Reputation-Oriented Reinforcement Learning Strategies for Economically-Motivated Agents in Electronic Market Environments

by

Thomas Thanh Tran

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

In this thesis, we propose a market model and learning algorithms for buying and selling agents in electronic marketplaces. We take into account the fact that multiple selling agents may offer the same good with different qualities, and that selling agents may alter the quality of their goods. We also consider the possible existence of dishonest selling agents in the market. In our approach, buying agents learn to maximize their expected value of goods using reinforcement learning. In addition, they model and exploit the reputation of selling agents to avoid interaction with the disreputable ones, and therefore to reduce the risk of purchasing low value goods. Our selling agents learn to maximize their expected profits by using reinforcement learning to adjust product prices, and also by altering product quality to provide more customized value to their goods. We experimentally evaluate our model on both microscopic and macroscopic levels. On the micro level, we examine the individual benefit of agents, in particular their level of satisfaction. Our experimental results confirm that in both modest and large-sized marketplaces, buying and selling agents following our proposed algorithms achieve better satisfaction than buying and selling agents who only use reinforcement learning. On the macro level, we study how a marketplace populated with our buying and selling agents would behave as a whole. Our results show that such a marketplace can reach an equilibrium state where the agent population remains stable and that this equilibrium is beneficial for the participant agents. The market model and learning algorithms presented in this thesis can therefore be used in designing desirable market environments and effective economically-motivated agents for e-commerce applications.

Acknowledgements

There are many people whom I would like to thank for their support, guidance and encouragement during my graduate study years at the University of Waterloo.

First and foremost, I would like to offer my warmest thanks to my supervisor, Dr. Robin Cohen, for her countless stimulating discussions and many hours spent reading my work. She has kindled in me the desire to do research from the first day, encouraged me to achieve my best ability, and guided me through the necessary research path. Without her, this thesis would not have been completed. I am also thankful to Dr. Cohen for her infinite support in almost every matter, including being available for me on a daily basis for not only intellectual but also everyday life issues, training me how to write good research papers, exposing me to the research communities of my fields, and teaching me how to develop a lecture via the courses for which I served as her teaching assistant. She has indeed done an excellent job in preparing me for my career as a teacher and researcher. In addition, forever kept in my heart is the good care that she has taken of me both academically and financially throughout the years of my graduate studies. From her I have learned many good things, but my most favourite ones are her responsible mind for duties and her caring heart for people.

Next, I would like to thank my other thesis committee members, Peter van Beek, Dale Schuurmans, Mohamed Kamel, and my external examiner, Julita Vassileva, for their helpful advice and valuable comments on my work.

I also wish to express my special appreciation to the Natural Sciences and Engineering Research Council of Canada (NSERC) for awarding me the Postgraduate Scholarships (PGS A and B), which have been a major financial support for me for my four graduate school years.

I am thankful to Relu Patrascu, Vlado Keselj, Michael Fleming, Fletcher Lu, Fuchun Peng, and Zhenmei Gu (members of the Artificial Intelligence Research Group), as well as Paul Kates and Anne Pidduck (Lecturers) for their interesting discussions and friendship.

I thank the departmental staff, especially Wendy Rush, the Administrative Co-ordinator, and Jessica Miranda, the Receptionist, for their precious help in various ways that made my student life run smoothly.

Many special thanks are reserved for my parents as well as my brothers and sisters, especially my mother and my brother Phuoc, for the moral support that they have provided me remotely from Vietnam.

Finally but most of all, I would like to thank my wife Michelle and my daughter Grace for their love, patience, support, understanding and encouragement. Michelle and Grace have also gone through the entire graduate experience with me. With out their continuing love and support I would not have made it.

Thomas Thanh Tran

University of Waterloo

Fall 2003

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we introduce the motivating problem for this thesis and discuss the possible value of solving the problem. We then present an overview of the thesis followed by its organization.

1.1 The Motivating Problem

Consider an electronic market environment populated by buying and selling software agents, respectively. The buying agents are interested in purchasing some particular good which is offered for sale by the selling agents. Each buying agent is free to communicate with the selling agents in the market, in order to get price bids for the good. Buying and selling agents are business-minded (or self-interested) agents whose goal is to maximize their own benefit. The task faced by a buying agent is to choose from several selling agents a suitable one such that purchasing the good from that selling agent would maximize its expected value of the good. The problem encountered by a selling agent is to decide how much to bid for the

good, once requested by a buyer. The selling agent wants to bid high in order to maximize its profit, but it does not want to bid so high that the buying agent may choose to purchase from another selling agent.

According to economics theory [4], if all the goods offered by different selling agents in the market are identical, then the prices asked by the selling agents will drop down to the selling agents' marginal cost[1]. Since the only difference between the goods sold by various selling agents is the price, the buying agents will purchase from the selling agent that offers the good at the lowest price. Other selling agents therefore have to lower their prices to beat or at least to match the current lowest price. This price competition will go on until the selling agents' prices are dropped as low as their production costs. Selling agents with higher production costs will have to leave the market because they are no longer able to compete against those with lower production costs. Finally, the price will reach its minimum value. Remaining in the market at this point are only those selling agents that have the same lowest production cost, because they are able to sell the good at this minimum price. This final price is the marginal cost of the remaining selling agents in the market. Thus, in a marketplace where sellers offer completely identical goods and have the same fixed marginal cost, the price of the good will converge to an equilibrium which is the sellers' marginal cost.

Consider a more realistic and also more interesting marketplace in which the goods offered by various selling agents are not necessarily identical, but may have different qualities as ascribed by buying agents in the market. Thus, different buying agents may have different personal preferences over the goods sold by the selling agents. In this situation, simply choosing the lowest price is no longer the

[1]Marginal cost is defined as the increase in total cost when one additional unit of goods is produced (page 164 of [4]).

best strategy for a buying agent. In fact, a buying agent may be willing to pay more for the good offered by a selling agent that has a better quality and therefore better meets its demanded product value. In addition, as with any real marketplace, let us assume that agents can freely enter or leave our marketplace (open market), and that a selling agent may alter the price and also the quality of its goods, in order to meet the buying agents' specific needs (dynamic market). Furthermore, assume that a buying agent can only examine the quality of the good it purchases after it receives the good from the selected selling agent (uncertain market), and that there may be dishonest selling agents in the environment (untrusted market). In such an open, dynamic, uncertain, and untrusted marketplace, a buying agent must learn to avoid interaction with the dishonest selling agents as well as to purchase the goods that are of the greatest value to it. Also, a selling agent must learn to not only adjust the price but also the quality of its goods in order to maximize its profit. How to design such buying and selling agents is the motivating problem for this thesis. More succinctly, the problem we are interested in addressing is how to design feasible and effective learning algorithms that guide the behaviours of buying and selling agents participating in open, dynamic, uncertain and untrusted electronic market environments. Let us now discuss the possible value of solving this problem.

1.2 Possible Value

We believe that being able to design effective learning agents participating in multi-agent electronic market environments is significantly useful for today's world. With the Internet proving to be cost effective and security concerns being addressed by technological advances, electronic commerce (e-commerce) is now a steadily growing

field and a viable option for many organizations [40]. The current trend of research and development in e-commerce is towards building market-based multi-agent systems populated with intelligent software agents that perform business transactions on behalf of their human owners [17, 18, 37]. A number of such systems have been developed with certain advantages. However, the agents of these systems often suffer from shortcomings such as not being autonomous [3, 14, 29], not having learning capability [3, 14, 29, 9, 10], incurring costly computation [77, 78], or being incapable of dealing with dishonest agents[2] [3, 14, 29, 9, 10, 77, 78]. Consequently, the design of feasible and effective learning agents that are autonomous and able to learn to adapt themselves to market environments that are open, dynamic, uncertain and untrusted would be both a key challenge for research and an important contribution for e-commerce applications.

Being dynamic and uncertain are two natural characteristics of market-based multi-agent systems. In these systems, information such as prices or product quality can be altered any time. Also, a buying agent does not know in advance from which selling agent it will be best for it to purchase. Similarly, a selling agent does not know for sure if it would be successful in making a sale to a particular buying agent. In addition, it is difficult to build an agent with all the prior knowledge it needs about other agents and its environment. The reason is that agents may change their behaviours over time, which in turn may affect environmental conditions. Moreover, agents may be built by different designers or at different times, and therefore may not have prior knowledge about one another. The only feasible way to cope with this difficulty is to equip agents with learning capabilities which allow them to

[2]Detailed descriptions of these related systems are provided in Section 2.1.2 (of Chapter 2). Contrast and experimental comparison of our work with these systems is discussed in Section 5.1 (of Chapter 5).

acquire necessary knowledge through observations of and interactions with other agents in the system. In this thesis, we propose learning algorithms for buying and selling agents that take into account the dynamic and uncertain nature of the agents' environment. We believe that the techniques presented in this thesis are useful not only in market-based multi-agent systems, but also in general multi-agent systems where agents cannot be built with all the prior knowledge they need about other agents and their environments.

Another characteristic of a market-based multi-agent system is the possible existence of dishonest agents. For instance, some selling agents may decide to cheat the buying agents to quickly increase their profits. Obviously, the honest agents in the market should have some way to deal with the dishonest ones and therefore to protect themselves from these agents. In particular, they should be able to learn to detect the dishonest agents and therefore to avoid interactions with them. Furthermore, the honest agents in the market should somehow be able to penalize the dishonest ones so that together they may cast the dishonest agents out of their environment. The buying algorithm we propose in this thesis introduces the modelling of reputation as an effective technique for dealing with dishonest selling agents. We believe that this technique may also be applicable in other multi-agent systems where there may be unreliable, antisocial or dishonest agents.

Agents exist and operate in some environment which is both physical and computational, and which may be closed or open. The increasing interconnection and networking of computers as well as the advanced development of agent communication languages and interaction protocols are making it possible to build open multi-agent systems where agents can interact with new agents coming from other systems. In such open multi-agent systems, agents should have suitable strategies to interact with new agents who may come and go as time goes on. The buying

algorithm we propose in this thesis takes this into account. In particular, the attitude taken by our agents is to neither favour nor be prejudiced against a new agent. Through observations of and interactions with the new agent, the agent will learn to raise (or lower) its defence level, depending on whether trust has been gained (or lost)[3]. This technique therefore can be used in any open multi-agent system where permanent resident agents may have to interact with agents that come and go.

1.3 Overview

The main contribution of this thesis is the proposal of effective learning algorithms for buying and selling agents in electronic market environments. Briefly speaking, our buying agents make use of the combination of reinforcement learning and reputation modelling, while our selling agents use reinforcement learning with the addition of product quality adjustment.

Since reinforcement learning explicitly considers the problem of an agent learning from interaction with an uncertain environment to achieve a goal, we believe that it should serve as a useful learning method for trading agents in market environments[4]. In addition, we believe that in a market environment, reputation of selling agents is an important factor that buying agents should model to exploit. If a buying agent, by modelling the reputation of selling agents, can distinguish good selling agents from bad ones (including the dishonest ones), the buying agent will be able to focus its business on the those selling agents with whom it has established a certain degree of trust, and to avoid interaction with those it considers untrustworthy. Clearly, this strategy should enhance the buying agent's chance of

[3]Details of this technique are described in Section 3.2.1.

[4]A detailed justification for our choice of reinforcement learning is offered in Section 5.2.2.

purchasing satisfactory goods, and reduce its risk of buying unsatisfactory ones, especially in a dynamic, uncertain and untrusted market environment. Reputation modelling also provides a means for handling the case of new agents in an open marketplace: An entirely new model of reputation can be initiated for a new selling agent by a buying agent in the market (or for all selling agents in the market by a new buying agent). This model reflects the level of reputation that the buying agent initially assigns for the selling agent. The initial reputation level will be updated over time by the buying agent, depending on whether or not trust has been established through interaction with the selling agent. Consequently, in our approach we suggest that buying agents model the reputation of selling agents. The challenge is then to determine how exactly to model the reputation of selling agents in a market, e.g., how to initiate the modelling, how to update it, and how to make use of it to influence purchase decisions of buying agents. Also, we need to specify how to integrate the modelling of reputation with reinforcement learning in order to improve the buying agents' satisfaction for goods purchased.

In any marketplace, a selling agent will be successful in making sales if the value of its goods satisfies the buying agents. Since quality and price are the two most important factors that compose the value of goods, the selling agent should set these factors appropriately. Hence, in our approach selling agents learn to set the price and also to adjust the quality of their goods, in order to maximize their profit. The challenges are how to combine the adjustments of the two factors in the selling algorithm, how to detect if any of the factors needs to be adjusted, and how often to adjust them.

In conjunction with the design of the algorithms for buying and selling agents, we are also required to model an electronic marketplace itself. In particular, we need to consider the open, dynamic, uncertain and untrusted characteristics of a

real marketplace in our model, as well as to design an appropriate set of rules (i.e., market mechanism) that specifies how buying and selling agents would interact with one another in the market to exchange goods. The challenge is how to come up with a feasible agent market model which can be used for e-commerce applications.

In this thesis, we experimentally measure the value of our proposed algorithms on both the microscopic and macroscopic levels. On the micro level, we are interested in examining the individual satisfaction of buying and selling agents. Our results confirm that buying and selling agents of our approach obtain better satisfaction than buying and also selling agents that only use reinforcement learning (with the buying agents not modelling the selling agents' reputation, and the selling agents not considering adjusting the quality of their goods)[5]. On the macro level, we study how a market populated by our buying and selling agents would behave as a whole. The experimental results show that such a market can reach an equilibrium state where the agent population remains stable and that such an equilibrium is beneficial for the participant agents (in the sense that there are fewer competitors for selling agents, and there are no selling agents offering goods under the demanded value of buying agents). We also contrast and experimentally compare our agents with those of related agent market models. The results show that our agents are able to achieve better performance in terms of satisfaction and computational time.

[5]For buying agents, better satisfaction means receiving more goods with high value. For selling agents, better satisfaction means making higher profit.

1.4 Organization

The thesis is organized as follows: Chapter 2 covers relevant research which has been done in related areas. Chapter 3 presents our proposed learning algorithms for buying and selling agents in electronic market environments. In addition, it describes an agent market model which is feasible and suitable for e-commerce applications. It also theoretically explores the question of how much our proposed buying agents can be harmed by dishonest selling agents, and demonstrates that it is possible for a proposed buying agent to avoid infinite harm caused by a dishonest selling agent. Chapter 4 experimentally evaluates our model by simulating electronic marketplaces populated with buying and selling agents. It investigates the micro behaviours of participant agents in both modest and large-sized marketplaces, as well as the macro behaviours of the marketplace as a whole. Chapter 5 offers a detailed discussion of the value of our model. In particular, it contrasts our model with other related e-commerce agent models, and experimentally compares the performance of our buying and selling agents with those proposed in the most related research. It also provides more detailed defense for specific design decisions made for the model and discusses in greater depth the inherent value of the framework for reputation modelling used within the model. Chapter 6 concludes the thesis with its contributions and possible future research directions.

Chapter 2

Background

This chapter reviews relevant research which has been done in related areas, and which is used to motivate various design decisions in this thesis. The first section introduces several agent models that have been developed for e-commerce. Then reinforcement learning methods are reviewed, followed by their applications in the area of agent and multi-agent systems. After that, reputation modelling is visited, including the description of some typical reputation mechanisms. Finally, the chapter ends with a summary of the main issues discussed.

2.1 Agent Models for E-Commerce

Before we introduce some representative agent models for e-commerce, let us start with a discussion on what we mean by *agents*, *multi-agent systems*, and *e-commerce*.

2.1.1 Agents, Multi-Agent Systems, and E-Commerce

Agents

The term *software agent* (or just *agent* for short) has been widely used in many fields of computer science, but surprisingly enough, this term has not yet had a universally accepted definition. For the purpose of this thesis, *we define an agent as a computer program that can perceive, reason, act, communicate and coordinate.* In addition, an agent acts on behalf of a user, carrying out actions that meet the user's goals and preferences. A selling agent in a market environment (making sales on behalf of a user), for example, should be able to know if any buying agent (making purchases on behalf of a user) has requested some good and whether it has that good for sale, then to reason at which price it would like to sell the good in order to make profit, and finally communicate with the buying agent to make an offer. *An agent is said to be intelligent if it is able to respond in a timely fashion to environmental changes (reactivity), to interact with other agents and possibly humans (sociality), and to take the initiative (pro-activeness).*

Agents and objects as defined in the object-oriented programming model have some similarities because objects are defined as computational entities that encapsulate some internal state, are able to perform actions (or methods) on that state, and communicate by message passing. However, agents and objects also have significant differences which constitute the novel concept of agents:

- Agents are *autonomous* or *semi-autonomous* where autonomy refers to the agents' ability to act without the intervention of humans or other systems. In particular, agents decide for themselves whether or not to perform an action upon request from other agents, while a public method of an object can be executed by other objects any time they wish.

- Agents are capable of reactive, pro-active, and social behaviours, while the standard object model has nothing to say about how to build systems that integrate these types of behaviours.
- Each agent is considered to have its own thread of control, while in the standard object model there is a single thread of control for the whole system.

The definitions of agents and intelligent agents presented here are based on [27] and Chapter 1 of [81], respectively. Other good references for the theme *agents* with different inherent theories and definitions are [54] and [82].

Multi-Agent Systems

Essentially, agents exist and operate in some environment, which may be closed or open, where in contrast to *closed*, the term *open* refers to the fact that new or existing agents can freely enter or leave the environment. There are circumstances where an agent exists and operates usefully by itself in an environment. However, increasing interconnection and networking has made such situations very rare, resulting in cases where agents usually interact with one another. *A multi-agent system is therefore defined as a system (or environment) composed of multiple interacting agents.*

A well-designed multi-agent system is one in which agents can operate effectively and interact with each other productively. The system should provide some computational infrastructure to allow agents' interaction to take place. Such infrastructure includes protocols for agents to communicate and protocols for agents to interact.

Communication protocols enable agents to exchange and to understand messages. Interaction protocols enable agents to have conversations, which, for the

purposes of multi-agent systems, are structured exchanges of messages. For example, a communication protocol may specify the following message types as valid messages to be exchanged between agents:

- Propose a course of action.
- Accept a course of action.
- Reject a course of action.
- Counter-propose a course of action, etc.

Based on these message types, an interaction protocol may be structured as follows for two agents to negotiate with each other:

- Agent 1 proposes a course of action to Agent 2.
- Agent 2 evaluates the proposal and sends a counter-proposal to Agent 1.
- Agent 1 accepts the counter-proposal.

We adopt the definition of multi-agent systems from [81] (Glossary Section). Our brief introduction to communication and interaction protocols is based on Chapter 2 of [81].

E-Commerce

Electronic Commerce (e-commerce) is the buying and selling of goods and services on the Internet. The fact that companies have been increasingly using Internet

technologies to do business makes e-commerce a rapidly growing field and a multi-billion dollar segment of the world economy[1]. Traditional telephone calls and paper-based procedures have been replaced by electronic data interchange, e-mail, shared databases, digital image processing, bar coding and interactive software for product design, marketing, ordering, delivery, payment and customer support. Digital links can be established between businesses, allowing them to bypass middlemen and inefficient multi-layered procedures. Handling transactions electronically can significantly reduce transaction costs and delivery time, especially for those goods that are purely digital such as software, images, videos and text products.

E-commerce has several components, including *(i)* automatic ordering, contracting and procurement, *(ii)* electronic order-tracking services, *(iii)* automatic billing and payment services, *(iv)* electronic fund transfer, *(v)* interactive businesses and financial transactions, *(vi)* electronic advertisements, and *(vii)* data mining of consumer information for customer profiling. Agent technologies can be applied to any of these areas where flexible autonomous (reactive, pro-active, and social) behaviours are desirable. In particular, the following factors should be considered in order to determine to what extent agent technologies are appropriate for an area of e-commerce:

- The time or money that can be saved when certain processes are automated (e.g., comparing products from multiple sellers).
- The level of difficulty in expressing preferences (e.g., shopping for a gift).
- The degree of risk when an agent is making a transaction decision (e.g., purchasing a book or making stock market buying and selling decisions).

[1]See the Preface of [37].

In general, the more time or money that can be saved through automation of processes, the less difficult to express preferences, and the lower degree of risk in agents' making transaction decisions, the more appropriate it is to employ agent technologies in e-commerce.

Our definition of e-commerce (more precisely, Web-based e-commerce) is adopted from [17]. Other useful references are Chapter 6 of [37] and Chapter 10 of [40].

2.1.2 E-Commerce Agent Models

In this subsection, we briefly review the relevant e-commerce agent models that are related to our work. These models focus mainly on the first component of e-commerce mentioned above, namely automatic ordering, contracting and procurement. We also discuss the relative advantages and disadvantages of these models.

BargainFinder and Jango

BargainFinder [3] of Andersen Consulting is a shopping agent that assists customers in deciding which merchant to deal with. The algorithm underlying BargainFinder is rather straightforward. Given a specific product, BargainFinder will request the product price from a list of different merchant Web sites, and display these prices to the customer. It is then up to the customer to compare the prices and choose the appropriate merchant. A drawback encountered by BargainFinder is the *merchant blocking problem*: Some online merchants are not interested in competing on price alone, because their products may be offered with some value-added services, which are bypassed by BargainFinder. As the result, these merchants decide to block all price requests originated from BargainFinder. Nevertheless, a number

of little-known merchants, who are willing to compete on price, request Andersen Consulting to include them in BargainFinder's list of merchants.

