB (Hotel Bonus) is the utility gained (including costs) by using Tampa Towers hotels instead of Shoreline Shanties, TP is the total number of towers packages. and UTP is the Unused Towers Price, which is the price paid for towers hotels that are not used in any client packages.

4.4.6 Shoreline Shanties Learner

The Shoreline Shanties learner is similar to the Tampa Towers learner, except that in the reward calculation the hotel bonus is not included.

Formal experimentation and optimization of the RL parameters of the hotel learners would need to be undertaken before drawing final conclusions. Intuitively the hotel markets are more simple than the CDA markets except that there is the added uncertainty of the closing times. Regardless of the outcome of this RL agent in TAC in 2001, one might suspect that there is a strong possibility of the hotel learners being successful using RL with more analysis in the future.

4.4.7 Entertainment Learners

The entertainment learners operate in a CDA environment, so they are modeled after the success of the CDA agents in chapter one. But, there are a few important differences. First of all the agent may buy or sell depending on the state of the market. Secondly, the supply and demand ratios are not fixed; this offers a more advanced situation than that in the CDA chapter (one discussed for future work). Finally, there are very few trades and the trading partners alter after every game. This also offers a more advanced situations, and the lack of data may prove very difficult to overcome without resorting to using the data of past games or off-line training. The three entertainment learners (Museum, Amusement Park, and Alligator Wrestling) all operate equivalently. They each have a state space of nine features. For a particular entertainment learner the features are:

- The number of tickets needed for each day (as given by the LP solver)
- The current ask price (or buy price, in which case the price is negative) for each day
- The current time segment

Similar to the other learners, there are sixteen actions for each entertainment learner. The only difference here is that an agent may sell instead of buy. This is taken into account in the linear programming solver, which will return a negative amount of tickets needed if it is beneficial to sell.

As can be noted, the state space for the entertainment auctions is slightly different than that of the CDA in chapter 1. This is because the linear programming solver takes into account the difference in reservation value and current ask price and this isn't specifically in the state space. The reservation values cannot be directly in the entertainment learners state space because which reservation values that need to be used depend on the hotel rooms and on the other entertainment tickets owned or auctioned for. In the future this will need to be addressed and more information may be given to the entertainment auction learner (although this will drastically increase the state space).

The reward for the entertainment ticket learners is the total utility minus the total cost, which is the profit and is given at the end of the game. This reward is exactly the reward given to the agent in the CDA section.

4.5 Results and Discussion

The TAC environment is much like the CDA except that it is much more complicated. Three important issues arise in TAC that are not seen in the CDA. Namely, the combinatorial explosion of bidding possibilities, the divided but dependent reward scheme for the multiple learners, and cooperation needed from the learners due to the interdependencies of the auctions. The combinatorial explosion of bidding problems greatly inhibits the exploration performance, and causes a need for even more data and learning time. The cooperation needed from the learners is implicit in the RL setup and is forced upon the learners through the divided reward structure.

In the RL strategy for TAC a decentralized view was taken, albeit before the individual RL learners were proved capable. A centralized view could also be explored. More precisely, having a main center of guidance that is aware of current states and actions of all of the RL bidding agents may prove worthy of adoption for future TAC participants. On the other hand an even more decentralized view could be used. That is, more information could be given to the learners in attempt to minimize their uncertainty. Unfortunately this would likely increase their state space greatly, thus adding to the problem of lack of training and data.

In the end it seems that the agent did not perform very well and did not show signs of learning - it placed twenty-first in the preliminary round and sixteenth in the seeding round. This could be from a number of elements of the agent, most notably the lack of data and the non-deterministic nature of TAC. This problem of large state space and lack of data may need to be tackled by excess off-line training against dummy or simple heuristic agents, or in the worst case analyzing data from previous games and using it for off-line learning. The interdependencies of the auctions in TAC is what causes TAC to be a parent, or more complex problem than the CDA. The fact that TAC is so highly uncertain may bode well for model free agents such as RL, because models may be very difficult to build and may be inaccurate in these situations.

A way to attempt to create a successful RL based agent for the TAC market could be to first create RL agents that could compete in the hotel auctions and flight auctions. The entertainment auctions are CDAs much like from chapter one, thus RL strategies have proved to be competitive for these auctions. Using these separate RL agents, the remaining problem would be to devise a suitable means for dividing the reward and for handling the auction interdependencies.

There are a number of improvements and changes that could be made to the current RL agent for TAC. The action space could be enlarged, and the problem of large (or even continuous action spaces) could be tackled. The reward scheme could also be varied to allow greater or less cooperation between agents, and more importantly to allow for more emphasis to be placed on certain clients or travel packages. And finally the data from previous games could be used as an off-line learning tool for the RL agent in hopes of overcoming the data shortage problem.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Our thesis has illustrated the use of reinforcement learning to produce automated bidding agents in electronic markets. We presented two case studies, involving the continuous double auction and the trading agents competition.

In the continuous double auction, we designed a very simple RL agent and compared its performance empirically against several well-known strategies. The empirical results suggest that reinforcement learning was quite successful at producing good strategies. The most surprising result is that the RL agents consistently increased the market efficiency, in all the markets we studied. The RL agents also achieved the best surplus in almost all market mixes we studied. We attribute the success of the RL agents to the fact that they can take advantage of the temporal nature of the auction environment. The other agents we tested against cannot exploit trends and periodicities in the market. We have to note, though that neither of the agents used in these experiments has been optimized in any way. Such an optimization would be necessary in order to make definite claims about the superiority of any given strategy. We also want to study the robustness of the RL agents with respect to different settings of the learning parameters.

One immediate direction for future research is to investigate the behavior of RL agents against more sophisticated agents, such as [27], [14], [22]. In particular, we want to study the behavior of RL agents in the presence of other learning agents, which also improve over time. It would also be important to evaluate the behavior of RL agents trained using one market mix against a different market mix. Since the

RL agents we used do not use any opponent-specific, we hope that they would be robust with respect to moderate changes in the market mix.

In our second case study we focused on the trading agents competition, an emerging benchmark for research on automated trading agents. We reported on the agent we used in the 2001 competition. Unfortunately, reinforcement learning was not as successful in this case. We attribute this result to the lack of training instances. compared to the size of the state space, and the added uncertainty due to the interdependencies of the three types of auctions. However, our result does not warrant putting the TAC environment beyond the capabilities of RL agents. In fact, continuous double auctions are a sub-problem of TAC, and we have seen that RL can be very successful for such problems. We have demonstrated in this thesis that RL can successfully compete in markets such as the CDA, which leads one to believe that RL could successfully compete in other independent auctions. What remains is to tackle the important and related problem of communication between RL agents competing in interdependent auctions. We anticipate that solving this problem would allow success in the TAC environment as well.

The success of RL in auctions such as the CDA is important not only from a mechanism design or experimental economics point of view, but also from a reinforcement learning point of view. The non-Markovian nature of the CDA may cause one to think that using an RL algorithm would cause the agent's policy to converge to a non-profitable bidding strategy or to not converge at all. RL algorithms have been shown to converge in systems that obey the Markov property, but there are no proofs about convergence in non-Markovian systems. However, our experimental results demonstrate that in this particular kind of environment, the policies of the RL agents do converge, and yield profitable bidding strategies. This result adds hope that convergence of RL algorithms for the CDA, or other particular kinds of multi-agent systems, may be proved mathematically in the future.

BIBLIOGRAPHY

[1] A.Greenwald, and J.Boyan. Bid determination in simultaneous auctions. In *Proceedings of the third ACM Conference on Electronic Commerce* (2001).

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

[3] Cliff, D. Minimal-intelligence agents for bargaining behaviors in market-based environments. Tech. Rep. HPL-97-91, Hewlett Packard Labs, 1997.

[4] C.Preist, A.Byde, and C.Bartolini. Economic dynamics of agents in multiple auctions. In *Proceedings of the fifth international conference on Autonomous agents 01* (2001).

[5] C.Preist, and van Tol, M. Adaptive agents in a persistent shout double auction. In *Proceedings of ICE-98* (1998).