Jango [14, 29] is another shopping agent, which can be viewed as an advanced BargainFinder. Once a specific product has been identified by a customer, Jango simultaneously queries multiple online merchants (from a list maintained by NetBot Inc.) for product availability, price, and other related information such as important product features. These results are then displayed to the customer, who will have to make the decision about which merchant to select. Jango cleverly gets around the merchant blocking problem by making product requests originate from the customer's Web browser, instead of a central site as in the case of BargainFinder. As the result, product requests made by Jango appear to online merchants as if they were made by real customers and are therefore not blocked by the merchants.

Although BargainFinder and Jango shopping agents provide customers with useful information for merchant comparison, at least three shortcomings may be identified:

- BargainFinder and Jango are not fully autonomous. In particular, they leave the task of analyzing the resultant information and selecting appropriate merchants completely for customers.
- The algorithms underlying these agents' operations do not capture information on product quality or value added services, which is of great importance for customers to decide which merchants to select.
- These agents are not equipped with any learning capabilities to help customers to improve their decision making in the future.

Kasbah

Kasbah [9] is another interesting agent model, designed by the MIT Media Lab as a multi-agent electronic marketplace suitable for user-to-user transactions. Specifically, it is a Web-based system where users create autonomous buying and selling agents that buy and sell goods on their behalf.

The Kasbah marketplace's job is to facilitate interaction between agents. It ensures that participant agents speak a common language, and matches up agents interested in selling or buying the same kind of items. When a selling agent is created, the Kasbah marketplace asks what the agent is interested in selling. The marketplace then sends the agent a list of all the potential buyers for that particular item. It also sends messages to all of these potential buyers, informing them of the existence of the new selling agents. Similarly, when the selling agent leaves the market, the marketplace notifies all of its potential buyers. The same process happens when a buying agent is created.

A user who wants to sell (or buy) a good can create a selling (or buying) agent, give it some strategic direction, and launch it into the Kasbah marketplace. The user directs the agent's behaviours by setting three parameters, namely *the desired price*, *the lowest (or highest) acceptable price*, and *the desired date to sell (or buy) the good by*. These parameters define the agent's goal: Ideally, the agent aims to sell (or buy) the good at the desired price. However, the agent may accept the price that is as low (or high) as the lowest (or highest) acceptable price, if that is what it takes to attract buyers' (or sellers') interest within the desired time frame. Once released into the marketplace, the agent pro-actively seeks out suitable buyers (or sellers), and negotiates with them on the user's behalf in order to make the "best possible deal". The user controls the agent's negotiation strategy by initially specifying a

negotiation function. The selling (or buying) agent uses this function to lower (or raise) the asking price over its given time frame. Kasbah supports three types of negotiation functions, namely linear, quadratic, and exponential, corresponding to the anxious, cool-headed, and frugal negotiation strategies, respectively.

The main advantage of Kasbah is that its agents are autonomous in making decisions, thus freeing users from having to find and negotiate with buyers and sellers. However, as admitted in [9], Kasbah's agents are not very smart as they do not make use of any AI learning techniques.

Recursive Agent Model

Vidal and Durfee [77, 78] propose a recursive agent model for an information economy such as the University of Michigan Digital Library. They divide agents into different classes corresponding to the agents' capabilities of modelling others. For example, 0-level agents are the agents that learn from their observations about the environment, and from any environmental rewards they receive[2]. 1-level agents are those agents that model others as 0-level agents. 2-level agents are those that model others as 1-level agents. Although in theory higher level agents could be recursively defined in the same manner, Vidal and Durfee's work only concentrates on the first three levels of agents, or more precisely, only up to 2-level sellers and 1-level buyers. Intuitively, agents with more complete models of others should do better. In practice, however, because of the computational costs associated with maintaining deeper (i.e., more complex) models, there should be a level at which the gains and the costs of having deeper models balance out for each agent. The main problem addressed in [77, 78] is to answer the question of when an agent

[2]This means that reinforcement learning is adopted by their 0-level agents. We contrast our use of reinforcement learning with theirs in Section 5.2.2.

benefits from having deeper models of others, or in other words, when it should stop keeping deeper models of others to take action.

Vidal and Durfee's work motivates and serves as a starting point for our research. However, we extend their information market model into a more general one where multiple sellers may offer goods of different qualities, the sellers may alter the quality of their goods, and there is a possibility of having dishonest sellers in the market.

The buying and selling agents proposed in [77, 78] are autonomous and learn by recursively modelling each other. As a result, the buying agents can learn to avoid sellers that have disappointed them in the past. However, the challenge of modelling sellers sufficiently well in order to detect dishonest sellers in the market is left for future work. In addition, those agents that keep deep recursive models of others suffer from the associated computational costs for maintaining these models. In fact, due to the infeasible complexity in implementing agents with deep models of others, the experimentation reported in [77, 78] is limited to only 1-level buyers and 2-level sellers. Moreover, their selling agents may not be able to achieve the goal of maximizing their expected profits because they only change product prices but do not consider adjusting product quality. This is especially problematic when their product quality does not meet the buyers' expectation.

We are also interested in designing buying and selling agents participating in market environments. Particularly, our goal is to design agents that are

- autonomous in making decisions (compared to the BargainFinder and Jango shopping agents [3, 14, 29]),
- equipped with learning capabilities in order to perform better and better (compared to agents of the Kasbah system [9]), and

- effective (i.e., they should be quick in identifying appropriate traders, and able to avoid costly computation as opposed to the agents proposed in [77, 78]).

We believe that in a market environment, reputation of sellers is an important factor that buyers can exploit to avoid interaction with dishonest sellers, therefore enhancing the opportunity to buy high value goods and reducing the risk of purchasing low value ones. Also, we believe that selling agents will increase their sales by not only adjusting the prices of their goods, but also by tailoring the quality of their goods to meet the buyers' specific needs. Consequently, our approach uses a combination of reinforcement learning and reputation modelling, and also gives selling agents the option of altering the quality of their goods. A full description of our model is provided in the next chapter.

Shopbots and Pricebots

Greenwald and Kephart [21, 22, 23] present a model of an agent economy consisting of shopbots (comparison shopping agents) and pricebots (selling agents using automated pricing algorithms). This is essentially a multi-agent marketplace where a single homogeneous good is offered for sale by S sellers and of interest to B buyers, with the assumption that $B >> S$. In this marketplace, each buyer b generates a purchase order at random times with rate ρ_b, while each seller reconsiders (and potentially resets) its price p_s at random times with rate ρ_s. Greenwald and Kepart are concerned with the dynamics of interaction among the pricebots' algorithms. Their ultimate aim is to identify those pricing algorithms that are most likely to be profitable, from both an individual and a collective standpoint. A variety of pricing algorithms were therefore simulated in this work. In particular, probabilistic pricing algorithms were explored in both high-information and low-information

settings. The results obtained via simulations show that the long run empirical frequencies of prices can converge to equilibria arbitrarily close to a Nash equilibrium; however, instantaneous price distribution need not converge. A Nash equilibrium is a vector of prices at which sellers maximize individual profits and from which they have no incentive to deviate [23].

Although the marketplace considered by Greenwald and Kepart may seem similar to ours, there are important differences between the two models:

- They assume that the number of buyers in the market is infinitely greater than the number of sellers ($B >> S$). This assumption gives them the convenience of studying the pricing strategies of sellers only, because the small number of sellers indicates that the behaviours of individual sellers will greatly influence the market. We think that this assumption is at times unrealistic. As a result, our model does not implement such an assumption and allows for any populations of buyers and sellers.

- They assume that the cost of production for all sellers is the same. In contrast, our model assumes that the production cost of multiple sellers may not be the same and that a seller may alter the quality of its goods by changing its production cost, in an attempt to reflect a more realistic environment.

- They assume that buyers in the market use one of the following two simple strategies: *(i) Bargain Hunter:* Buyer selects the seller with the lowest price, and purchases the good if that lowest price is less than the buyer's valuation of the good. *(ii) Any Seller:* Buyer selects a seller at random, and purchases the good if the price charged by that seller is less than the buyer's evaluation of the good. In contrast, our market model allows buyers to adopt any

buying strategy. In addition, our proposed buying strategy that combines reinforcement learning and reputation modelling goes beyond the consideration of price alone in selecting the most appropriate seller.

Auction Agent Models

There has also been considerable research on designing a different kind of electronic marketplace, termed an auction, where buyers are bidding competitively in order to acquire a good from a seller [5, 19, 24, 26, 50, 60, 83]. The desire for building various auction models naturally leads to the need for developing algorithms for the agents that participate in different auction environments (e.g., [5, 19, 24, 26, 50]). There has in fact been particular effort in designing algorithms for so-called combinatorial auctions, where buyers can place bids for arbitrary combinations of items [19, 25, 59], and for parallel auctions, where items are open for auction simultaneously [5].

The market mechanism[3] for the agent model that we consider in this thesis is distinct from that of the auctions discussed above. In the next chapter, we elaborate on the form of bidding protocol between the buyers and sellers in our marketplace, clarifying that it is the buyers in our model who are acting as auctioneers, and not the sellers as is the case in traditional auctions. In Section 6.3.4, we revisit the topic of designing algorithms for agents in other auction environments, as a possible branching point of our research, for future work.

[3]A market mechanism is a set of rules that govern how buyers and sellers in the market should interact and when a deal should be formed [11].

2.2 Reinforcement Learning

In this section, we first introduce the reinforcement learning problem followed by some illustrating examples. We then review some basic methods for solving the problem, and finally present some applications of reinforcement learning to the area of agent and multi-agent systems.

Our main references for the reinforcement learning problem, its demonstrating examples, and the fundamental methods for solving it are [54] and [65]. References for the applications of reinforcement learning will be listed as these applications are described.

2.2.1 The Reinforcement Learning Problem

The reinforcement learning problem is the problem of learning from interaction to achieve a goal. In this problem, an agent observes a current state s of the environment, performs an action a on the environment, and receives a feedback r from the environment. This feedback is also called *reward*, or *reinforcement*. The goal of the agent is to maximize the cumulative reward it receives in the long run.

More specifically, the agent and its environment interact at each of a sequence of discrete time steps, $t = 1, 2, 3, ...$ At each time step t, the agent receives some representation of the environmental state, $s_t \in S$, where S is the set of possible states. On that basis, the agent selects and performs an action $a_t \in A$, where A is the set of possible actions. One time step later, as the consequence of its action, the agent receives a numerical reward $r_{t+1} \in \mathbb{R}$, and finds itself in a new state s_{t+1}. Figure 2.1 illustrates the agent-environment interaction.

At each time step, the agent implements a policy π, which is a mapping from

the current state into the desirable action to be performed in that state. Solving a reinforcement learning problem means finding a policy that maximizes the reward that the agent receives over the long run. Such a policy is called the *optimal policy*.

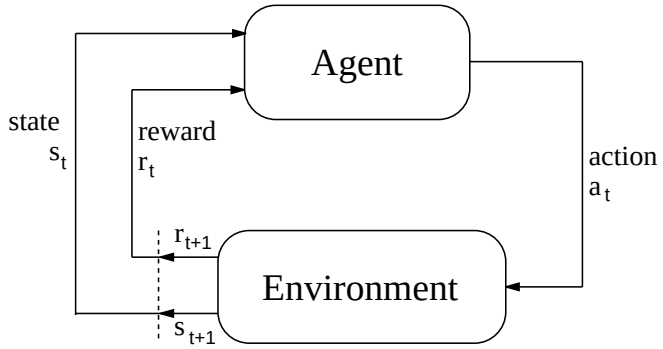

Figure 2.1: The agent-environment interaction in the reinforcement learning problem.

2.2.2 Examples

This section clarifies the terms *environmental state*, *actions* and *rewards* by presenting a number of examples. Section 2.2.3 then describes some fundamental methods for solving the reinforcement learning problem.

Game Playing

A reinforcement learning agent can be used to help a chess player to improve his play. At each time step, the environmental state can be the current configuration of all the pieces on the chessboard. The actions are all the possible moves. The rewards may be zero for most of the time and $+1$ when the player wins. Alternatively, we may give a reward of -1 for every time step until the game ends. This will encourage the agent to win as quickly as possible.

Bio-reactor

Consider a bio-reactor which is essentially a large vat of nutrients and bacteria used to produce a useful chemical. The rate at which the useful chemical is produced depends on moment-by-moment temperatures and stirring rates for the bio-reactor. Reinforcement learning can be applied to determine the optimal temperature and stirring rate at each time step. The states in this case are likely to be sensor readings of thermocouple plus symbolic inputs representing the ingredients in the vat and the target chemical. The actions may be target temperatures and target stirring rates that are passed to low-level control systems that, in turn, activate heating elements and motors to obtain the targets. The rewards may be moment-by-moment measures of the rate at which the useful chemical is produced. We notice that in this example each state is a vector (or list) of sensor readings and symbolic inputs, and each action is also a vector composed of target temperature and stirring rate. It is typical for a reinforcement learning problem to have states and actions represented as vectors. Rewards, however, should always be single numerical values.

Recycling Robot

Consider a mobile robot whose job is to collect empty pop cans in an office environment. The robot has sensors to detect cans, an arm with gripper to pick up a can to place it in an onboard bin, as well as a navigation system to help it move around. It operates on a rechargeable battery. A Reinforcement learning agent can be used to make high-level decisions on how to search for cans. Basically, the agent has to decide which of the following three actions the robot should perform: *(i)* actively searching for a can for a certain period of time, *(ii)* remaining stationary

to wait for someone to bring it a can, or *(iii)* heading back to his home base to recharge its battery. At each time step, the environmental state should be the state of the battery. The rewards can be zero most of the time, +1 when the robot is able to collect an empty can, and −1 if the battery goes all the way down. In this example, the reinforcement learning agent is not the entire robot, and the environmental states describe conditions within the robot, not the external environment. Reinforcement learning is used for the robot to make decisions on which action to take at each state, in order to maximize the cumulative reward the robot receives over the long run.

Inverted Pendulum

The setup for a famous reinforcement learning problem known as *inverted pendulum* or *pole balancing* is shown in Figure 2.2.

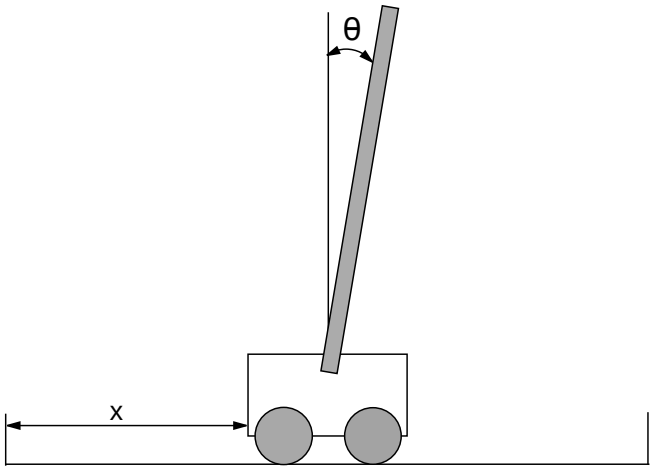

Figure 2.2: The *inverted pendulum* (also known as the *pole balancing*) problem.

The goal of the inverted pendulum problem is to apply forces on a cart moving

along a track so that the pole hinged from the cart will not fall over. A failure is said to occur if the pole falls past a given angle from vertical, or if the cart reaches an end of the track. The pole is reset to vertical after each failure. For this problem, a state may consist of the distance x and the angle θ, which are continuous variables. The actions are clearly *jerk left* or *jerk right*, which are discrete. The rewards could be $+1$ for a time step during which a failure did not occur. Alternatively, we could choose the rewards to be -1 for each failure and zero at all other times.

2.2.3 Reinforcement Learning Methods

Any method that is suited for solving the reinforcement learning problem described above is considered a reinforcement learning method. This subsection introduces three well-known, fundamental classes of methods for solving the reinforcement learning problem, namely dynamic programming, Monte Carlo, and temporal-difference learning methods. Note that a form of the temporal-difference learning method is used in our proposed algorithms for buying and selling agents in electronic marketplaces, which are presented in the next chapter.

Let us start with some necessary definitions and notations, which will be used later in the description of the reinforcement learning methods.

Definitions and Notations

Let $r_{t+1}, r_{t+2}, r_{t+3}, ...,$ be the sequence of rewards received after time step t. We define the *return*, R_t, to be

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + ... + r_T \tag{2.1}$$

where T is the *final time step*. This definition makes sense in applications in which there is a natural notion of the final time step, i.e., when the agent-environment interaction breaks naturally into subsequences called *episodes*. In cases where the agent-environment interaction does not break into episodes but goes on continually without limit, we define the return, R_t to be

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} (\gamma^k r_{t+k+1}) \tag{2.2}$$

where γ is called the *discount rate* ($0 \leq \gamma \leq 1$). The discount rate γ determines the present value of future rewards. That is, a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately. The goal of the agent in the reinforcement learning problem is to maximize the expected return, denoted by $E\{R_t\}$.

We define the *value of a state s under policy π*, denoted by $V^\pi(s)$, to be the expected return when starting in s and following π thereafter,

$$V^\pi(s) = E_\pi\{R_t | s_t = s\}. \tag{2.3}$$

The function V^π is called the *state value function* under policy π.

Similarly, we define the *value of taking action a in state s under policy π*, denoted by $Q^\pi(s, a)$, to be the expected return starting from s, taking action a, and thereafter following policy π,

$$Q^\pi(s, a) = E_\pi\{R_t | s_t = s, a_t = a\}. \tag{2.4}$$

The function Q^π is called the *state-action value function* under policy π.

Let $P^a_{ss'}$ denote the probability that the environment changes to a new state s' when the agent executes action a in state s. $P^a_{ss'}$ is called the *transition probability*. Let $R^a_{ss'}$ denote the expected value of the next reward received when the agent

executes action a in state s and the environment changes to the next state s'. $R^a_{ss'}$ is called the *expected immediate reward*. The quantities $P^a_{ss'}$ and $R^a_{ss'}$ specify the most important aspects of the agent's environment.

Dynamic Programming

Dynamic programming (DP) refers to a collection of algorithms that can be used to discover optimal policies for the reinforcement learning problem. DP methods are well developed mathematically, but require a complete model of the environment, given by a set of transition probabilities, $P^a_{ss'}$, and a set of expected immediate rewards, $R^a_{ss'}$. Figure 2.3 below shows a representative DP method called *Value Iteration*. This method iteratively computes the state value function $V(s)$ with the maximum one taken over all actions. It then derives the required policy $\pi(s)$ based on this optimal value function.

Initialize $V(s)$ arbitrarily for all s.

Repeat

$\Delta \leftarrow 0$

For each $s \in S$:

$v \leftarrow V(s)$

$V(s) \leftarrow \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$ (a small positive number)

Output a policy π such that

$\pi(s) = \arg\max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$

Figure 2.3: A DP method called *value iteration*.

Monte Carlo

Monte Carlo (MC) methods are ways of solving the reinforcement learning problem based on averaging sample returns. Unlike DP methods, MC methods do not require a complete model of the environment. They only need *experience*, that is, sample sequences of states, actions, and rewards from an online or simulated interaction with the environment. MC methods are defined only for episodic tasks, in which the agent-environment interaction is divided into episodes. It is only upon the completion of an episode that the value functions, namely $V(s)$ or $Q(s, a)$, and policies are updated. MC methods are thus incremental in an episode-by-episode sense, not in a step-by-step sense.

Figure 2.4 introduces a MC algorithm that uses the idea of policy iteration. It first evaluates the state-action value function $Q(s, a)$ under an arbitrary policy π_0. Then, it improves π_0 using $Q(s, a)$ to yield a better policy π_1. It repeatedly evaluates $Q(s, a)$ under π_1, and improves it again to yield an even better policy π_2, and so on. As many episodes are experienced, the approximate policy and the approximate value function will asymptotically approach the optimal policy and the corresponding optimal value function, respectively.

Initialize, for all $s \in S$ and all $a \in A$:

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Repeat forever:

(a) Generate an episode using π

// Evaluating $Q(s, a)$ under π

(b) For each pair (s, a) appearing in the episode:

$r \leftarrow$ the return following the first occurrence of (s, a)

Append r to $Returns(s, a)$

$Q(s, a) \leftarrow$ average($Returns(s, a)$)

// Improving π

(c) For each s in the episode:

$\pi(s) \leftarrow \arg\max_a Q(s, a)$

Figure 2.4: An MC algorithm using *policy iteration*.

Temporal-Difference Learning

Like MC methods, temporal-difference (TD) learning methods can learn directly from experience without a model of the environment. Unlike MC methods, which must wait until the end of an episode to update the value function (only then is the return R_t known), TD methods only need to wait until the next time step. TD methods are thus incremental in a step-by-step sense.

One of the most widely-used TD methods is known as the *Q-learning algorithm*, as illustrated in Figure 2.5. For a state s, the Q-learning algorithm chooses an action a to perform such that the state-action value $Q(s, a)$ is maximized. If performing

action a in state s produces a reward r and a transition to state s', then the corresponding state-action value $Q(s, a)$ is updated accordingly. State s is now replaced by s' and the process is repeated until reaching the terminal state.

Initialize $Q(s, a)$ arbitrarily
Initialize s
Repeat
  Choose a to perform in s to maximize $Q(s, a)$:
    $a \leftarrow \arg\max_a Q(s, a)$
  Take action a, observe reward r and transition to new state s'
  Update $Q(s, a)$:
    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
    // where α is called *the learning rate* ($0 \leq \alpha \leq 1$) and
    // γ is the discount rate ($0 \leq \gamma \leq 1$)
  $s \leftarrow s'$
until s is terminal

Figure 2.5: A TD learning algorithm called *Q-learning*.

In this thesis, we use a simple form of TD learning as follows. Given a state s, our agent will select an action that maximizes the state value $V(s)$. While interacting with the environment, the agent attempts to make its estimated state value become more and more accurate. To do this, the agent adjusts the value of the earlier state to be closer to the value of the later state, using a simple updating rule. Figure 2.6 illustrates this learning process with the updating rule shown in equation (2.5), where the learning rate α is initially set to 1 and then reduced over time to a small positive fraction.

Initialize $V(s)$ for all s

Repeat forever

Choose a to perform in s to maximize $V(s)$:

$a \leftarrow \arg\max_a V(s)$

Take action a, observe transition from state s to state s' with the corresponding state values being $V(s)$ and $V(s')$

Update $V(s)$:

$$V(s) \leftarrow V(s) + \alpha[V(s') - V(s)] \tag{2.5}$$

// where α is the learning rate ($0 \leq \alpha \leq 1$)

$s \leftarrow s'$

Figure 2.6: A simple form of TD learning methods used in this thesis.

2.2.4 Applications of Reinforcement Learning in Agent and Multi-Agent Systems

We complete our overview of reinforcement learning by discussing sample applications where it has been used. Reinforcement learning has been applied extensively in various learning problems for agent and multi-agent systems. This is reflected by the growing number of publications in the area [31, 41, 45, 46, 49, 57, 62, 66, 79, 80].

The successful application of reinforcement learning to a wide range of agent and multi-agent problems motivates us to consider reinforcement learning as a promising method for the problem of designing effective learning agents for electronic market environments, i.e., the problem that we are interested in addressing in this thesis. Section 5.2.2 offers a more detailed justification for our choice to incorporate reinforcement learning.

Note that in multi-agent systems, it is often the case that agents are working together to complete a common goal or are modelling each other to address potential conflicts. This contrasts with our market model environment, where agents are concerned with learning to improve their individual satisfaction. In Section 6.3.5, we describe some possible extensions of our model for multi-agent systems other than market-based ones.

The Block Pushing Problem

Sen et al. [62] address the problem of how multiple agents can learn to appropriately coordinate their activities in order to accomplish a common task. They attempt to achieve coordination without agents' sharing information with one another. In particular, they apply the Q-learning algorithm to a *block pushing problem*, the problem in which multiple agents are independently instructed to move a block from a starting position to some goal position. Their work shows that agents can learn complementary strategies to fulfill a common task without any knowledge about each other. The main result presented in [62] is that although individual agents are independently optimizing their own environmental rewards, global coordination between the agents can be obtained without any explicit or implicit form of communication.

Agent Coordination by Explicit Communication

Weiss [79] addresses the problem of coordination in multi-agent systems using a different approach. According to his approach, agents learn to coordinate their actions by explicitly communicating with one another. He introduces two reinforcement learning based algorithms called the Action Estimation (ACE) algorithm and the

Action Group Estimation (AGE) algorithm. In both algorithms, the agents first learn to estimate the goal relevance of their actions. They then coordinate their actions and generate appropriate action sequences based on their goal relevance estimates. The main difference between the ACE algorithm and the AGE algorithm is that the agents executing the AGE algorithm do not compete for carrying out individual actions (as those executing the ACE algorithm), but for carrying out groups of actions.

The Predator-Prey Problem

Tan [66] applies the Q-learning method to the predator-prey problem. He considers an environment in which two types of agents, namely predators and preys, act and live. The goal of a predator is to learn to catch a prey. Each agent can choose its action from four possible two-dimensional ones, namely moving up, moving down, moving left, and moving right. At each time step, each prey randomly moves around, and each predator must learn to select its next move based on the decision policy it has gained through the Q-learning algorithm. Tan's work shows that communication can be used as a means for improving learning. In particular, he identifies two kinds of important information which the predators should exchange in order to support one another in their learning: *(i)* Visual input, which describes the relative distance from a predator to the closest prey within its limited visual field. *(ii)* Decision policies, which describes what a predator has learned so far with respect to the state-action values (equation (2.4)).

Learning Agent for Visual Search Tasks

Minut and Mahadevan [43] propose a model of selective attention for visual search tasks. The problem they address is, given an object and an environment, how to build a vision agent that learns where the object is most likely to be found, and how to direct its gaze towards the object. The system must produce a set of landmarks $\{L_0, L_1, \ldots, L_n\}$ (the regions in the environment), together with a policy on this set which leads the camera towards the most probable region containing the target object. The vision agent consists of two interacting modules: A reinforcement learning module learns a policy on a set of regions in the room for reaching the target object. It uses the Q-learning method with the expected value of the sum of discounted rewards (equation (2.2)) chosen as the objective function. By selecting an appropriate gaze direction at each time step, the reinforcement learning module provides top-down control in the selection of the next fixation point. The second module performs bottom-up visual processing to provide the agent with a set of locations of interest in the current image, and also to detect and identify the target object. Minut and Mahadevan's experimental results show that the number of fixations to the target object significantly decreases with the number of training epochs. Also, their results show that the learned policy to find the target object is invariant to small physical displacements and object inversion.

2.3 Models of Trust and Reputation

Multi-agent systems are typically intended for large, open, dynamic, and unpredictable environments. In such environments, an agent often has incomplete (or even incorrect) knowledge about other agents. To reduce the risk of fraud and

deception, different levels of security (such as passwords, digital certificates, and access control capabilities) have been built into the infrastructure level of multi-agent environments. Although these hard security techniques guarantee that an agent is authenticated and authorized, they do not ensure that the agent will exercise its authorization in a desirable way. Mechanisms of reputation (or trust) are therefore proposed as soft security techniques to complement the existing hard security ones. The need for such soft security mechanisms is especially more noticeable in e-commerce applications where it is very important for an agent to know the credibility of the agent that it would like to initiate transactions with. Consequently, modelling reputation as a computational concept has recently become an interesting topic to many researchers. In fact, there has been a series of workshops on Deception, Fraud and Trust in Agent Societies (e.g., [16, 30, 55]).

In developing learning algorithms for agents in electronic marketplaces, we use a reputation mechanism, in addition to reinforcement learning, to provide added robustness to buying agents. By dynamically maintaining sets of reputable and disreputable selling agents, buying agents should together isolate and weed out dishonest selling agents, and therefore obtain better satisfaction in doing business with the reputable ones. Details on how a buying agent may build and update its sets of reputable and disreputable selling agents will be discussed in the next chapter. In this section, we briefly present three different models of trust and reputation that may serve as typical examples for soft security mechanisms. We focus on this research, in order to discuss important challenges in the design of these models that have motivated the research of this thesis.

2.3.1 A Social Mechanism of Reputation Management

Yu and Singh [84] propose a social mechanism of reputation management for electronic communities. In their approach, agent a assigns a trust rating to agent b, denoted by $T_a(b)$, based on

(i) its direct interactions with b,

(ii) the ratings of b as given by b's neighbors, and

(iii) a's ratings of those neighbors.

Trust ratings are defined to be numerical values in between -1 and 1. Initially, all trust ratings are set to 0. After an interaction with agent b, agent a may increase or decrease $T_a(b)$ using positive evidence α or negative evidence β respectively, depending on whether or not b has cooperated with a. Agent a also maintains two thresholds, ω_a and Ω_a, where $-1 < \omega_a < \Omega_a < 1$. Agent a will trust agent b if $T_a(b) \geq \Omega_a$; agent a will mistrust agent b if $T_a(b) \leq \omega_a$; otherwise, if $\omega_a < T_a(b) < \Omega_a$ then agent a must decide on some other grounds; that means, it does not have sufficient information to decide whether it should trust agent b. Each agent has a set of potentially changing neighbors, with whom it may directly interact. How an agent evaluates the reputation of another will depend in part on the testimonies of the latter's neighbors. Let $\chi = \langle a_0, ..., a_n \rangle$ be a referral chain from agent a_0 to agent a_n, where a_i is a neighbor of a_{i+1}. Then, a_0 can use the referral chain χ to compute its trust rating $T_0(n)$ towards a_n as follows:

$$T_0(n) = T_0(1) \otimes ... \otimes T_{n-1}(n) \tag{2.6}$$

where the *trust propagation operator* $\otimes$ is defined as

$$x \otimes y = \begin{cases} x \times y & \text{if } x \geq 0 \text{ and } y \geq 0, \\ -|x \times y| & \text{otherwise.} \end{cases} \tag{2.7}$$

The reputation mechanism proposed in [84] represents a feasible approach in formalizing trust as a computational concept. Nevertheless, it has at least the following drawbacks: *(i)* In computing a trust rating using equation (2.6), different referral chains may result in different and possibly conflicting values. *(ii)* Its updating scheme using constant factors α and β does not take into consideration the extent to which an agent has (or has not) cooperated. That is, a greatly cooperative agent and a slightly cooperative agent will receive the same increasing amount in their trust ratings. Similarly, a greatly disappointing agent and a slightly disappointing agent will receive the same decreasing amount in their trust ratings.

2.3.2 REGRET

Sabater and Sierra [55] propose a model of reputation called REGRET. The most important feature of this model is that it takes into account the individual dimension, the social dimension, and the ontological dimension of reputation. Let us briefly describe these dimensions.

1. Individual Dimension:

   The individual reputation rating of agent b at current time t from agent a's point of view on aspect φ, denoted as $R_{a\to b}(\varphi)$, is calculated as a weighted mean of the individual impression ratings, giving more relevance to recent impressions:

   $$R_{a\to b}(\varphi) = \sum_i \rho(t, t_i) W_i \tag{2.8}$$

   where $\rho(t, t_i)$ is a normalized value that has higher value when t_i is closer to t, W_i is the individual impression (or reputation) rating at time t_i, and φ is the aspect in which the reputation of agent b is measured (e.g., product price, product quality etc.).

2. Social Dimension:

   According to REGRET, an individual inherits the reputation of the group it belongs to by default. Also, an individual uses the experiences of his group's members as a help to shape his opinion on some manner. This idea introduces three social reputation ratings:

   - The reputation rating based on the interaction of agent a with members of group B, the group that agent b belongs to:

     $$R_{a \to B}(\varphi) = \sum_{b_i \in B} \omega^{ab_i} R_{a \to b_i}(\varphi) \tag{2.9}$$

     where $\sum_{b_i \in B} \omega^{ab_i} = 1$.

   - The reputation rating based on what the members of group A, the group that agent a belongs to, think about agent b (the agent being evaluated):

     $$R_{A \to b}(\varphi) = \sum_{a_i \in A} \omega^{a_i b} R_{a_i \to b}(\varphi) \tag{2.10}$$

     where $\sum_{a_i \in A} \omega^{a_i b} = 1$.

   - The reputation rating based on what the members of A think about group B

     $$R_{A \to B}(\varphi) = \sum_{a_i \in A} \omega^{a_i B} R_{a_i \to B}(\varphi) \tag{2.11}$$

     where $\sum_{a_i \in A} \omega^{a_i B} = 1$.

   The reputation rating that combines the individual reputation rating and the three social reputation ratings is defined as

   $$SR_{a \to b}(\varphi) = \xi_{ab} R_{a \to b}(\varphi) + \xi_{aB} R_{a \to B}(\varphi) + \xi_{Ab} R_{A \to b}(\varphi) + \xi_{AB} R_{A \to B}(\varphi) \tag{2.12}$$

where $\xi_{ab} + \xi_{aB} + \xi_{Ab} + \xi_{AB} = 1$, and $SR_{a \to b}(\varphi)$ denotes the reputation rating that agent a assigns to agent b for aspect φ, taking into account the social dimension.

3. Ontological Dimension

The reputation rating calculated by (2.12) is only linked to a single aspect, namely φ. In REGRET, the reputation of an agent is not considered as a single and abstract concept, but rather a multi-faceted concept. For example, the reputation of being a good seller may summarize the reputation of offering products at reasonable prices, the reputation of delivering products punctually, and the reputation of having good product quality. The different types of reputation and how they are combined to obtain new types is called the ontological dimension of reputation. Each individual agent usually has a different ontological structure to combine reputations, and a different way to weigh the importance of reputations when they are combined. For instance, agent a may calculate the reputation of agent b as a good seller using the formula:

$$\begin{aligned} OR_{a \to b}(good\_seller) = \ & \omega_1 SR_{a \to b}(product\_price) + \omega_2 SR_{a \to b}(delivery\_date) + \\ & \omega_3 SR_{a \to b}(product\_quality) \end{aligned} \tag{2.13}$$

where $\omega_1 + \omega_2 + \omega_3 = 1$; $OR_{a \to b}(good\_seller)$ denotes the reputation of b in aspect *being a good seller* from agent a's perspective, taking into account the ontological dimension; and $SR_{a \to b}(product\_price)$, $SR_{a \to b}(delivery\_date)$, and $SR_{a \to b}(product\_quality)$ are respectively computed by equation (2.12) above.

The main advantage of [55] is that it presents a model that takes into account the social dimension and the ontological structure of reputation. However, a number of drawbacks could be identified: *(i)* The process of calculating the final reputation rating is very complex and quite computationally expensive. *(ii)* REGRET does not specify how an agent may decide on the individual impression W_i, which is a building block for computing $R_{a \to b}(\varphi)$ of (2.8). *(iii)* There are no guidelines for how to set the normalized values (e.g., ω^{ab_i}, $\omega^{a_i B}$, ξ_{ab}, ξ_{aB} etc.), which obviously play important roles in calculating the respective reputation ratings. *(iv)* [55] does not discuss how groups of agents are formed so that the social dimension formulas (e.g., equations (2.9), (2.10), and (2.11)) can be applied. One conclusion that we reached, upon examination of this work, was that it would be beneficial to adopt a less complex model of reputation in this thesis. For example, we set aside the social dimension of reputation modelling, but do revisit this aspect as part of future work in Section 6.3.1.

2.3.3 A Model for Trust Acquisition and Propagation

Esfandiari and Chandrasekharan [16] introduce a model for trust acquisition and trust propagation. They define trust as a function T between any two agents a_1 and a_2 from a set A of agents:

$$T : A \times A \mapsto [0, 1]. \tag{2.14}$$

For example, $T(Mike, Jim) = 0.8$ means Mike trusts Jim 80%. Three different ways in which trust is acquired individually are explored:

1. Trust Acquisition by Observation:

This is the case where trust decisions are made based on observation. If agent a_1 is able to observe past performances of agent a_2, then it will consider the acquired statistics as a value based on which to calculate trust. In general, for any two agents a_1 and a_2, a trust situation S and an observed performance statistics O, the observed trust $T_{obs}(a_1, a_2)$ is evaluated as the probability of O given S.

$$T_{obs}(a_1, a_2) = P(O|S) \tag{2.15}$$

2. Trust Acquisition by Interaction:

   Instead of observation, agent a_1 may decide on how it should trust agent a_2 by asking a_2 some questions for which it already knows the answers. The interaction-based trust $T_{inter}(a_1, a_2)$ is then calculated based on the number of correct answers provided by agent a_2. That is,

$$T_{inter}(a_1, a_2) = \text{number of correct answers/total number of answers} \tag{2.16}$$

3. Trust Acquisition Using Institutions:

   The way trust is acquired in this case is analogous to that in human society. Agent a_1 may trust agent a_2 if a_2 possesses some significant identities such as badges, uniforms, degrees, titles etc., or if a_2 belongs to a recognized institution. For example, agent a_1 may set

$$T_{inst}(a_1, a_2) = 1 \tag{2.17}$$

   because a_2 is a member of a well-known organization.

Socially, trust is acquired via propagation. Esfandiari and Chandrasekharan's model of trust propagation is closely related to [85]. The agent community is

modelled as a directed, labelled graph, where an edge (a, b) represents the trust value $T(a, b)$ that agent a has on agent b (see Figure 2.7).

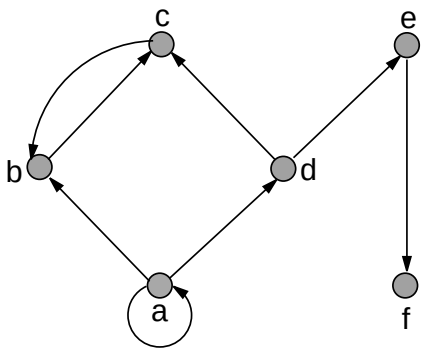

Figure 2.7: A directed graph for trust propagation.

Their choice of a directed graph highlights the fact that trust is not symmetric ($T(a, b)$ is not necessarily equal to $T(b, a)$), and is not reflexive ($T(a, a)$ is not necessarily equal to 1). Absent edges represent unknown trust values. They take into account the fact that trust is only weakly transitive and therefore should be decreased along the chain:

$$T_{prop}(a, c) = T(a, b_1) \times T(b_1, b_2) \times \ldots \times T(b_n, c) \tag{2.18}$$

where b_i's are the intermediate agents in the path from a to c.

Two problems are identified in using directed graph for calculating propagated trust:

- Different paths may give contradictory values.
- Cycles in a path can arbitrarily decrease the trust value. For instance, one may decide to loop 3 times at a in the above graph before reaching a neighboring agent.

Their proposed solution is to replace a strict trust calculation with a trust interval. The bounds of the interval are calculated as the minimum and maximum possible values obtained by applying the above calculation to the paths that contain no cycles.

Esfandiari and Chandrasekharan propose a model with different methods for trust acquisition. They also suggest a solution for dealing with the problem of using a directed graph to calculate propagated trust. However, the following shortcomings need to be considered: *(i)* Their model does not discuss how to combine different individual trust ratings acquired by observation, interaction, and using institution (e.g., T_{obs}, T_{inter}, and T_{inst}) into a unified value. *(ii)* Also, their model does not specify how to combine the trust acquired individually and the trust acquired socially (by propagation). *(iii)* Their definition of trust does not make a clear distinction between distrust and lack of knowledge about trust. For instance, it is not clear whether $T(a, b) = 0$ means that agent a distrusts agent b or that agent a has no knowledge about the trustworthiness of agent b. *(iv)* There is no discussion in [16] about which identities and institutions are acceptable for trust, and how to determine what trust values to be assigned to which identities (institutions).

As we mentioned earlier, our approach in designing algorithms for trading agents in electronic marketplaces is to use a combination of reinforcement learning and reputation modelling. In contrast to [16] and [55], the proposed reputation mechanism enables buying agents to quickly identify the reputable selling agents while avoiding the disreputable ones. As opposed to [16], our approach allows for a clear distinction between distrust and lack of knowledge about trust. The reputation updating scheme we use bases its updating conditions on appropriate market factors, making it ready and suitable for electronic market settings, compared to the above-described reputation models. Especially, in contrast to [84], we propose the

use of variable cooperation and non-cooperation factors to implement the common idea that transactions with higher values should be better appreciated than lower ones. We also theoretically explore how to set a penalty factor to realize the traditional assumption that reputation should be difficult to build up but easy to tear down. Our reputation modelling is fully described in conjunction with the proposed buying algorithm in Section 3.2.1. Detailed discussion on the possible advantages of our reputation mechanism in contrast with the above-presented ones is offered in Section 5.3.

2.4 Chapter Summary

In this chapter we review relevant research that has been done in related areas. We begin the first section with a discussion on the concepts of agent, multi-agent systems, and e-commerce. We define an agent as a computer program that can perceive, reason, act, communicate and coordinate. We differentiate agents from objects in the object-oriented programming model by pointing out that agents are capable of flexible autonomous behaviours, namely reactivity, sociality and proactiveness. We define a multi-agent system as a system (or environment) composed of multiple interacting agents, and discuss that a multi-agent system should include computational infrastructure such as protocols for agents to communicate and protocols for agents to interact. We discuss several components of e-commerce and identify some factors based on which to decide whether agent technologies are appropriate for e-commerce. We then move on to review different agent models for e-commerce, ranging from non-learning, non-autonomous to learning and autonomous models. We discuss the advantages and shortcomings of these models in preparation for the description of our model presented in the next chapter.

We introduce reinforcement learning in the next section. First, we give a formal definition of the reinforcement learning problem, followed by a number of examples to illustrate this problem. We then review the three fundamental classes of methods for solving the reinforcement learning problem, namely dynamic programming, Monte Carlo, and temporal-difference learning methods. We note that a form of the temporal-difference learning method is used in the algorithms we propose in the next chapter for buying and selling agents in electronic marketplaces. We end the section with an overview of various works in agent and multi-agent systems where reinforcement learning has been used as a major tool for addressing the problems.

Reputation modelling is the theme of discussion in the last section. We first motivate the need for having reputation (or trust) mechanisms, not only as soft security techniques to complement the existing hard security ones, but also as an essential, desirable feature for e-commerce applications: Clearly, an agent would want to know the credibility of a buying or selling agent before initiating a transaction with that agent. We mention that a reputation mechanism is used in combination with reinforcement learning to provide added robustness to our proposed agents. Finally, we present three different models that serve as typical examples for reputation mechanisms, and clarify how our proposed reputation model aims to overcome certain shortcomings in these approaches.