[6] Das, R., Hanson, J. E., Kephart, J. O., and Tesauro, G. Agent-human interactions in the continuous double auction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (2001), Morgan-Kaufmann, pp. ?–?

[7] D.Zeng, and K.Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies 48* (1998), 125–141.

[8] E.Aurell, M.Boman, M.Carlsson, J.Eriksson, N.Finne, S.Janson, P.Kreuger, and L.Rasmusson. A constraint programming agent for automated trading. *ERCIM News 51* (2002).

[9] E.Gimenez-Funes, L.Godo, and J.Rodriguez-Aguilar. Designing bidding strategies for trading agents in electronic auctions. In *Proceedings of the Third International Conference on Multi-Agent Systems* (1998).

[10] E.Oliveira, J.Fonseca, and N.Jennings. Learning to be competitive in the market. In *Proceedings of the AAAI Workshope on Negotiation: Setting Conflicts and Identifying Opportunities* (2000).

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

[12] G.Tesauro, and J.Kephart. Pricing in agent economies using multi-agent q-learning. In *Proceedings of Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (1999).

[13] G.Tesauro, and J.Kephart. Pseudo-convergent q-learning by competitive pricebots. In *Proceedings of the Seventeenth International Conference on Machine Learning* (2000).

[14] He, M., Leung, H., and Jennings, N. R. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. *IEEE Transaction on Knowledge and Data Engineering* (To appear).

[15] He, M., and N.Jennings. Southamptontac: Designing a successful trading agent. In *Fifteenth European Conference on Artificial Intelligence* (200).

[16] J.Hu, and M.Wellman. Online learning about other agents in a dynamic multiagent system. In *Proceedings of the Second International Conference on Autonomous Agents* (1998).

[17] J.Oliver. A machine learning approach to automated negotiation and prospects for electronic commerce. *Journal of Management Information Systems 13* (1996), ?–?

[18] J.Vidal, and Durfee. Agents learning about agents: A framework and analysis. In *Proceedings of the Second International Conference on Autonous Agents* (1998).

[19] Kephart, J., and Greenwald, A. Shopbot economics. *Autonomous Agents and Multi-Agent Systems: Special Issue on Game-theoretic and Decision-theoretic Agents* (To appear).

[20] O.Buffet, A.Dutech, and F.Charpillet. Incremental reinforcement learning for designing multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents* (2001).

[21] P.Anthony, W.Hall, V.Dang, and N.Jennings. Autonomous agents for participating in multiple online auctions. In *Proceedings of the IJCAI Workshop on E-Business and the Intelligent Web* (2001).

[22] Park, S., Durfee, E. H., and Birmingham, W. P. An adaptive agent bidding strategy based on stochastic modeling. In *Proceedings of the Third International Conference on Autonomous Agents* (1999), pp. 147–153.

[23] R.Schapire, P.Stone, D.McAllester, M.Littman, and J.Csirik. Modeling auction price uncertainty using boosting-based conditional density estimation. In *Machine Learning: Proceedings of the Nineteenth International Conference* (2002).

[24] Rust, J., Miller, J., and Palmer, R. Characterizing effective trading strategies: Insights from the computerized double auction tournament. *Journal of Economic Dynamics and Control 18* (1994), 61–96.

[25] Stone, P., and Greenwald, A. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research: Special Issue on Dynamic Pricing* (To appear).

[26] Sutton, R. S., and Barto, A. G. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[27] Tesauro, G., and Bredin, J. L. Strategic sequential bidding in auctions using dynamic programming. In *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)* (2002).

[28] Tesauro, G., and Das, R. High-performance bidding agents for the continuous double auction. In *Proceedings of the IJCAI Workshop on Economic Agents, Models, and Mechanisms.* (2001), pp. ?–?

[29] V.L.Smith. An experimental study competitive market behavior. *Journal of Economic Dynamics and Control 70* (1962), 111–137.

[30] W.Vickrey. Auctions and bidding games. In *Recent Advances in Game Theory* (1962).