Chapter 3

The Proposed Algorithms

The problem of how to develop algorithms that guide the behaviours of personal, intelligent agents participating in electronic marketplaces is a subject of increasing interest from both the academic and industrial research communities [9, 10, 14, 21, 37, 83]. Since a multi-agent electronic market environment is, by its very nature, open (agents can enter or leave the environment at will), dynamic (information such as prices, product quality etc. may be altered), uncertain (agents lack perfect knowledge of one another) and untrusted (there may be dishonest agents), it is very important that participant agents are equipped with effective and feasible learning algorithms to accomplish their delegated tasks or achieve their delegated goals. In this chapter, we propose reputation-oriented reinforcement learning based algorithms for buying and selling agents in electronic market environments.

The chapter is organized as follows: Section 3.1 describes our agent market model, which can be used for e-commerce applications. Section 3.2 presents the proposed algorithms for buying and selling agents respectively, followed by a simple numerical example to illustrate how the algorithms work. Section 3.3 investigates

the worst case scenario of a buying agent to answer the question of how much the buying agent could be harmed by a dishonest selling agent. Section 3.4 discusses the roles of the parameters used in the proposed algorithms and provides some general guidelines to set these parameters. Section 3.5 ends the chapter with a summary of the main points discussed in the chapter.

3.1 The Agent Market Model

We model the agent environment as an open marketplace which is populated with economically-motivated agents. The nature of an open marketplace allows the economic agents, which we classify as *buyers* and *sellers*, to freely enter or leave the market. Buyers and sellers are self-interested agents whose goal is to maximize their own benefit.

Our market environment is rooted in an information delivery infrastructure such as the Internet, which provides agents with virtually direct and free access to all other agents. The process of buying and selling goods is realized via a *contract-net* like mechanism [13, 63], which consists of three elementary phases:

(i) When a buyer b is in need of some good g, it will announce its request for that good to all the sellers, using multi-cast or possibly broadcast.

(ii) After receiving the request from b, those sellers that have good g available for sales will send a message to b, stating their price bids for delivering the good.

(iii) Buyer b evaluates the submitted bids and selects a suitable seller to purchase good g. Buyer b then pays the chosen seller and receives the good from that seller.

Thus, the buying and selling process can be viewed as an *auction* where sellers play the role of bidders and buyers play the role of auctioneers, and a seller is said to be *winning the auction* if it is able to sell its good to the buyer[1]. Figure 3.1 illustrates the three basic phases of this process.

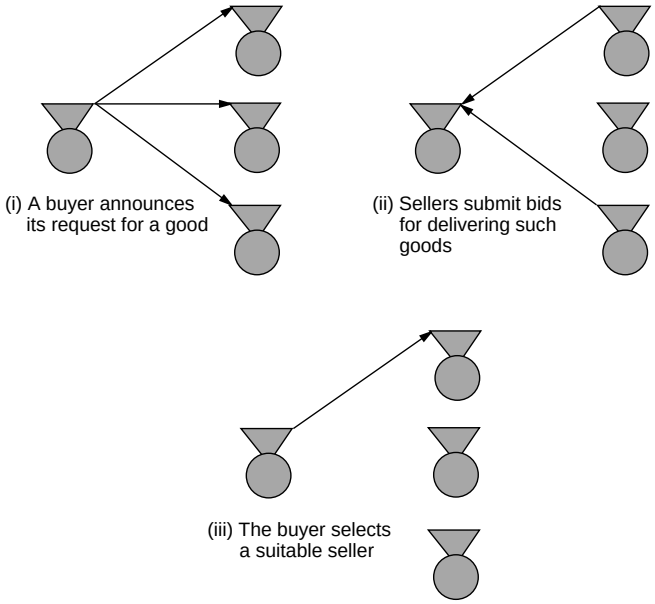

Figure 3.1: Three basic phases of the buying and selling process.

In order for our marketplace to be more realistic and also more interesting, we make the following assumptions:

- The quality of a good offered by different sellers may not be the same.
- A seller may alter the quality (in addition to the price) of its goods.

[1]Note that in traditional auctions, buyers are bidders and sellers are auctioneers. So, our view is opposite to the traditional view of auctions, with buyers auctioning off themselves as potential customers.

- It is possible that some dishonest sellers exist in the market.
- A buyer can examine the quality of the good it purchases only after it receives that good from the selected seller.
- Each buyer has some way to evaluate the good it purchases, based on the price and the quality of the good received.

Thus in our market environment, a buyer tries to find those sellers whose goods best meet its demanded value, while a seller tries to maximize its profit by setting suitable prices for and providing more customized value to its goods, in order to satisfy the buyers' needs.

3.2 The Proposed Learning Algorithms

This section presents our proposed reputation-oriented reinforcement learning algorithms for buyers and sellers in electronic marketplaces, respectively. The algorithms are aimed at maximizing the expected values of goods and avoiding the risk of purchasing low quality goods for buyers, and maximizing the expected profits for sellers. Note that it is possible for both a seller s and a buyer b to be *winning* in a business transaction. This happens when seller s could choose a price p to sell good g to buyer b that maximized its expected profit, and buyer b decided that purchasing good g at price p from seller s would maximize its expected value of goods. We also provide a simple numerical example to illustrate how the algorithms work.

3.2.1 Buying Algorithm

Consider the scenario where a buyer b announces its request for some good g. Let G be the set of goods, P be the set of prices, and S be the set of all sellers in the marketplace. G, P, and S are finite sets.

Buyer b models the reputation of all sellers in the market using function $r^b : S \mapsto (-1, 1)$, which is called the *reputation function* of b. Initially, buyer b sets the *reputation rating* $r^b(s) = 0$ for every seller $s \in S$. That means that initially buyer b is neither in favour of nor has prejudice against any sellers in the market[2]. After each transaction with a seller s, buyer b will update (increase or decrease) $r^b(s)$ depending on whether or not s satisfies b in the transaction. A seller s is considered *reputable* by buyer b if $r^b(s) \geq \Theta$, where Θ is buyer b's *reputation threshold* ($0 < \Theta < 1$). A seller s is considered *disreputable* by buyer b if $r^b(s) \leq \theta$, where θ is buyer b's *disreputation threshold* ($-1 < \theta < 0$). A seller s with $\theta < r^b(s) < \Theta$ is neither reputable nor disreputable to buyer b. In other words, b does not have enough information to decide on the reputation of s. Let S_r^b and S_{dr}^b be the sets of reputable and disreputable sellers to buyer b respectively, i.e.,

$$S_r^b = \{s \in S \mid r^b(s) \geq \Theta\} \subseteq S, \tag{3.1}$$

and

$$S_{dr}^b = \{s \in S \mid r^b(s) \leq \theta\} \subseteq S. \tag{3.2}$$

Buyer b will focus its business on the reputable sellers and stay away from the disreputable ones.

Buyer b estimates the expected value of the goods it purchases using the *expected*

[2]Buyer b also sets $r^b(s) = 0$ for any new seller s, who has just entered the market. Likewise, a new buyer b sets $r^b(s) = 0$ for every seller s in the market, once b first joins in the environment.

value function $f^b : G \times P \times S \mapsto \mathbb{R}$. Hence, the real number $f^b(g, p, s)$ represents buyer b's expected value of buying good g at price p from seller s.

Since multiple sellers may offer good g with different qualities and a seller may alter the quality of its goods, buyer b puts more trust in the sellers with good reputation. Thus, it chooses among the reputable sellers in S_r^b a seller $\hat{s}$ that offers good g at price p with maximum expected value:

$$\hat{s} = \arg \max_{s \in S_r^b} f^b(g, p, s), \tag{3.3}$$

where arg is an operator such that $\arg f^b(g, p, s)$ returns s.

If no sellers in S_r^b submit bids for delivering g (or if $S_r^b = \emptyset$), then buyer b will have to choose a seller $\hat{s}$ from the non-reputable sellers, provided that $\hat{s}$ is not a disreputable seller:

$$\hat{s} = \arg \max_{s \in (S - (S_r^b \cup S_{dr}^b))} f^b(g, p, s). \tag{3.4}$$

In addition, with a small probability ρ, buyer b chooses to explore (rather than exploit) the marketplace by randomly selecting a seller $\hat{s} \in (S - S_{dr}^b)$. This gives buyer b an opportunity to discover new reputable sellers. Initially, the value of ρ should be set to 1, then decreased over time to some fixed minimum value determined by b.

After paying seller $\hat{s}$ and receiving good g, buyer b can examine the quality $q \in Q$ of good g, where Q is a finite set of real values representing product qualities. It then calculates the true value of good g using the *true product value function* $v^b : G \times P \times Q \mapsto \mathbb{R}$. For instance, if buyer b considers the quality of good g to be twice more important than its price, it may set $v^b(g, p, q) = 2q - p$.

The expected value function f^b is now incrementally learned in a reinforcement

learning framework:

$$\Delta = v^b(g, p, q) - f^b(g, p, \hat{s}), \tag{3.5}$$

$$f^b(g, p, \hat{s}) \leftarrow f^b(g, p, \hat{s}) + \alpha\Delta, \tag{3.6}$$

where α is called the *learning rate* ($0 \leq \alpha \leq 1$). Similar to ρ, the learning rate α should initially be set to a starting value of 1 and then reduced over time to a fixed minimum value chosen depending on individual buyers[3].

Thus, if $\Delta = v^b(g, p, q) - f^b(g, p, \hat{s}) \geq 0$ then $f^b(g, p, \hat{s})$ is updated with the same or a greater value than before. This means that seller $\hat{s}$ has a good chance to be chosen by buyer b again if it continues offering good g at price p in the next auction. Conversely, if $\Delta < 0$ then $f^b(g, p, \hat{s})$ is updated with a smaller value than before. This implies that seller $\hat{s}$ may not be selected by buyer b in the next auction if it continues selling good g at price p.

In addition to updating the expected value function, the reputation rating $r^b(\hat{s})$ of seller $\hat{s}$ also needs to be updated. Let $\vartheta^b(g) \in \mathbb{R}$ be the product value that buyer b demands for good g. In other words, the demanded product value $\vartheta^b(g)$ is buyer b's threshold for the true product value $v^b(g, p, q)$. We use a reputation updating scheme motivated by that proposed in [84] as follows:

If $v^b(g, p, q) - \vartheta^b(g) \geq 0$, that is, if seller $\hat{s}$ offers good g with value greater than or equal to the value demanded by buyer b, then its reputation rating $r^b(\hat{s})$ is increased by

$$r^b(\hat{s}) \leftarrow \begin{cases} r^b(\hat{s}) + \mu(1 - r^b(\hat{s})) & \text{if } r^b(\hat{s}) \geq 0, \\ r^b(\hat{s}) + \mu(1 + r^b(\hat{s})) & \text{if } r^b(\hat{s}) < 0, \end{cases} \tag{3.7}$$

where μ is a positive factor called the *cooperation factor*[4] ($\mu > 0$).

[3]This adjustment of the learning rate is standard to reinforcement learning methods [65].

[4]Buyer b will consider seller $\hat{s}$ as being *cooperative* if the good $\hat{s}$ sells to b has value greater than or equal to that demanded by b.

Otherwise, if $v^b(g,p,q) - \vartheta^b(g) < 0$, that is, if seller $\hat{s}$ sells good g with value less than that demanded by buyer b, then its reputation rating $r^b(\hat{s})$ is decreased by

$$r^b(\hat{s}) \leftarrow \begin{cases} r^b(\hat{s}) + \nu(1 - r^b(\hat{s})) & \text{if } r^b(\hat{s}) \geq 0, \\ r^b(\hat{s}) + \nu(1 + r^b(\hat{s})) & \text{if } r^b(\hat{s}) < 0, \end{cases} \tag{3.8}$$

where ν is a negative factor called the *non-cooperation factor*[5] ($\nu < 0$).

The set of reputable sellers to buyer b now needs to be updated based on the new reputation rating $r^b(\hat{s})$, as in one of the following two cases:

- If $(\hat{s} \in S_r^b)$ and $(r^b(\hat{s}) < \Theta)$ then buyer b no longer considers $\hat{s}$ as a reputable seller, i.e.,

$$S_r^b \leftarrow S_r^b - \{\hat{s}\}. \tag{3.9}$$

- If $(\hat{s} \notin S_r^b)$ and $(r^b(\hat{s}) \geq \Theta)$ then buyer b now considers $\hat{s}$ as a reputable seller, i.e.,

$$S_r^b \leftarrow S_r^b \cup \{\hat{s}\}. \tag{3.10}$$

Finally, the set of disreputable sellers also needs to be updated:

- If $(\hat{s} \notin S_{dr}^b)$ and $(r^b(\hat{s}) \leq \theta)$ then buyer b now considers $\hat{s}$ as a disreputable seller, i.e.,

$$S_{dr}^b \leftarrow S_{dr}^b \cup \{\hat{s}\}. \tag{3.11}$$

Setting μ and ν

The co-operation and non-cooperation factors, μ and ν, are used to adjust the reputation ratings of sellers once the buyer has examined the quality of the good

[5]Buyer b will consider seller s as being *non-cooperative* if the good $\hat{s}$ sells to b has value less than that demanded by b.

purchased.

To protect itself from dishonest sellers, buyer b may require $|\nu| > |\mu|$ to implement the traditional assumption that reputation should be difficult to build up, but easy to tear down. Moreover, buyer b may vary μ and ν as increasing functions of the true product value v^b to reflect the common idea that a transaction with higher value should be more appreciated than a lower one (i.e., the reputation rating of a seller that offers higher true product value should be better increased and vice versa).

In particular, we propose the following equations for the calculation of μ and ν. If $v^b(g,p,q) - \vartheta^b(g) \geq 0$, we define the cooperation factor μ as

$$\mu = \begin{cases} \dfrac{v^b(g,p,q) - \vartheta^b(g)}{\Delta v^b} & \text{if} \quad \frac{v^b(g,p,q) - \vartheta^b(g)}{\Delta v^b} > \mu_{min}, \\ \mu_{min} & \text{otherwise,} \end{cases} \tag{3.12}$$

where $\Delta v^b = v^b_{max} - v^b_{min}$ with v^b_{max} and v^b_{min} being the maximum and minimum value of the true product value function $v^b(g,p,q)$[6]. We prevent μ from becoming zero when $v^b(g,p,q) = \vartheta^b(g)$ by using the value μ_{min}.

However, if $v^b(g,p,q) - \vartheta^b(g) < 0$, we define the non-cooperation factor ν as

$$\nu = \lambda(\frac{v^b(g,p,q) - \vartheta^b(g)}{\Delta v^b}), \tag{3.13}$$

where λ is called the *penalty factor* ($\lambda > 1$) to implement the above mentioned idea that $|\nu|$ should be greater than $|\mu|$. If applying equation (3.8) using ν as defined in (3.13) results in the updated value $r^b(\hat{s}) \leq -1$, that is, seller $\hat{s}$ is so non-cooperative, then buyer b will place $\hat{s}$ in the disreputable set S^b_{dr} by setting $r^b(\hat{s}) = \theta$.

Let us now look at the proposed learning algorithm for sellers.

[6] v^bmax and v^b_{min} are derived from the maximum and minimum elements of the finite sets P and Q. See the subsection labelled "True Product Value Function v^b" under Section 3.4 for an example.

3.2.2 Selling Algorithm

Consider the scenario where a seller $s \in S$ has to decide on the price to sell some good g to a buyer b. Let B be the (finite) set of buyers in the marketplace, and let function $h^s : G \times P \times B \mapsto \mathbb{R}$ estimate the expected profit for seller s. Thus, the real number $h^s(g, p, b)$ represents the expected profit for seller s if it sells good g at price p to buyer b. Let $c^s(g, b)$ be the cost of seller s to produce good g for buyer b. Note that seller s may produce various versions of good g, which are tailored to meet the needs of different buyers. Seller s will choose a price $\hat{p}$ greater than or equal to cost $c^s(g, b)$ to sell good g to buyer b such that its expected profit is maximized:

$$\hat{p} = \arg \max_{\substack{p \in P \\ p \geq c^s(g,b)}} h^s(g, p, b), \tag{3.14}$$

where in this case arg is an operator such that $\arg h^s(g, p, b)$ returns p.

The expected profit function h^s is learned incrementally using reinforcement learning:

$$h^s(g, p, b) \leftarrow h^s(g, p, b) + \alpha(\phi^s(g, p, b) - h^s(g, p, b)), \tag{3.15}$$

where $\phi^s(g, p, b)$ is the actual profit of seller s if it sells good g at price p to buyer b, and is defined as follows:

$$\phi^s(g, p, b) = \begin{cases} p - c^s(g, b) & \text{if seller } s \text{ wins the auction,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Thus, if seller s does not win the auction then $(\phi^s(g, p, b) - h^s(g, p, b))$ is negative, and by (3.15), $h^s(g, p, b)$ is updated with a smaller value than before. This means that price $\hat{p}$ will probably not be chosen again to sell good g to buyer b in future auctions, but rather some lower price will. Conversely, if seller s wins the auction then price $\hat{p}$ will probably be re-selected in future auctions.

If seller s succeeded in selling good g to buyer b once, but subsequently fails for a number of auctions, say for m consecutive auctions (where m is seller s specific constant), then it may not only be because s has set a too high price for good g, but probably also because the quality of g does not meet buyer b's expectation. Thus, in addition to lowering the price via equation (3.15), seller s may optionally add more value (quality) to g by increasing its production cost[7]:

$$c^s(g,b) \leftarrow (1 + Inc)c^s(g,b), \tag{3.17}$$

where Inc is seller s specific constant called the *quality increasing factor*.

In contrast, if seller s is successful in selling good g to buyer b for n consecutive auctions, it may optionally reduce the quality of good g, and thus try to further increase its future profit:

$$c^s(g,b) \leftarrow (1 - Dec)c^s(g,b), \tag{3.18}$$

where Dec is seller s specific constant called the *quality decreasing factor*.

3.2.3 An Example

For the purpose of illustrating how the proposed algorithms work, we provide in this subsection a simplified numerical example including simple buying and selling situations, respectively.

Buying Situation

Consider a simple buying situation where a buyer b announces its need of some good g. Suppose that there are 6 sellers in the marketplace, namely

$$S = \{s_i \mid i = 1 \ldots 6\},$$

[7]This supports the common assumption that it costs more to produce high quality goods.

and that the sets of reputable and disreputable sellers to buyer b are

$$S_r^b = \{s_j \mid j = 1 \ldots 3\} \subset S,$$

and

$$S_{dr}^b = \emptyset.$$

Furthermore, suppose $\Theta = 0.4$, $\theta = -0.8$, $v^b(g, p, q) = 2.5q - p$, $\alpha = 0.8$, $\vartheta^b(g) = 6.10$. For the purpose of simplicity, assume the cooperation factor μ and the non-cooperation factor ν to be constants, with $\mu = 0.2$ and $\nu = -0.4$. Let the reputation ratings $r^b(s_i)$ be given as follows:

s_i	s_1	s_2	s_3	s_4	s_5	s_6
$r^b(s_i)$	0.40	0.45	0.50	0.30	0.25	0.20

Table 3.1: Reputation ratings of different sellers to buyer b.

After buyer b's announcement of its request for good g, the sellers bid with the following prices to deliver good g to buyer b:

s_i	s_1	s_2	s_3	s_4	s_5	s_6
p	4	5	4.5	4	5	3.5

Table 3.2: Prices offered by different sellers for good g.

Assume that buyer b's expected values of buying good g at various prices from different sellers are

s_i	s_1	s_2	s_3	s_4	s_5	s_6
p	4	5	4.5	4	5	3.5
$f^b(g, p, s_i)$	6.15	7.25	6.65	5.50	5.75	5.20

Table 3.3: Buyer b's expected value of buying good g at various prices from different sellers.

Then, by equation (3.3), buyer b buys good g from seller s_2 at price $p = 5$ with

$$f^b(g, p, s_2) = 7.25 = \max_{s \in S_r^b} f^b(g, p, s).$$

Suppose buyer b examines the quality q of good g and finds that $q = 5$. It then calculates the true value of good g:

$$v^b(g, p, q) = 2.5q - p = 2.5(5) - 5 = 7.50.$$

Buyer b now updates its expected value function using equations (3.5) and (3.6):

$$\begin{aligned} \Delta &= v^b(g, p, q) - f^b(g, p, s_2) \\ &= 7.50 - 7.25 = 0.25, \text{ and} \\ f^b(g, p, s_2) &\leftarrow f^b(g, p, s_2) + \alpha\Delta \\ &\leftarrow 7.25 + (0.80)(0.25) = 7.45. \end{aligned}$$

Finally, since $v^b(g, p, q) - \vartheta^b(g) = 7.50 - 6.10 \geq 0$, buyer b increases the reputation rating $r^b(s_2)$ of seller s_2 according to equation (3.7):

$$\begin{aligned} r^b(s_2) &\leftarrow r^b(s_2) + \mu(1 - r^b(s_2)) \\ &\leftarrow 0.45 + (0.20)(1 - 0.45) = 0.56. \end{aligned}$$

Thus, by providing good g with high value, seller s_2 has improved its reputation to buyer b and remained in the set S_r^b of reputable sellers to b.

Selling Situation

Consider how a seller in the above-mentioned marketplace, say seller s_4, behaves according to the proposed selling algorithm. Suppose $c^{s_4}(g, b) = 2.5$, $\alpha = 0.8$, and $Inc = Dec = 0.1$.

Upon receiving buyer b's announcement of its request for good g, seller s_4 has to decide on the price to sell g to b. Assume that seller s_4's expected profits to sell good g to buyer b at various prices are

p	2.5	2.75	3.0	3.25	3.5	3.75	4.0	4.25	4.5
$h^{s_4}(g, p, b)$	0.00	0.25	0.50	0.75	1.00	1.25	1.50	0.00	0.00

Table 3.4: Expected profits of seller s_4 in selling good g to buyer b at different prices.

Table 3.4 indicates that seller s_4 does not expect to be able to sell good g to buyer b at price $p \geq 4.25$. By equation (3.14), seller s_4 chooses price $\hat{p} = 4$ to sell good g to buyer b:

$$\hat{p} = \arg \max_{\substack{p \in P \\ p \geq c^s(g,b)}} h^{s_4}(g, p, b) = 4.$$

Since buyer b chooses to buy good g from another seller, namely s_2, the actual profit of seller s_4 is zero, i.e., $\phi^{s_4}(g, \hat{p}, b) = 0$. Hence, seller s_4 updates its expected profit using equation (3.15) as follows:

$$\begin{aligned} h^{s_4}(g, \hat{p}, b) &\leftarrow h^{s_4}(g, \hat{p}, b) + \alpha(\phi^{s_4}(g, \hat{p}, b) - h^{s_4}(g, \hat{p}, b)) \\ &\leftarrow 1.50 + (0.80)(0 - 1.50) = 0.30. \end{aligned}$$

Thus, according to equation (3.14), the price $p = 4$ will not be chosen again (but rather the lower one $p = 3.75$) to sell good g to buyer b in future auctions.

Assume that seller s_4 has failed to sell good g to buyer b for a number of auctions. It therefore decides to add more quality to good g by increasing its production cost using equation (3.15):

$$\begin{aligned} c^s(g,b) &\leftarrow (1 + Inc)c^s(g,b) \\ &\leftarrow (1 + 0.10)(2.5) = 2.75. \end{aligned}$$

By doing so, seller s_4 hopes that good g may now meet buyer b's demanded value and that it will be able to sell g to b in future auctions.

See Appendix A for a more specific example clarifying the auction process in an information good environment, and Appendix B for a glossary of all the mathematical symbols used in our proposed algorithms.

3.3 Worst Case Scenario

In our proposed buying algorithm, reinforcement learning and reputation modelling are used in combination as two layers of learning to enhance buyers' performance. Naturally, we are interested in addressing the question of whether or not the proposed reputation mechanism can protect a buyer from being harmed infinitely by a dishonest seller. In other words, we would like to investigate the worst case scenario of a proposed buyer caused by a dishonest seller.

We start by mathematically defining the concepts of *gain*, *loss*, and *being better off* of a buyer in order to lay the necessary ground of terminology. We then prove two propositions that together address the issue in consideration. In particular,

we show that if a proposed buyer b is *cautious* in setting its penalty factor, the maximum loss that b may incur due to the non-cooperative transactions with a dishonest seller s is bounded above by a constant term. This result guarantees that a buyer following our proposed algorithm will not be harmed infinitely by a dishonest seller, and therefore will not incur infinite loss.

For coherency, we make use of in this section the notations that are introduced in Section 3.2.1 to describe our proposed reputation mechanism.

Definition 3.1 *A buyer b is said to gain in a transaction with a seller s if it purchases from s some good g with value v^b greater than or equal to its demanded product value ϑ^b. The difference $(v^b - \vartheta^b)$ is called the gain of b in the transaction*[8]*. In this case, seller s is said to be cooperative with b, value v^b is called s's cooperative value, and the transaction is called the cooperative transaction.*

Definition 3.2 *A buyer b is said to lose in a transaction with a seller s if it purchases from s some good g with value v^b less than its demanded product value ϑ^b. The difference $(\vartheta^b - v^b)$ is called the loss of b in the transaction. In this case, seller s is said to be non-cooperative with b, value v^b is called s's non-cooperative value, and the transaction is called the non-cooperative transaction.*

Definition 3.3 *A buyer b is said to be better off (or satisfied) after a series of transactions with a seller s if its total gain is greater than its total loss in these transactions.*

Proposition 3.1 *Let s be a seller who is cooperative with a buyer b in order to get back to its previous reputation rating after a non-cooperative transaction. Buyer b*

[8]No loss is also considered as a gain.

will be better off if it sets

$$\lambda > \frac{1}{1 - (\frac{\vartheta^b - v_0^b}{\Delta v^b})} \tag{3.19}$$

where, as described in Section 3.2.1, λ is the penalty factor, ϑ^b is b's demanded product value, v_0^b is seller s's non-cooperative value, and $\Delta v^b = v_{max}^b - v_{min}^b$ with v_{max} and v_{min} being the maximum and minimum product values, respectively.

Proof:

Consider the scenario where a proposed buyer b purchases from a seller s some good g with value v_0^b less than b's demanded product value ϑ^b. Let $a = \vartheta^b - v_0^b$ ($a > 0$) be the loss of b in this non-cooperative transaction with s.

According to the proposed reputation mechanism, b will decrease the reputation rating of s using equation (3.8). Let $r_0^b(s)$ and $r_1^b(s)$ be the reputation of s before and after the transaction, respectively. We first consider the case where $r_0^b(s) \geq 0$. By equations (3.8) and (3.13) we have

$$\begin{aligned} r_1^b(s) &= r_0^b(s) + \nu(1 - r_0^b(s)) \\ &= r_0^b(s) - \frac{\lambda a}{\Delta v^b}(1 - r_0^b(s)) \end{aligned} \tag{3.20}$$

Now, we will show that when s is cooperative by offering good g with value $v_1^b \geq \vartheta^b$ to get back to its previous reputation rating $r_0^b(s)$, the gain $(v_1^b - \vartheta^b)$ of b in this transaction will be greater than its loss a in the previous transaction if b sets λ as in (3.19).

To make our notations simpler without loss of generality, from now on we will drop the superscript b and variable s in the notations.

Suppose $r_1 \geq 0$. The situation is shown in Figure 3.2.

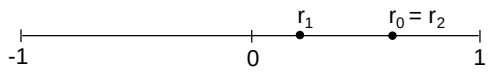

Figure 3.2: The case $r_1 \geq 0$.

By offering value $v_1 > \vartheta$, the reputation rating of s is increased to r_2. According to equations (3.7) and (3.12) we have

$$\begin{aligned} r_2 &= r_1 + \mu(1 - r_1) \\ &= r_1 + \frac{v_1 - \vartheta}{\Delta v}(1 - r_1) \\ &= r_0 - \frac{\lambda a}{\Delta v}(1 - r_0) + \frac{v_1 - \vartheta}{\Delta v}(1 - (r_0 - \frac{\lambda a}{\Delta v}(1 - r_0))) \end{aligned} \tag{3.21}$$

In order for s to get back to its previous reputation position r_0 we must have $r_2 = r_0$, that is

$$\begin{aligned} r_0 - \frac{\lambda a}{\Delta v}(1 - r_0) + \frac{v_1 - \vartheta}{\Delta v}(1 - (r_0 - \frac{\lambda a}{\Delta v}(1 - r_0))) &= r_0 \\ \frac{v_1 - \vartheta}{\Delta v}(1 - (r_0 - \frac{\lambda a}{\Delta v}(1 - r_0))) &= \frac{\lambda a}{\Delta v}(1 - r_0) \\ \frac{v_1 - \vartheta}{\Delta v}(1 - r_0 + \frac{\lambda a}{\Delta v}(1 - r_0))) &= \frac{\lambda a}{\Delta v}(1 - r_0) \\ (v_1 - \vartheta)(1 - r_0 + \frac{\lambda a}{\Delta v}(1 - r_0))) &= \lambda a(1 - r_0) \\ v_1 - \vartheta &= \frac{\lambda a(1 - r_0)}{(1 - r_0) + \frac{\lambda a}{\Delta v}(1 - r_0)} \\ v_1 - \vartheta &= \frac{\lambda a}{1 + \frac{\lambda a}{\Delta v}} \end{aligned} \tag{3.22}$$

$v_1 - \vartheta > a$ iff

$$\begin{aligned} \frac{\lambda a}{1+\frac{\lambda a}{\Delta v}} &> a \\ \lambda a &> a(1+\frac{\lambda a}{\Delta v}) \\ \lambda &> 1+\frac{\lambda a}{\Delta v} \\ \lambda - \frac{\lambda a}{\Delta v} &> 1 \\ \lambda(1-\frac{a}{\Delta v}) &> 1 \\ \lambda &> \frac{1}{1-\frac{a}{\Delta v}} \end{aligned} \tag{3.23}$$

For the case $r_1 < 0$, the situation is shown in Figure 3.3.

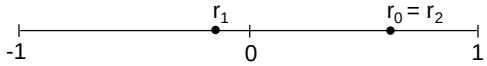

Figure 3.3: The case $r_1 < 0$.

By equations (3.7) and (3.12) we have

$$\begin{aligned} r_2 &= r_1 + \mu(1+r_1) \\ &= r_1 + \frac{v_1 - \vartheta}{\Delta v}(1+r_1) \\ &= r_0 - \frac{\lambda a}{\Delta v}(1-r_0) + \frac{v_1-\vartheta}{\Delta v}(1+(r_0 - \frac{\lambda a}{\Delta v}(1-r_0))) \end{aligned} \tag{3.24}$$

Seller s gets back to its previous reputation position r_0 iff $r_2 = r_0$. That is,

$$\begin{aligned} r_0 - \frac{\lambda a}{\Delta v}(1-r_0) + \frac{v_1-\vartheta}{\Delta v}(1+(r_0-\frac{\lambda a}{\Delta v}(1-r_0))) &= r_0 \\ \frac{v_1-\vartheta}{\Delta v}(1+(r_0-\frac{\lambda a}{\Delta v}(1-r_0))) &= \frac{\lambda a}{\Delta v}(1-r_0) \\ \frac{v_1-\vartheta}{\Delta v}(1+r_0-\frac{\lambda a}{\Delta v}(1-r_0))) &= \frac{\lambda a}{\Delta v}(1-r_0) \\ (v_1-\vartheta)(1+r_0-\frac{\lambda a}{\Delta v}(1-r_0))) &= \lambda a(1-r_0) \\ v_1-\vartheta &= \frac{\lambda a(1-r_0)}{1+r_0-\frac{\lambda a}{\Delta v}(1-r_0)} \end{aligned} \tag{3.25}$$

$v_1 - \vartheta > a$ iff

$$\begin{aligned} \frac{\lambda a(1-r_0)}{1+r_0-\frac{\lambda a}{\Delta v}(1-r_0)} &> a \\ \lambda a(1-r_0) &> a(1+r_0-\frac{\lambda a}{\Delta v}(1-r_0)) \\ \lambda(1-r_0)+\frac{\lambda a}{\Delta v}(1-r_0) &> 1+r_0 \\ \lambda+\frac{\lambda a}{\Delta v} &> \frac{1+r_0}{1-r_0} \\ \lambda(1+\frac{a}{\Delta v}) &> \frac{1+r_0}{1-r_0} \\ \lambda &> \frac{\frac{1+r_0}{1-r_0}}{1+\frac{a}{\Delta v}} \end{aligned} \tag{3.26}$$

Since $r_1 < 0$ we have

$$\begin{aligned} r_0 - \frac{\lambda a}{\Delta v}(1 - r_0) &< 0 \\ r_0 - \frac{\lambda a}{\Delta v} + \frac{\lambda a}{\Delta v} r_0 &< 0 \\ r_0(1 + \frac{\lambda a}{\Delta v}) &< \frac{\lambda a}{\Delta v} \\ r_0 &< \frac{\frac{\lambda a}{\Delta v}}{1 + \frac{\lambda a}{\Delta v}} \\ r_0 &< \frac{\lambda a}{\Delta v + \lambda a} \end{aligned} \tag{3.27}$$

Using (3.27) we have

$$\frac{\frac{1+r_0}{1-r_0}}{1 + \frac{a}{\Delta v}} < \frac{\frac{1+\frac{\lambda a}{\Delta v + \lambda a}}{1-\frac{\lambda a}{\Delta v + \lambda a}}}{1 + \frac{a}{\Delta v}} = \frac{\frac{\Delta v + 2\lambda a}{\Delta v}}{1 + \frac{a}{\Delta v}} = \frac{\Delta v + 2\lambda a}{\Delta v + a} \tag{3.28}$$

Thus (3.26) holds if

$$\begin{aligned} \lambda &> \frac{\Delta v + 2\lambda a}{\Delta v + a} \\ \lambda(\Delta v + a) - 2\lambda a &> \Delta v \\ \lambda(\Delta v + a - 2a) &> \Delta v \\ \lambda(\Delta v - a) &> \Delta v \\ \lambda &> \frac{\Delta v}{\Delta v - a} \\ \lambda &> \frac{1}{1 - \frac{a}{\Delta v}} \end{aligned} \tag{3.29}$$

Combining (3.23) and (3.29), for both cases of r_1 we have

$$\lambda > \frac{1}{1 - \frac{a}{\Delta v}} \tag{3.30}$$

Let us now consider the case where $r_0 < 0$ (Figure 3.4).

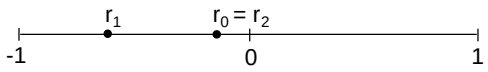

Figure 3.4: The case $r_0 < 0$.

As mentioned before, because of seller s's non-cooperative transaction, buyer b will decrease the reputation rating of s down to r_1 ($-1 < r_1 < 0$). By (3.8) and (3.13) we have

$$\begin{aligned} r_1 &= r_0 + \nu(1 + r_0) \\ &= r_0 - \frac{\lambda a}{\Delta v}(1 + r_0) \end{aligned} \tag{3.31}$$

By offering high value $v_1 > \vartheta$, the reputation rating of s is increased to r_2. Using (3.7) and (3.12) we have

$$\begin{aligned} r_2 &= r_1 + \mu(1 + r_1) \\ &= r_1 + \frac{v_1 - \vartheta}{\Delta v}(1 + r_1) \\ &= r_0 - \frac{\lambda a}{\Delta v}(1 + r_0) + \frac{v_1 - \vartheta}{\Delta v}(1 + r_0 - \frac{\lambda a}{\Delta v}(1 + r_0)) \end{aligned} \tag{3.32}$$

In order for s to get back to its previous reputation position r_0, we set $r_2 = r_0$,

that is

$$\begin{aligned} r_0 - \frac{\lambda a}{\Delta v}(1+r_0) + \frac{v_1-\vartheta}{\Delta v}(1+r_0-\frac{\lambda a}{\Delta v}(1+r_0)) &= r_0 \\ \frac{v_1-\vartheta}{\Delta v}(1+r_0-\frac{\lambda a}{\Delta v}(1+r_0)) &= \frac{\lambda a}{\Delta v}(1+r_0) \\ v_1-\vartheta &= \frac{\lambda a(1+r_0)}{(1+r_0)-\frac{\lambda a}{\Delta v}(1+r_0)} \\ v_1-\vartheta &= \frac{\lambda a}{1-\frac{\lambda a}{\Delta v}} \end{aligned} \tag{3.33}$$

$v_1 - \vartheta > a$ iff

$$\frac{\lambda a}{1-\frac{\lambda a}{\Delta v}} > a \tag{3.34}$$

Since $r_1 \in (-1, 0)$, we have

$$\begin{aligned} r_0 - \frac{\lambda a}{\Delta v}(1+r_0) &> -1 \\ \frac{\lambda a}{\Delta v}(1+r_0) &< 1+r_0 \\ \frac{\lambda a}{\Delta v} &< 1 \\ 1-\frac{\lambda a}{\Delta v} &> 0 \end{aligned} \tag{3.35}$$

Using (3.35), (3.34) is equivalent to

$$\begin{aligned} \lambda a &> a(1-\frac{\lambda a}{\Delta v}) \\ \lambda &> 1-\frac{\lambda a}{\Delta v} \\ \lambda(1+\frac{a}{\Delta v}) &> 1 \\ \lambda &> \frac{1}{1+\frac{a}{\Delta v}} \end{aligned} \tag{3.36}$$

Because

$$\frac{1}{1-\frac{a}{\Delta v}} > \frac{1}{1+\frac{a}{\Delta v}} \tag{3.37}$$

(3.36) will certainly hold if (3.30) holds. In other words, for both cases $r_0 \geq 0$ and $r_0 < 0$, buyer b will be better off in the transactions with seller s if b sets the penalty factor

$$\lambda > \frac{1}{1-\frac{a}{\Delta v}} \tag{3.38}$$

and the proof is complete. □

Corollary 3.1 *Let s be a seller who is cooperative with a buyer b in order to get back to its previous reputation rating after a non-cooperative transaction. For all non-cooperative values v of s, buyer b will be better off if it sets*

$$\lambda > \frac{1}{1-(\frac{\vartheta - v_{min}}{\Delta v})} \tag{3.39}$$

where, as described in Section 3.2.1, λ is the penalty factor, ϑ is b's demanded product value, and $\Delta v = v_{max} - v_{min}$ with v_{max} and v_{min} being the maximum and minimum product values, respectively.

Proof:

For all non-cooperative values v, we have $\vartheta - v_{min} \geq \vartheta - v$, and therefore

$$\frac{1}{1-(\frac{\vartheta - v_{min}}{\Delta v})} \geq \frac{1}{1-(\frac{\vartheta - v}{\Delta v})} \tag{3.40}$$

It follows that for all non-cooperative values v, Proposition 3.1 holds if b sets λ as in (3.39), and the corollary follows directly. □

In practice, we find that a buyer b is sufficiently satisfied when it sets

$$\lambda \approx \frac{1}{1-\frac{\vartheta - v_{min}}{\Delta v}} \tag{3.41}$$

Definition 3.4 *A buyer b is said to be cautious if it sets*

$$\lambda > \frac{1}{1 - (\frac{\vartheta - v_{min}}{\Delta v})} \tag{3.42}$$

where, as described in Section 3.2.1, λ is the penalty factor, ϑ is b's demanded product value, and $\Delta v = v_{max} - v_{min}$ with v_{max} and v_{min} being the maximum and minimum product values, respectively.

Proposition 3.2 *The maximum loss of a cautious buyer b is bounded above by*

$$\frac{|\theta|(v_{max} - \vartheta)}{1 + \theta} + (\vartheta - v_{min}) \tag{3.43}$$

where, as described in Section 3.2.1, θ is the disreputation threshold, ϑ is b's demanded product value, and v_{max} is the maximum product value.

Proof:

First, we notice that the loss of buyer b will be reduced when a seller s decides to be cooperative in a transaction, even in order to further behave non-cooperatively in the following transactions. In particular, let r be the current reputation rating of s. Suppose somehow s realizes that its reputation rating r is close to b's disreputation threshold θ and therefore decides to alternate between cooperative and non-cooperative actions, in order to avoid being placed in the disreputable set. Let r' be the reputation rating of s resulting from its cooperative behaviour. The situation is illustrated in Figure 3.5.

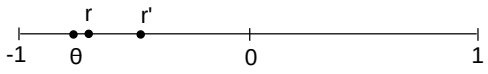

Figure 3.5: Buyer b's loss will be reduced if seller s decides to cooperate even in order to be non-cooperative in following transactions.

Let d be the gain of b in that cooperative transaction. Let l be the loss of b when s alternatively does not cooperate resulting in its reputation rating being moved back to r. Since b is cautious, by Corollary 3.1, $d > l$. Hence, the loss so far of b is reduced by $(d - l)$ after these two transactions with s. It can therefore be inferred that the more s continues to alternate between cooperative and non-cooperative actions, the more the loss of b will be reduced.

It is now clear that the greatest loss of a buyer b is caused by a seller s who is continuously non-cooperative until its reputation rating is arbitrarily close to the disreputation threshold θ, when it performs the final non-cooperative transaction with minimum value v_{min}, resulting in buyer b's loss $(\vartheta - v_{min})$, the greatest loss that b may incur in a transaction.

Let s be such a seller with initial reputation rating $r_0 = 0$. Let $v_0, v_1, \ldots, v_{n-1}$ be the non-cooperative values that s continuously offers until its reputation rating is arbitrarily close to θ. Let $r_1, r_2, \ldots, r_n$ be the reputation ratings of s resulting from the offerings of $v_0, v_1, \ldots, v_{n-1}$, respectively. So, r_n approaches θ from the right and $r_n \approx \theta$ (Figure 3.6).

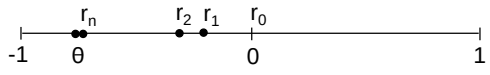

Figure 3.6: Seller s is consecutively non-cooperative.

Let L be the total loss of be caused by this series of non-cooperative transactions. Then,

$$L = \sum_{i=0}^{n-1} (\vartheta - v_i) \tag{3.44}$$

By (3.8) and (3.13) we have

$$
\begin{aligned}
r_1 &= 0 + \lambda(\frac{v_0 - \vartheta}{\Delta v})(1-0) = -\lambda(\frac{\vartheta - v_0}{\Delta v})(1+0) \\
r_1 &< -\lambda(\frac{\vartheta - v_0}{\Delta v})(1+\theta) \\
r_2 &= r_1 + \lambda(\frac{v_1 - \vartheta}{\Delta v})(1+r_1) = r_1 - \lambda(\frac{\vartheta - v_1}{\Delta v})(1+r_1) \\
r_2 &< -\lambda(\frac{\vartheta - v_0}{\Delta v})(1+\theta) - \lambda(\frac{\vartheta - v_1}{\Delta v})(1+\theta) \\
r_2 &< -\lambda(\frac{1+\theta}{\Delta v})[(\vartheta - v_0) + (\vartheta - v_1)] \\
r_3 &= r_2 + \lambda(\frac{v_2 - \vartheta}{\Delta v})(1+r_2) = r_2 - \lambda(\frac{\vartheta - v_2}{\Delta v})(1+r_2) \\
r_3 &< -\lambda(\frac{1+\theta}{\Delta v})[(\vartheta - v_0) + (\vartheta - v_1)] - \lambda(\frac{\vartheta - v_2}{\Delta v})(1+\theta) \\
r_3 &< -\lambda(\frac{1+\theta}{\Delta v})[(\vartheta - v_0) + (\vartheta - v_1) + (\vartheta - v_2)] \\
&\vdots \\
r_n &< -\lambda(\frac{1+\theta}{\Delta v})[(\vartheta - v_0) + (\vartheta - v_1) + (\vartheta - v_2) + \ldots + (\vartheta - v_{n-1})] \\
r_n &< -\lambda(\frac{1+\theta}{\Delta v})L \quad \text{(using (3.44))}
\end{aligned}
\tag{3.45}
$$

Since r_n approaches θ from the right and $r_n \approx \theta$, (3.45) gives

$$
\begin{aligned}
-\lambda(\frac{1+\theta}{\Delta v})L &> \theta \\
\lambda(\frac{1+\theta}{\Delta v})L &< -\theta = |\theta| \\
L &< \frac{|\theta|\Delta v}{\lambda(1+\theta)}
\end{aligned}
\tag{3.46}
$$

Since b is a cautious buyer, by Definition 3.4 we have

$$
\lambda > \frac{1}{1 - (\frac{\vartheta - v_{min}}{\Delta v})} \tag{3.47}
$$

We notice that

$$\begin{aligned}\frac{1}{1-(\frac{\vartheta-v_{min}}{\Delta v})} &= \frac{1}{\frac{\Delta v-\vartheta+v_{min}}{\Delta v}} \\ &= \frac{\Delta v}{v_{max}-v_{min}-\vartheta+v_{min}} \\ &= \frac{\Delta v}{v_{max}-\vartheta}\end{aligned} \tag{3.48}$$

So, (3.47) and (3.48) give us

$$\lambda > \frac{\Delta v}{v_{max}-\vartheta} \tag{3.49}$$

Thus, (3.46) becomes

$$\begin{aligned}L &< \frac{|\theta|\Delta v}{(\frac{\Delta v}{v_{max}-\vartheta})(1+\theta)} \\ L &< \frac{|\theta|\Delta v(v_{max}-\vartheta)}{\Delta v(1+\theta)} \\ L &< \frac{|\theta|(v_{max}-\vartheta)}{1+\theta}\end{aligned} \tag{3.50}$$

Adding the final loss of $(\vartheta - v_{min})$, the maximum loss of b is bounded above by

$$\frac{|\theta|(v_{max}-\vartheta)}{1+\theta} + (\vartheta - v_{min}) \tag{3.51}$$

The proof is therefore complete. □

Thus, letting b be a cautious buyer as defined in definition 3.4, we have shown that the worst case scenario of b in dealing with a seller s is where s continuously behaves non-cooperatively until its reputation rating $r^b(s)$ is arbitrarily close to the disreputation threshold θ, when it performs the final non-cooperative transaction with minimum value v_{min}, resulting buyer b's loss $(\vartheta - v_{min})$. The maximum loss of the buyer is, however, bounded above by the constant term as shown in (3.51).

3.4 Discussion on Parameters

In this section we discuss the roles of several parameters used in our proposed algorithms and provide some general guidelines to choose them.

Reputation Threshold Θ

The reputation threshold Θ ($0 < \Theta < 1$) is a buyer b's specific constant, which buyer b uses to determine whether it should consider a seller s as a reputable seller. Consequently, the stricter (or more conservative) b is, the higher value it would choose for Θ. In addition, the more untrustful the market environment is, the higher the value b should set Θ to. As the range of Θ is $(0, 1)$, a buyer b of medium strictness acting in a market of medium trust probably chooses Θ to be 0.50. This explains why we used this value for Θ in our experiments (described in the next chapter).

Disreputation Threshold θ

The disreputation threshold θ ($-1 < \theta < 0$) is also buyer b's specific constant. Buyer b uses this constant to decide whether a seller s should be rated as a disreputable seller. Obviously, if b chooses θ to be too low, dishonest sellers will not be placed in the disreputable set as they should be, resulting in b's frequently purchasing unsatisfactory value goods. In contrast, if buyer b sets θ to be too high, more sellers will be placed in the set of disreputable sellers, with the extreme case where all sellers in the market are rated as disreputable sellers. Moreover, due to the fact that b will not re-select disreputable sellers to do business with according to the proposed algorithm, θ should be set low enough in order for b to avoid situations where it may carelessly place a seller s in the disreputable set without having

enough evidence of s's being non-cooperative. This also gives those sellers, who are willing to improve their products, opportunities to make good offers to b. Considering these reasons, we suggest that θ should take values in the range $[-0.9, -0.7]$. In fact, in our experiments we set $\theta = -0.9$ to make sure that a seller is placed in the disreputable set only when it is a really non-cooperative or dishonest seller and therefore deserves that treatment.

True Product Value Function v^b

Each buyer b has its own way to evaluate the good it purchases using the true product value function v^b. Basically, v^b is a function of the price p that buyer b pays for the good, and also of the quality q that b examines the good after receiving it from the seller. Buyer b formulates v^b based on its idea of the relative importance of these two factors. For example, if b considers quality to be more important than price, it may set $v^b = aq - p$ with $a > 1$.

- Although the product quality q is represented by a single numerical value, it could be a multi-faceted concept. That is, buyer b may judge the quality of a product based on a combination of various factors such as physical product characteristics, whether the product is distributed on time, whether the product is supported after purchase etc. As such, buyer b may calculate q as a weighted sum of these factors.

- Since p and q are elements in the finite sets of prices and quality values respectively, there exist the maximum and minimum values (v^b_{max} and v^b_{min}) of the true product value function v^b. If we continue with the above-mentioned example then $v^b_{max} = aq_{max} - p_{min}$ and $v^b_{min} = aq_{min} - p_{max}$. The existence of v^b_{max} and v^b_{min} justifies their use in equations (3.12) and (3.13).

Demanded Product Value ϑ^b

After a transaction with a seller s, buyer b needs to decide if it should increase (or decrease) the reputation rating of s, based on whether or not the true value v^b of the good offered by s meets buyer b's demanded product value ϑ^b. In other words, the demanded product value ϑ^b serves as buyer b's threshold for the true product value v^b. Let us give an example of how a buyer b may calculate ϑ^b. For a particular good g, buyer b should have in its mind the lowest quality q^b_{low} that it would like g at least to have, and the highest price p^b_{high} that it would agree to pay for that quality. Buyer b then can calculate the demanded product value ϑ^b based on q^b_{low} and p^b_{high} using the true product value function v^b. If we reuse the above example of v^b then buyer b will calculate its demanded product value as follows: $\vartheta^b = aq^b_{low} - p^b_{high}$. This choice of ϑ^b means that in order to satisfy buyer b's demand, seller s will have to offer good g with quality greater than q^b_{low} if it intends to sell good g to buyer b at price greater than p^b_{high}.

Exploration Probability ρ

The exploration probability ρ allows buyer b to discover new reputable sellers by, at probability ρ, considering choosing a seller from the set of non-disreputable sellers (rather than the smaller set of reputable sellers). That is, in addition to the reputable sellers, buyer b will also consider the sellers that are neither reputable nor disreputable. Since these are sellers whose reputation buyer b has not yet had enough information to decide on, some of them may have the potential of becoming reputable sellers. Certainly, buyer b needs to explore at probability $\rho = 1$ at the beginning, because at this point b does not have reputation information of any seller in the market and its set of reputable sellers is empty. However, as b is able

to build up some members for its reputable set after a number of transactions, it should exploit the market more and explore it less. That means, b should gradually decrease ρ over time down to some fixed, minimum value ρ_{min}. Of course, there is a trade off in choosing a value for ρ_{min}. The higher ρ_{min}, the more opportunities to explore but the fewer chances to exploit. In marketplaces where new sellers often enter the market, ρ_{min} may be set to as high a value as 0.3, i.e., b will explore the market 30% of the time. However, in marketplaces where new sellers rarely join the market, ρ_{min} should be set to low value (e.g., from 0.05 to 0.10). In fact, we set $\rho_{min} = 0.10$ in our experimentation.

Learning Rate α

As suggested by its name, the learning rate α influences the rate buyer b learns its expected value function f^b, and the rate seller s learns its expected profit function h^s. Let us just look at the case of b since the case of s is similar. Initially, buyer b just stores some initialized, incorrect values of f^b in its internal database. Thus, it needs to quickly update those initialized values with the actual ones by setting $\alpha = 1$. Over time, as b has roughly learned what the values of f^b should be, only small fractions of the current values of f^b need to be used for adjusting the previous values; that means, α should be decreased. In fact, reducing α over time will help the reinforcement learning method to converge [54, 65]. In our experimentation, we gradually decrease α over time from the starting value of 1 down to $\alpha_{min} = 0.1$.

3.5 Chapter Summary

In this chapter we describe an agent market model that is suitable for e-commerce applications. The agent environment is modelled as an open marketplace that

allows its participating buying and selling agents to enter or leave the environment at will. The process of buying and selling goods between agents is realized via a three-phase mechanism similar to the contract net protocol. Our market model takes into account the fact that the quality of a good offered by different selling agents may not be the same, and that a selling agent may alter the quality of its goods. It also considers the possibility of having dishonest selling agents in the market.

We then present our proposed reinforcement learning and reputation based algorithms for buying and selling agents, respectively. Our buying agents learn to maximize their expected values of goods using reinforcement learning. In addition, they model and exploit the reputation of selling agents to avoid interaction with the dishonest ones, and therefore to reduce the risk of purchasing low value goods. Our selling agents learn to maximize their expected profits by using reinforcement learning to adjust prices for and by providing more customized value to their goods. We illustrate the proposed algorithms with a simplified numerical example.

We also investigate the question of whether or not a dishonest seller can infinitely harm a *cautious* buyer (as defined in Definition 3.4), and if not, what the upper bound of the buyer's maximum loss would be in the worst case scenario. We address this question by defining the value gain and value loss of a buyer, and then prove two propositions which together show that the maximum loss of a cautious buyer is bounded above by the constant shown in (3.43). The significance of this result is that our proposed buyers will not be harmed infinitely by dishonest sellers and therefore will not incur infinite value loss, if they are cautious in setting their penalty factor λ according to equation (3.42).

Finally, we discuss the parameters used in the proposed algorithms and provide some general guidelines on how to choose these parameters.

Chapter 4

Experimental Evaluation

In this chapter we experimentally evaluate our model by simulating electronic marketplaces populated with buying and selling agents. In particular, we would like to investigate the microscopic behaviours of the participant agents and the macroscopic behaviours of the market as a whole.

On the micro level, we are interested in examining the individual benefits of agents, particularly their level of satisfaction. Our first aim is to show that an agent that uses reinforcement learning will fare better than an agent that does not make use of any learning method. Next and of greater interest to us, we would like to confirm that in both modest and large-sized marketplaces, buyers and sellers following the proposed algorithms achieve better satisfaction than buyers and sellers who only use reinforcement learning, but the buyers do not model sellers' reputation and the sellers do not consider adjusting the quality of their goods.

On the macro level, we study how a market populated with our buyers and sellers would behave as a whole. In particular, we are interested in knowing if such a market would reach an equilibrium state where the agent population remains

stable (as some sellers who repeatedly fail to sell their goods may decide to leave the market), and if so, how beneficial this equilibrium would be for the participant agents.

To address the micro behaviours issue, we compare the satisfaction of buyers and sellers following the proposed algorithms with that of buyers and sellers who only use reinforcement learning with the buyers not modelling sellers' reputation and the sellers not altering the quality of their goods. Since the higher product value a buyer receives the better satisfied it is, and the higher profit a seller makes the more satisfied it is, we use the true product value v^b and the actual profit ϕ^s as the criteria for comparing the satisfaction level of buyers and sellers, respectively. Thus, we run simulations to record and compare the true product values obtained by a buyer using the proposed algorithm with those obtained by a buyer not modelling sellers' reputation, after they each have made the same number of purchases in the same marketplace. Similarly, we record and compare the actual profits made by a seller following the proposed algorithm and by a seller not considering adjusting product quality, after these two sellers have participated in the same number of auctions in the same marketplace. We simulate both modest and large sized marketplaces in order to know whether the size of a marketplace would influence the level of satisfaction of its agents.

To study the macro behaviours of the market, we simulate a marketplace populated with buyers and sellers following our proposed buying and selling algorithms, respectively. We periodically record the seller population in the market, assuming that those sellers who are no longer able to make sales will decide to leave the market. We examine the graph of seller population varying against the number of auctions to determine if an equilibrium state has been reached. We also investigate if an obtained equilibrium would be beneficial to the participating agents in terms

of their satisfaction.

The simulated marketplaces presented in this chapter have been implemented using Java 2 and run on a Dell Dimension XPS T500 workstation powered with a regular 500 MHz Pentium processor, 128 MB RAM, and Windows NT 4.0 platform.

This chapter is organized as follows: Section 4.1 presents the experiments and results, regarding the micro behaviours of participant agents in modest sized marketplaces as well as large sized ones. Section 4.2 describes the macro behaviours of our market model. Section 4.3 discusses the lessons learned from the experimentation and finally, Section 4.4 provides a summary for the chapter.

4.1 Micro Behaviours

In this section we study the micro behaviours of buying and selling agents participating in modest and large sized marketplaces.

The experimental results reported in this section are based on the average of 100 runs, each of which has 5000 auctions.

4.1.1 Modest Sized Marketplaces

We present three experiments on modest sized marketplaces populated with 8 sellers and 4 buyers. The first experiment aims to show that an agent using reinforcement learning should fare better than an agent not using any learning method. The second experiment confirms that in a marketplace where there are sellers changing the quality of their goods, a buyer following the proposed buying algorithm should obtain a greater level of satisfaction (than that obtained by a buyer using reinforcement learning but not modelling sellers' reputation). The third experiment

demonstrates that in a marketplace where buyers make use of a learning strategy, a seller following the proposed selling algorithm should achieve better satisfaction than a seller using reinforcement learning but not considering adjusting the quality of its goods.

Reinforcement Learning Agents vs. Non-Learning Agents

We would like to confirm that an agent using reinforcement learning should obtain better satisfaction than an agent not making use of any learning method. In particular, we compare the satisfaction level of a buyer using reinforcement learning for selecting sellers with that of a buyer selecting sellers at random. For this purpose, we set up the marketplace such that, among the 4 buyers, buyer b_0 and b_1 choose sellers randomly, while buyer b_2 and b_3 use reinforcement learning for selecting sellers. We let the 8 sellers, namely $s_0, ..., s_7$, offer goods with qualities of 10, 11, 12, 13, 14, 15, 16, and 40, respectively. Other parameters are set as follows:

- The true product value $v^b(g, p, q) = 3.5q - p$, where p and q represent the price and quality of the good g purchased, respectively[1].
- The learning rate α and the exploration probability ρ are both set to 1 initially, and then decreased over time (by factor 0.995) down to $\alpha_{min} = 0.1$ and $\rho_{min} = 0.1$.
- The quality q of a good is chosen to be equal to the cost for producing that good. This choice supports the common assumption that it costs more to produce high quality goods.

[1]This reflects the fact that for the buyers, quality is considerably more important than price.

Since the higher true product values a buyer receives the better satisfied it is, we record and present in Figure 4.1 the average true product values obtained by buyer b_0 and b_1 (graph (i)) and by buyer b_2 and b_3 (graph (ii)).

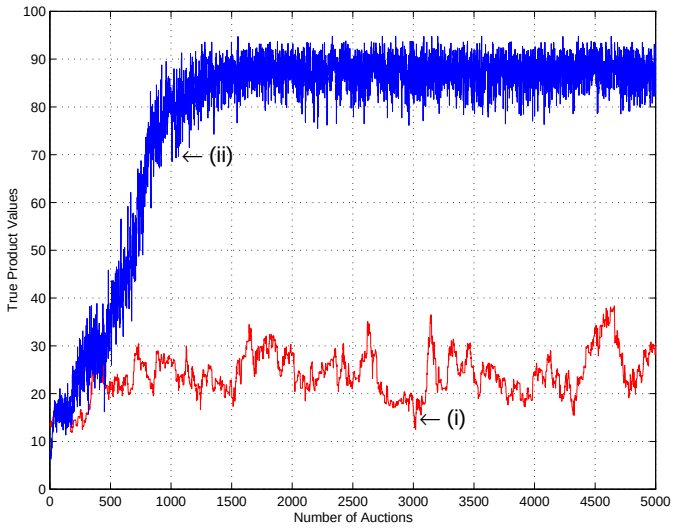

Figure 4.1: Comparison of true product values obtained by a buyer selecting sellers at random (graph (i)), and by a buyer using reinforcement learning (graph (ii)).

It can be observed clearly from Figure 4.1 that the buyer using reinforcement learning achieves a much higher satisfaction level than that achieved by the buyer choosing sellers at random. The reinforcement learning buyer outperforms the buyer selecting sellers at random after just a few hundred auctions. In particular, after about a thousand auctions, the reinforcement learning buyer is able to learn which seller provides the good with the highest value, namely seller s_7, and therefore constantly makes purchases from that seller. As a result, graph (ii) reaches to the range of about 85 and above 90 in the long run, showing that most of the goods purchased by the reinforcement learning buyer have high values. In contrast, the buyer randomly selecting sellers receives goods with much lower values (less than

30 for most of the auctions). The reason is that since this buyer selects sellers at random, it consequently doesn't focus on making purchases from seller s_7, the most valuable seller.

The fact that the buyer using reinforcement learning fares better can also be confirmed by looking at the buyers' histograms of true product values. Figure 4.2(a) shows that the buyer using random strategy obtains very low product values in all of the purchases it makes. For instance, in more than 2000 purchases made, the true product values it obtains are only from 20.0 to 25.0. On the contrary, Figure 4.2(b) demonstrates that the buyer using reinforcement learning achieves much higher product values in most of the purchases it makes. For example, in more than 2500 purchases made, this buyer receives products with the mean value being as high as 90.

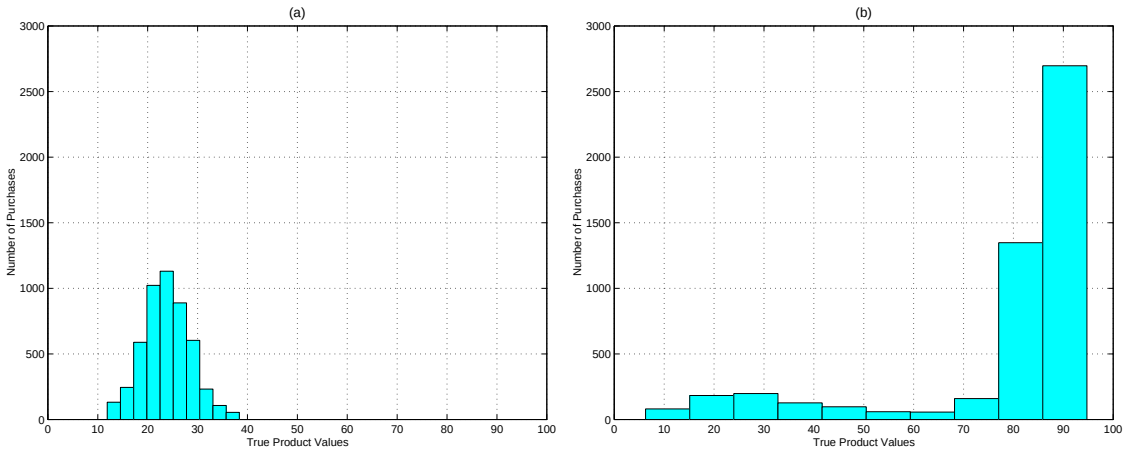

Figure 4.2: Histograms of true product values obtained by a buyer using random strategy (a), and obtained by a buyer using reinforcement learning (b).

Buyers' Satisfaction

In this experiment we would like to show that in a marketplace where there are sellers altering the quality of their goods, a buyer following the proposed buying algorithm should achieve greater satisfaction. Towards this goal, we let buyer b_2 and b_3 follow the proposed buying algorithm, while buyer b_0 and b_1 use reinforcement learning but do not model the reputation of sellers. Among the eight sellers, the first half (namely seller s_0, s_1, s_2, and s_3) offers fixed-quality goods but the second half offers goods with quality altered. We consider two cases in which a seller may change the quality of its goods:

(i) Quality Chosen Randomly: For each auction, the quality of a good is chosen randomly from the interval $[low\_Q, high\_Q]$, where low_Q and $high\_Q$ are seller specific constants.

(ii) Quality Switched between Two Values: The quality is switched between a rather high value and a very low one. This strategy may be used by dishonest sellers who try to attract buyers with high quality goods first and then cheat them with really low quality ones.

For the first case, the true product value v^b, the learning rate α, the exploration probability ρ, and the quality q are chosen as in the first experiment. Other parameters are set as follows:

- The reputation threshold $\Theta = 0.5$, and the disreputation threshold $\theta = -0.9$.
- The demanded product value $\vartheta^b(g) = 102$. Thus, even when a seller has to sell at cost, it must offer goods with quality of at least 40.8 in order to meet the buyers' requirement [2].

[2]Because $v^b(p, q) = 3.5q - p = 3.5(40.8) - 40.8 = 102$.

- If $v^b - \vartheta^b \geq 0$, we define the cooperation factor μ as in equation (3.12):

$$\mu = \begin{cases} \dfrac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} & \text{if} \quad \frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} > \mu_{min}, \\ \mu_{min} & \text{otherwise,} \end{cases} \tag{4.1}$$

  where $\mu_{min} = 0.005$, $v^b_{max} = 3.5q_{max} - p_{min}$, $v^b_{min} = 3.5q_{min} - p_{max}$, $q_{max} = p_{max} = 49.0$, and $q_{min} = p_{min} = 1.0$. In this definition, we vary μ as an increasing function of v^b to reflect the idea that the reputation rating of a seller that offers higher product value should be better increased. We prevent μ from becoming zero when $v^b = \vartheta^b$ by using the value of μ_{min}.

- If $v^b - \vartheta^b < 0$, we define the noncooperation factor ν as in equation (3.13):

$$\nu = \lambda(\frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}}), \tag{4.2}$$

  where we set the penalty factor $\lambda = 3$ as said in (3.41). In this definition, we also vary ν as an increasing function of v^b to support the idea that the lower product value a seller offers, the more its reputation rating should be decreased. The use of factor $\lambda > 1$ indicates that a buyer will penalize a non-cooperative seller λ times greater than it will award a cooperative seller. This implements the traditional assumption that reputation should be difficult to build up, but easy to tear down.

- Sellers s_0, s_1, s_2, and s_3 offer goods with fixed qualities of 32.0, 36.0, 40.0, and 44.0, respectively.

- Sellers s_4, s_5, s_6, and s_7 alter the quality of their goods by, for each auction, choosing at random a quality value in the interval $[low\_Q, high\_Q]$, where $low\_Q = 32.0$ and $high\_Q = 48.0$.

It should be obvious from the settings that a buyer would achieve greater satisfaction by making as many purchases as possible from seller s_3 (who offers the highest fixed product quality) instead of purchasing from those sellers that randomly change the quality of their goods. Table 4.1 shows the average number of purchases made from each seller by buyer b_2 and b_3 - the buyers that follow the proposed buying algorithm (labelled as $b_{2,3}$), and by buyer b_0 and b_1 - the buyers not modelling sellers' reputation (labelled as $b_{0,1}$). Indeed, buyer $b_{2,3}$ made 1086 more purchases from s_3, which is approximately 28.8% of the number of purchases made from s_3 by $b_{0,1}$. Buyer $b_{2,3}$ also made about 684 fewer purchases from those sellers that randomly alter the quality of their goods, which is about 85.8% of the number of purchases made by $b_{0,1}$ from the sellers altering the quality of their goods.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
$b_{0,1}$	134.5	142.5	160.7	**3765.2**	**195.8**	**205.4**	**195.3**	**200.6**
$b_{2,3}$	4.5	7.9	23.3	**4851.2**	**27.6**	**28.6**	**27.9**	**28.9**

Table 4.1: Number of purchases made from different sellers by a buyer not modelling sellers' reputation ($b_{0,1}$), and by a buyer following the proposed algorithm ($b_{2,3}$).

As an alternative view, Figure 4.3 shows the true product values obtained over the number of auctions by the buyer following the proposed buying algorithm (graph (ii)), and by the buyer using reinforcement learning but not modelling sellers' reputation (graph (i)). Clearly, the buyer following the proposed algorithm receives higher product values and therefore achieves better satisfaction. In particular, the mean of true product values obtained by the buyer following the proposed algorithm is 106.71, which is 2.7% higher than that of 103.91 obtained by the buyer not modelling sellers' reputation. In addition, we notice that the buyer following the

proposed algorithm is able to obtain relatively high product values within a short period of time. This shows that modelling sellers' reputation also allows a buyer to quickly identify appropriate sellers in the market and therefore achieve reasonable satisfaction in the very first number of auctions.

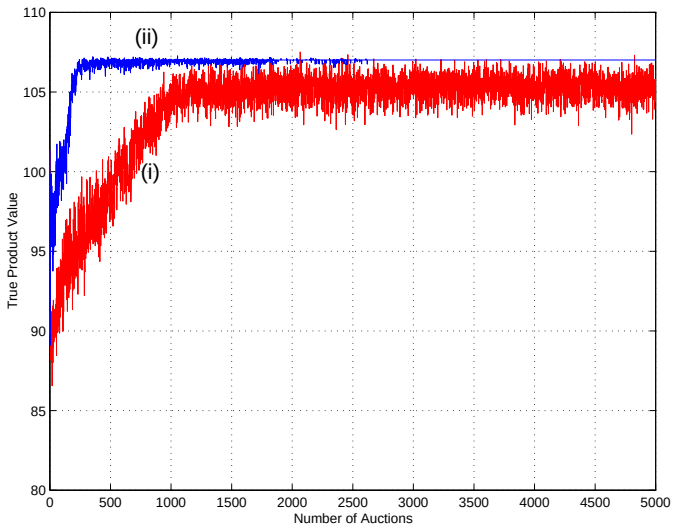

Figure 4.3: Comparison of true product values obtained by a buyer not modelling sellers' reputation (graph (i)), and by a buyer following the proposed algorithm (graph (ii)).

Alternatively, Figure 4.4(a) and (b) present the histograms of true product values obtained by a buyer not exploiting the reputation of sellers and by a buyer following the proposed algorithm, respectively. We notice that the number of purchases in Figure 4.4(b) where the true product values are in the high interval $[107, 109]$ is almost 4000, while that in Figure 4.4(a) is just a few. The number of purchases in Figure 4.4(b) where the true product values are in lower intervals such as $[105, 107]$ is about 1600 purchases less (or 62.7% less) than that in Figure 4.4(a). This indicates that the buyer following the proposed algorithm obtains

more goods with higher value and fewer goods with lower value. In other words, the buyer following the proposed algorithm achieves a better level of satisfaction.

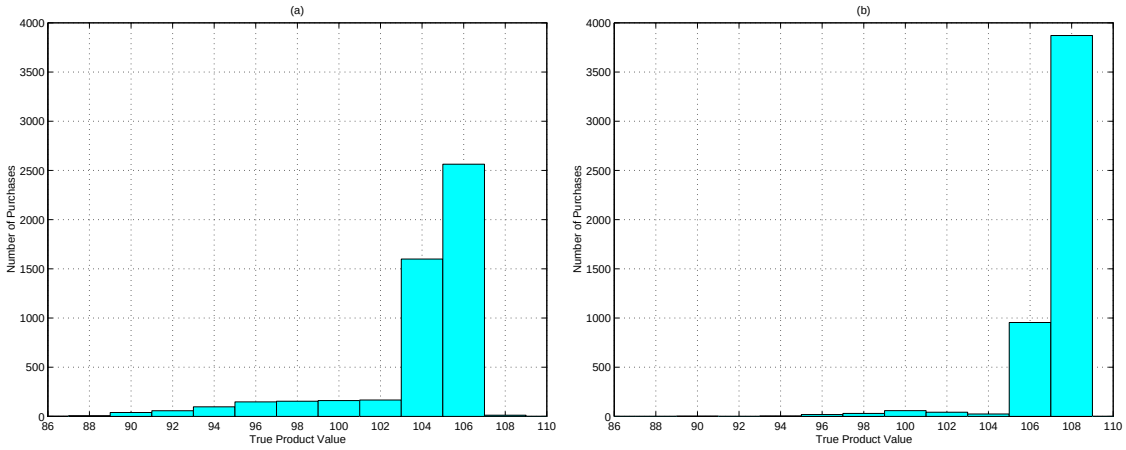

Figure 4.4: Histograms of true product values obtained by a buyer not modelling sellers' reputation (a), and by a buyer following the proposed algorithm (b).

For the second case, we let sellers s_0, s_1, s_2, and s_3 offer goods with fixed qualities of 38.0, 40.0, 42.0, and 44.0, respectively; while sellers s_4, s_5, s_6, and s_7 are made dishonest sellers who offer goods with quality switched between 45.0 and 1.0. Other parameters are the same as in the previous case. It is clear from the settings that a successful buyer should make as many purchases as possible from seller s_3 and try to stay away from the dishonest sellers.

Table 4.2 indeed shows that the buyer following the proposed algorithm ($b_{2,3}$) makes more purchases from seller s_3 but fewer purchases from the dishonest sellers, in comparison with the buyer not modelling sellers' reputation ($b_{0,1}$). In particular, buyer $b_{2,3}$ makes approximately 12.3% more purchases from seller s_3 and about 95.1% fewer purchases from the dishonest sellers, compared to the numbers of purchases buyer $b_{0,1}$ makes from s_3 and from the dishonest sellers, respectively.

Seller	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
$b_{0,1}$	118.1	138.0	151.4	**4263.3**	**88.3**	**79.9**	**78.3**	**82.8**
$b_{2,3}$	11.9	23.5	162.9	**4785.7**	**4.0**	**4.0**	**4.0**	**4.0**

Table 4.2: Number of purchases made from different sellers by a buyer not modelling sellers' reputation ($b_{0,1}$), and by a buyer following the proposed algorithm ($b_{2,3}$).

Figure 4.5 displays the true product values over the number of auctions obtained by the buyer following the proposed buying algorithm (graph (ii)) and by the buyer only using reinforcement learning but not modelling sellers' reputation (graph (i)). Clearly, the figure confirms that the buyer following the proposed algorithm receives higher product values on average and therefore achieves better satisfaction. In particular, our detailed calculation shows that the mean product value obtained by the buyer not modelling sellers' reputation is 101.49 while that obtained by the buyer following the proposed algorithm is 106.27, which is 4.7% higher.

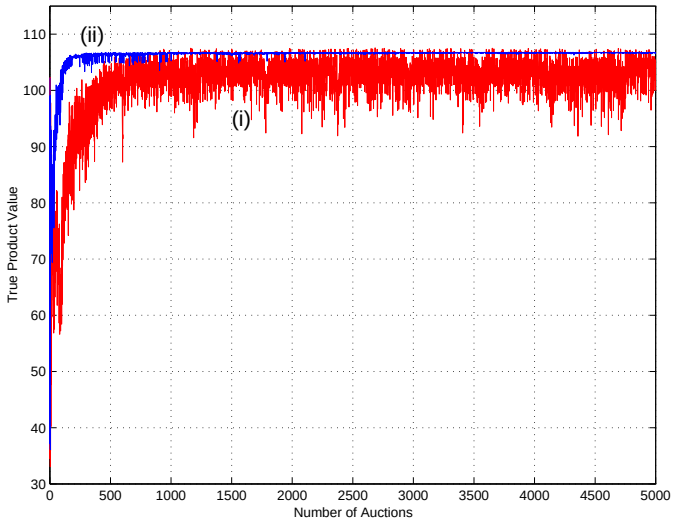

Figure 4.5: Comparison of true product values obtained by a buyer not modelling sellers' reputation (graph (i)), and by a buyer following the proposed algorithm (graph (ii)).

Alternatively, Figure 4.6(a) and (b) provides the histograms of true product values obtained by the buyer not modelling sellers' reputation and by the buyer following the proposed algorithm, respectively.

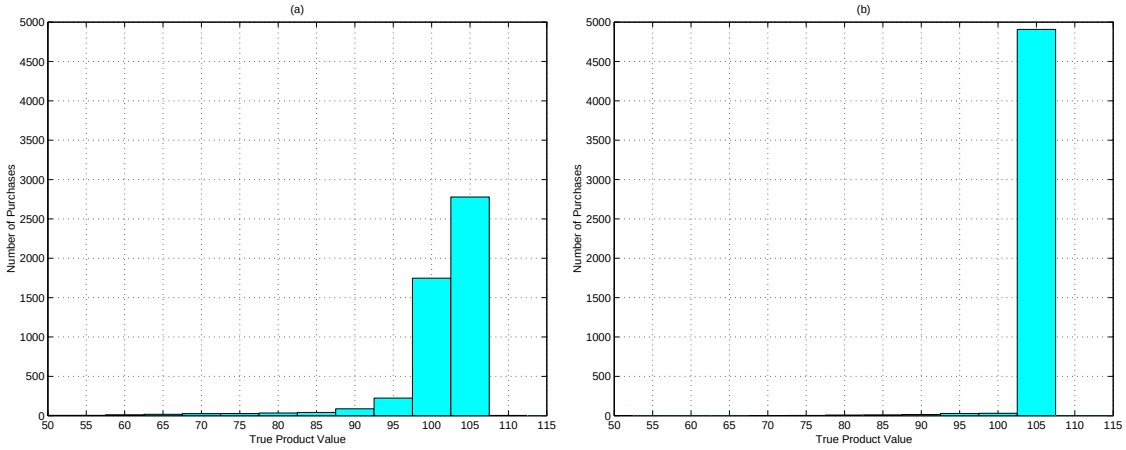

Figure 4.6: Histograms of true product values obtained by a buyer not modelling sellers' reputation (a), and by a buyer following the proposed algorithm (b).

Once again we notice that the buyer following the proposed algorithm is able to purchase more goods with higher values and fewer goods with lower values, and therefore achieves a better level of satisfaction. In fact, the histograms indicate that the buyer following the proposed algorithm purchases about 2150 more goods with approximately mean value of 105 (or 78.2% more) and almost no goods with mean value less than 105, compared to those purchased by the buyer not modelling sellers' reputation.

Sellers' Satisfaction

This experiment aims to demonstrate that in a marketplace where buyers make use of a learning strategy, a seller following the proposed selling algorithm should achieve better satisfaction than a seller using reinforcement learning but not considering altering the quality of its goods. Since the more often a seller is successful in selling its goods to buyers, the higher profit it makes and the better satisfied it

is, we record and compare the number of sales made by a seller following the proposed algorithm with that made by a seller not considering adjusting the quality of its goods. Alternatively, we also compare the actual profits made by these two sellers after they have participated in the same number of auctions in the same marketplace.

To achieve our goal, we let the first seven sellers, namely s_0, s_1, s_2, s_3, s_4, s_5, and s_6, use reinforcement learning and offer goods with fixed qualities of 38.0, 38.5, 39.0, 39.5, 40.0, 40.5, and 41.0, respectively; while we let seller s_7 follow the proposed selling algorithm and offer goods with an initial quality of 38.0. All the four buyers, namely b_0, b_1, b_2, and b_3, follow the proposed buying algorithm. We set the number of consecutive unsuccessful auctions (after which a seller following the proposed algorithm may consider improving the quality of its goods) $m = 10$, and the number of consecutive successful auctions (after which a seller following the proposed algorithm may consider reducing the quality of its goods) $n = 10$. Both the quality increasing factor Inc and the quality decreasing factor Dec are set to 0.05. Other parameters are chosen as in the second experiment.

Table 4.3 below presents the number of sales made by each seller to the four buyers. It can be seen clearly from the table that seller s_7 (the seller that follows the proposed selling algorithm) makes the greatest number of sales among all sellers, and therefore achieves better satisfaction. In particular, the number of sales made by seller s_7 is about 2.33 times greater than that made by seller s_6, the most successful seller among those using reinforcement learning but not considering adjusting product quality. The success of seller s_7 is due to the fact that, although s_7 initially offers goods with rather low quality (of 38.0), it learns to improve the quality of its goods using to the proposed selling algorithm, and therefore becomes a reputable seller to the buyers.

	b_0	b_1	b_2	b_3
s_0	104.78	111.50	113.13	103.25
s_1	107.22	113.26	115.06	105.41
s_2	108.96	116.00	117.98	107.11
s_3	113.49	120.24	120.81	111.09
s_4	119.21	122.87	125.12	115.53
s_5	131.94	137.51	133.34	131.45
s_6	172.79	165.62	162.63	157.93
s_7	**391.61**	**363.00**	**361.93**	**418.23**

Table 4.3: Number of sales made by each seller to the four buyers.

The fact that seller s_7 obtains greater satisfaction than seller s_6 can also be seen by comparing the actual profits made by these two sellers. Figure 4.7 displays the actual profit values made by seller s_6 (graph (i)) and by seller s_7 (graph (ii)) over 5000 auctions, respectively. It is clearly shown in the figure that the profit made by seller s_7 is much higher than that made by seller s_6. In fact, the mean profit value of seller s_7 is 1.2391, which is approximately 2.40 times greater than seller s_6's mean profit value of 0.5153. This is because after the first few hundred auctions, seller s_7 is able to learn to improve the quality of its goods to meet the buyers' demand, and therefore constantly makes successful sales to the buyers.

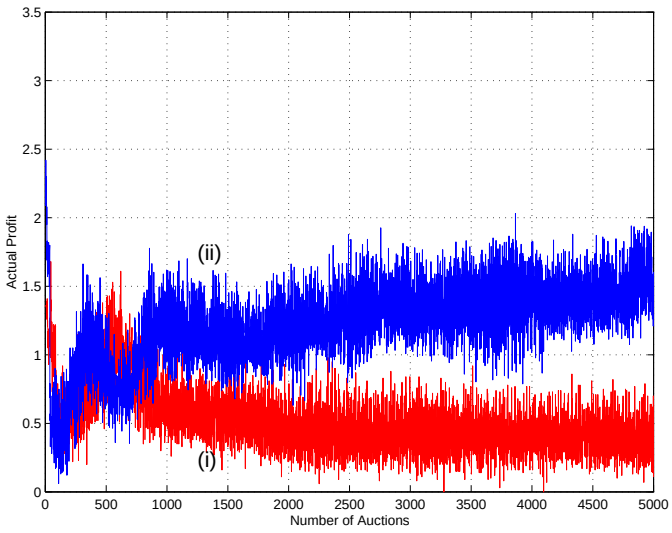

Figure 4.7: Comparison of actual profit values made by seller s_6, the most successful seller among those that use reinforcement learning but do not consider adjusting product quality (graph(i)), and by seller s_7, the seller that follows the proposed selling algorithm (graph (ii)).

Alternatively, figure 4.8(a) and (b) display the histograms of actual profit values made by seller s_6 and by seller s_7, respectively. We notice that seller s_6 makes very few sales in which the mean profit value is 1.5, and only about 650 sales in which the mean profit value is 1.0; while seller s_7 is able to make almost 2500 sales in which the mean profit value is 1.5, and over 2000 sales in which the mean profit value is 1.0. Seller s_6 makes almost 4000 sales with very low mean profit value of 0.5, while seller s_7 makes only about 250 sales with that low mean profit value. In other words, seller s_7 is able to make more sales with higher profits and fewer sales with lower profits, and therefore obtain a better level of satisfaction.

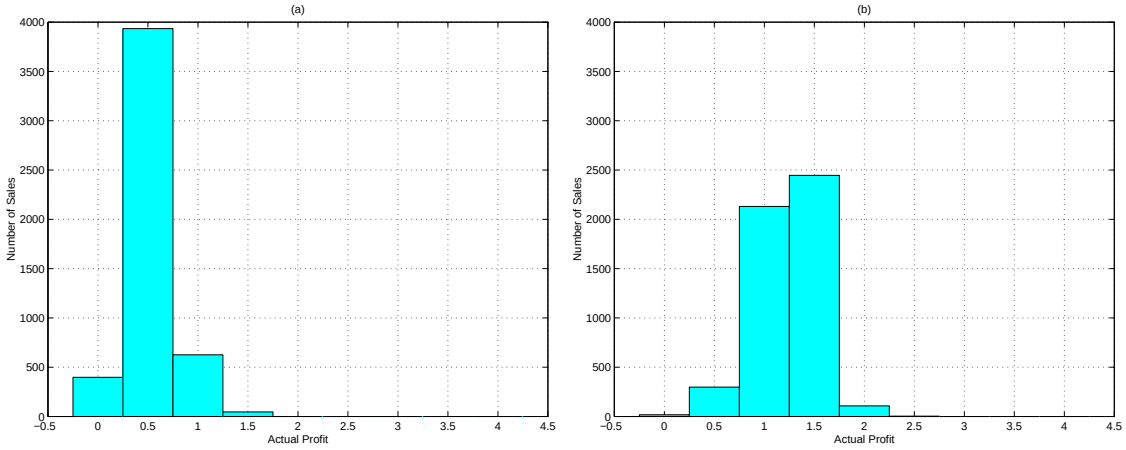

Figure 4.8: Histograms of actual profits made by seller s_6, the seller that uses reinforcement learning but does not consider adjusting product quality (a), and by seller s_7, the seller that follows the proposed selling algorithm (b).

4.1.2 Large Sized Marketplaces

This experiment aims to confirm that in a large sized marketplace, buyers that follow the proposed buying algorithm still obtain better satisfaction, compared to buyers that use reinforcement learning but do not model sellers' reputation; and sellers that follow the proposed selling algorithm still have more opportunities to win auctions, compared to sellers that use reinforcement learning but do not consider improving the quality of their goods.

We simulate a large marketplace populated with 160 sellers and 120 buyers. The seller population is divided into four groups:

- Group A consists of seller s_0, s_1, ..., and s_{39} that offer goods with quality chosen randomly from the interval $[32.0, 42.0]$.

- Group B consists of seller s_{40}, s_{41}, ..., and s_{79}. These are dishonest sellers who try to attract buyers with high quality goods ($q = 45$) and then cheat them with really low quality ones ($q = 1$).

- Group C consists of seller s_{80}, s_{81}, ..., and s_{119} that offer goods with fixed quality $q = 39.0$.

- Group D consists of seller s_{120}, s_{121}, ..., and s_{159} that also offer goods with an initial quality of 39.0. However, these sellers follow the proposed selling algorithm to improve the quality of their goods.

The buyer population is divided into two groups:

- Group I consists of buyer b_0, b_1, ..., and b_{59}. These buyers use reinforcement learning alone and do not model sellers' reputation.

- Group II consists of buyer b_{60}, b_{61}, ..., and b_{119}. These buyers follow the proposed buying algorithm.

We set the demanded product value $\vartheta^b(g) = 100$. Thus, even when a seller has to sell at cost, it must offer goods with quality of at least 40 in order to meet the buyers' requirement[3]. The decreasing factor for the learning rate α and the exploration probability ρ is 0.9997. This high decreasing factor gives the buyers opportunities to visit every seller (in this large seller population) with high learning rate values before the learning rate is further decreased. Other parameters are the same as in the previous experiment.

It should be obvious that a successful buyer would focus its business on group D of sellers and try to keep away from group A and B as much as possible.

[3]Because $v^b(p, q) = 3.5q - p = 3.5(40) - 40 = 100$.

The experimental results reported in this subsection are based on the average taken over the buyer population in which each buyer is exposed to 5000 auctions.

Buyers' Satisfaction

We compare the satisfaction level of a buyer following the proposed buying algorithm with that of a buyer not modelling sellers' reputation by looking at their numbers of purchases made to the four groups of sellers. Alternatively, we also examine the histograms and graphs of true product values obtained by these two buyers. We are also interested in seeing how better the buyer following the proposed algorithm is able to avoid interaction with the group of dishonest sellers, compared to the buyer not modelling sellers' reputation.

Table 4.4 shows the number of purchases made to four groups of sellers by the buyer not modelling sellers' reputation (labelled as b_I), and the buyer following the proposed algorithm (labelled as b_{II}).

	Group A	Group B	Group C	Group D
b_I	937.0	650.2	1196.0	2216.8
b_{II}	622.2	160.0	790.3	3427.5

Table 4.4: Number of purchases made to four groups of sellers by a buyer not modelling sellers' reputation (b_I), and by a buyer following the proposed algorithm (b_{II}).

As showed in the table, buyer b_{II} makes about 315 fewer purchases (or 33.6% fewer) from group A of sellers and 490 fewer purchases (or 75.4% fewer) from group B of sellers respectively, compared to the number of purchases made to these two

groups of sellers by buyer b_I. In addition, buyer b_{II} makes approximately 1210 more purchases (or 54.6% more) from group D of sellers, compared to the number of purchases made to that group by buyer b_I. In other words, buyer b_{II} focuses its business on the best group of sellers (group D) and stays away from the undesired ones (group A and B), and therefore obtains better satisfaction.

As an alternative view, Figure 4.9(a) and (b) present the histograms of product values obtained by a buyer not maintaining reputation ratings of sellers, and by a buyer following the proposed buying algorithm, respectively.

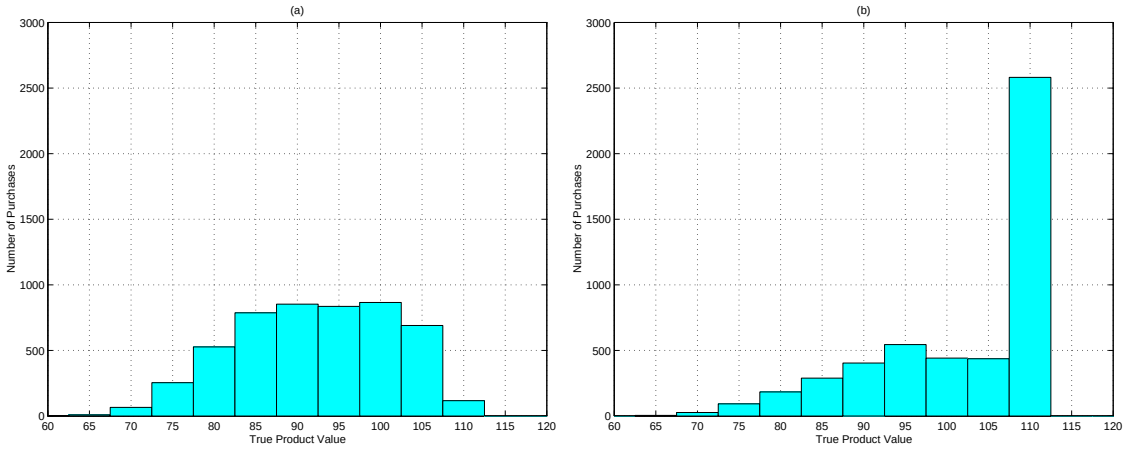

Figure 4.9: Histograms of true product values obtained by a buyer not modelling sellers' reputation (a), and by a buyer following the proposed buying algorithm (b).

The histograms clearly show that the buyer following the proposed algorithm receives fewer goods with low values (65 - 105) and more goods with high value (110), and is therefore better satisfied. In particular, the buyer following the proposed algorithm makes about 2400 more purchases with high mean product value of 110 (or about 16 times greater) than those made by the buyer not modelling sellers' reputation.

The fact that the buyer following the proposed algorithm obtains better satisfaction than the one not modelling sellers' reputation can also be seen by observing the graphs of product values over the number of auctions obtained by these two buyers as shown in Figure 4.10.

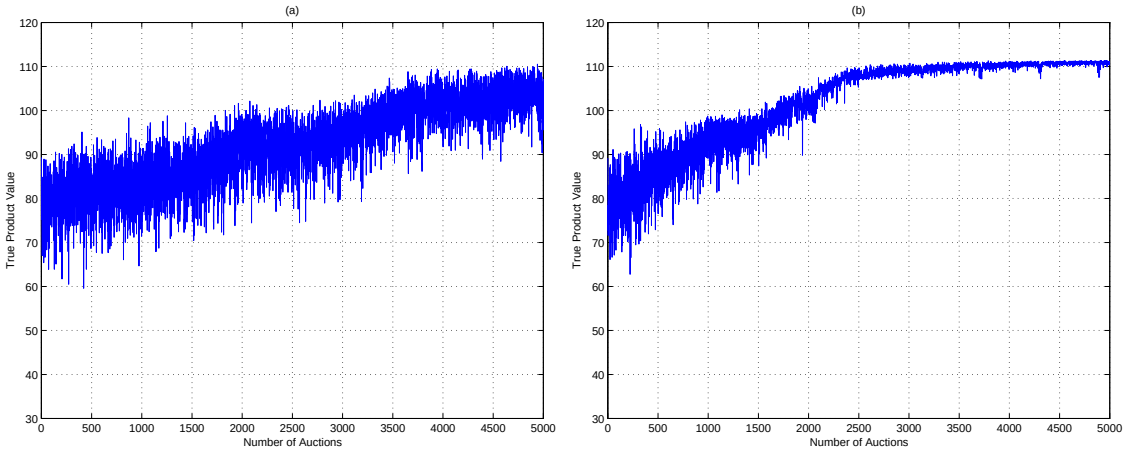

Figure 4.10: Graphs of true product values over number of auctions obtained by a buyer not modelling sellers' reputation (a), and by a buyer following the proposed buying algorithm (b).

Clearly, we can see that the graph of true product values obtained by the buyer following the proposed algorithm (b) is higher than that obtained by the buyer not modelling sellers' reputation (a). In fact, the mean product value obtained by the buyer following the proposed algorithm is 102.01, which is about 10.7% higher than the mean product value of 92.18 obtained by the buyer not modelling sellers' reputation.

In addition, we would like to see how better the buyer using the proposed algorithm is able to avoid interaction with the group of dishonest sellers (i.e., group B), compared to the buyer not modelling sellers' reputation. Figure 4.11(a) and (b)

show the graphs of profit made by the dishonest sellers from the buyer not modelling sellers' reputation, and from the buyer following the proposed buying algorithm, respectively. We can see that graph (a) is higher than graph (b), indicating that the dishonest sellers are able to make more profit from those buyers that do not make use of a reputation mechanism. Moreover, the profit in graph (b) is reduced to zero after about 2700 auctions, implying that from that point on the dishonest sellers are not able to make any profit from the buyer following the proposed algorithm, since they are considered as disreputable sellers and therefore no longer chosen by the buyer.

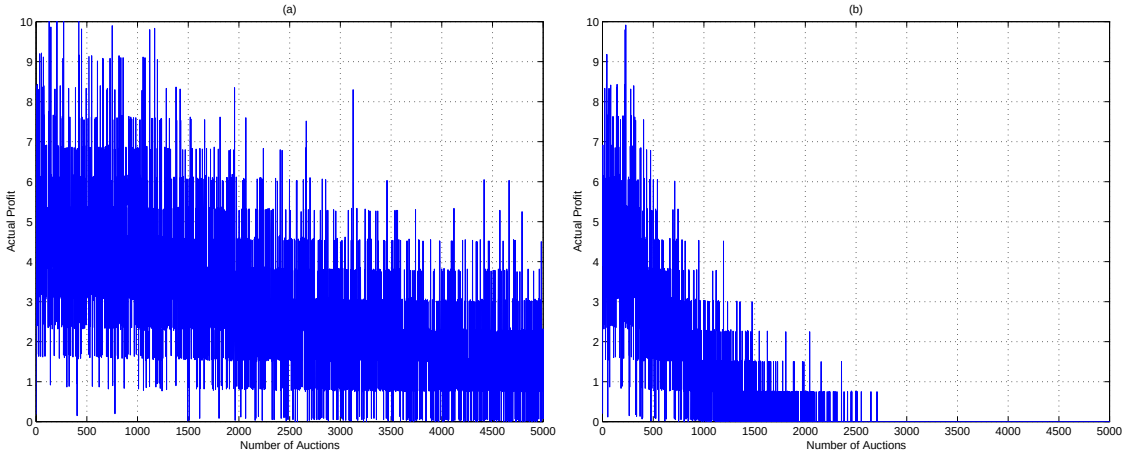

Figure 4.11: Graphs of profit values over number of auctions made by the dishonest sellers from a buyer not modelling sellers' reputation (a), and from a buyer following the proposed buying algorithm (b).

Sellers' Satisfaction

We compare the satisfaction level of the four groups of sellers by examining their sales numbers and graphs of profit values made to a buyer.

Table 4.5 shows the number of sales made by the four groups of sellers to a buyer.

Group A	Group B	Group C	Group D
779.6	405.1	993.2	2822.1

Table 4.5: Number of sales made by the four groups of sellers to a buyer.

Group D is able to make the most number of sales. In particular, the number of sales made by this group is approximately 3.6 times greater than that made by group A, 7 times greater than that made by group B, and 2.8 times greater than that made by group C, respectively.

Figure 4.12(a), (b), (c), and (d) show the graphs of profit values over the number of auctions made from a buyer by group A, B, C, and D of sellers, respectively.

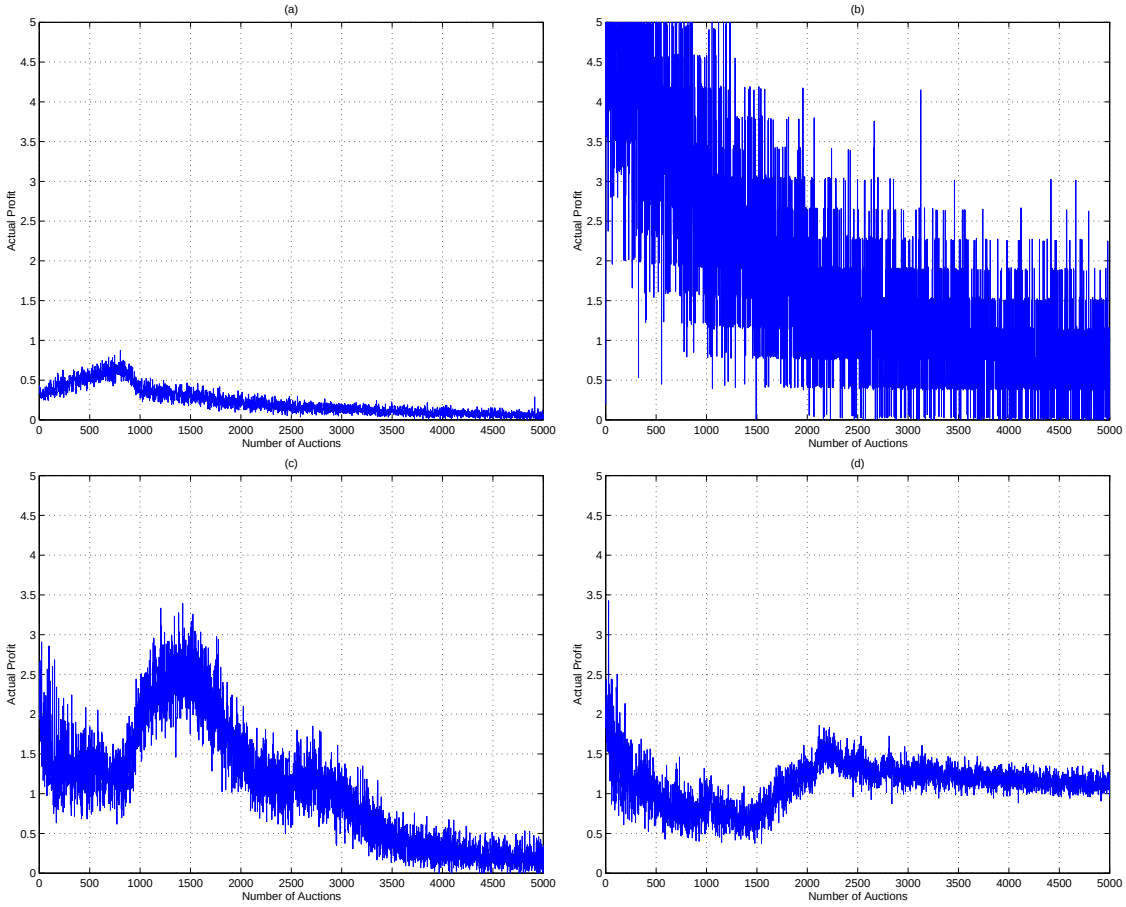

Figure 4.12: Graphs of actual profit values over number of auctions made from a buyer by group A (a), group B (b), group C (c), and group D of sellers (d).

The goods offered by sellers in group A usually do not meet the buyers' need since their quality is chosen randomly. As a result, this group of sellers receives low profit (graph (a)). The dishonest sellers in group B attract buyers with high quality goods, and then cheat them with really low quality ones, in order to make big profit. Consequently, their sales are on and off (mostly made to the group of buyers that do not model sellers' reputation as shown in Figure 4.11), resulting in greatly fluctuating profit (graph (b)). Group C of sellers offers goods with fixed

quality and is able to make relatively high profit in the first 1500 auctions. However, as the sellers in group D improve the quality of their goods, the sellers in group C start losing their sales in the long run. Graph (c) shows that their profit begins to go down after about 1500 auctions, and reaches the mean of about 0.25 after 3500 auctions. Although sellers in group D start with relatively low quality goods, they consider improving the quality of their goods according to the proposed selling algorithm. As a result, they make more and more sales and their profit increases substantially after 1500 auctions, reaching the mean of about 1.25 (graph (d)), which is five times greater than that of group C.

4.2 Macro Behaviours

On the macro level, we would like to study how a market populated with our buyers and sellers would behave as a whole. Since those sellers who repeatedly fail to sell their goods may decide to leave the market, we are particularly interested in knowing if such a market would reach an equilibrium state where the agent population remains stable. We would also like to know if this equilibrium state would be beneficial for the participating agents in terms of their satisfaction.

To achieve this goal, we simulate a fairly large marketplace populated with buyers and sellers following our proposed buying and selling algorithms, respectively. At each auction we record the seller population in the market, assuming that those sellers who are no longer able to make sales, and therefore profit, will decide to leave the market. We examine the graph of seller population varying against the number of auctions to determine if an equilibrium state has been reached. We also investigate if an obtained equilibrium would be beneficial for the participating agents in terms of their satisfaction, by examining the true product values obtained

and the prices paid by the buyers.

Our simulated marketplace is set up with 60 sellers and 80 buyers. The seller population is divided into two groups:

- Group A consists of seller s_0, s_1, ..., and s_{29} that offer goods with initial quality $q_{start} = 38.0$. Each of these sellers considers improving the quality of its goods up to a value chosen randomly in the interval $[q_{start}, q_1)$ where $q_1 = 42.0$.

- Group B consists of seller s_{30}, s_{31}, ..., and s_{59}. These sellers also offer goods with initial quality $q_{start} = 38.0$; however, they consider improving the quality of their goods up to value $q_2 = 44.0$.

All buyers in the market have the true product value function $v^b(g, p, q) = 3.5q - p$, where p and q respectively represent the price and quality of the good g purchased, and the demanded product value $\vartheta^b(g) = 105.0$. Thus, even when a seller sells at cost, it must offer goods with quality of at least 42.0 in order to meet the buyers' demand[4]. Because a seller will not sell under cost, it is obvious that the goods offered by sellers in group A do not meet the buyers' demand. Consequently, these sellers will not be able to make sales in the long run and will eventually decide to leave the market. Sellers in group B, however, satisfy the buyers' requirement and will remain in the market as they improve the quality of their goods up to the value of 44.0, greater than the minimum required value of 42.0.

Other parameters are chosen similar to previous experiments as follows:

- The quality q of a good is chosen to be equal to the cost for producing that

[4]Because $3.5(42.0) - 42.0 = 105.0$.

good. This supports the common assumption that it costs more to produce high quality goods.

- The reputation threshold $\Theta = 0.5$ and the disreputation threshold $\theta = -0.9$.

- If $v^b - \vartheta^b \geq 0$, we define the cooperation factor μ as in equation (3.12), which is repeated below for convenient reading:

$$\mu = \begin{cases} \dfrac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} & \text{if} \quad \frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} > \mu_{min}, \\ \mu_{min} & \text{otherwise,} \end{cases} \tag{4.3}$$

  where $\mu_{min} = 0.005$, $v^b_{max} = 3.5q_{max} - p_{min}$, $v^b_{min} = 3.5q_{min} - p_{max}$, $q_{max} = p_{max} = 49.0$, and $q_{min} = p_{min} = 1.0$. We vary μ as an increasing function of v^b to implement the common opinion that the reputation rating of a seller that offers goods with higher product value should be better increased. We prevent μ from becoming zero when $v^b = \vartheta^b$ by using the value of μ_{min}.

- If $v^b - \vartheta^b < 0$, we define the noncooperation factor ν as in equation (3.13), and also repeat it below for convenient following:

$$\nu = \lambda(\frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}}), \tag{4.4}$$

  where we set the penalty factor $\lambda = 3$ as in (3.41). We also vary ν as an increasing function of v^b to support the idea that the lower product value a seller offers, the more its reputation rating should be decreased. The use of factor $\lambda > 1$ indicates that a buyer will penalize a non-cooperative seller λ times greater than it will award a cooperative seller. This implements the traditional assumption that reputation should be difficult to build up, but easy to tear down.

- The exploration probability ρ and the learning rate α are both set to 1 initially, and decreased over time (by factor 0.9997) down to $\rho_{min} = \alpha_{min} = 0.1$.

- The number of consecutive unsuccessful auctions (after which a seller following the proposed algorithm may consider improving the quality of its goods) $m = 10$, and the number of consecutive successful auctions (after which a seller following the proposed algorithm may consider reducing the quality of its goods) $n = 10$.

- The quality increasing factor $Inc = 0.05$, and the quality decreasing factor $Dec = 0.05$.

The following reported results are based on the average of 100 runs where, in each run, we let each buyer participate in 10000 auctions. Examining a large number of auction provides a long term overall view on the market behaviours.

Figure 4.13 shows how the seller population varies against the number of auctions. At the beginning the seller population consists of 60 sellers. Because the sellers in group A offer goods that do not meet the buyers' demand, they lose more and more sales. As a result, these sellers are no longer able to make profit in the long run and have to leave the market. This situation is reflected by a decrease in the graph of seller population from after 500 auctions to approximately 3000 auctions. The sellers in group B, however, are able to satisfy the buyers' need and therefore remain in the market. This is showed in the graph by an equilibrium value (of 30) obtained after 3000 auctions.

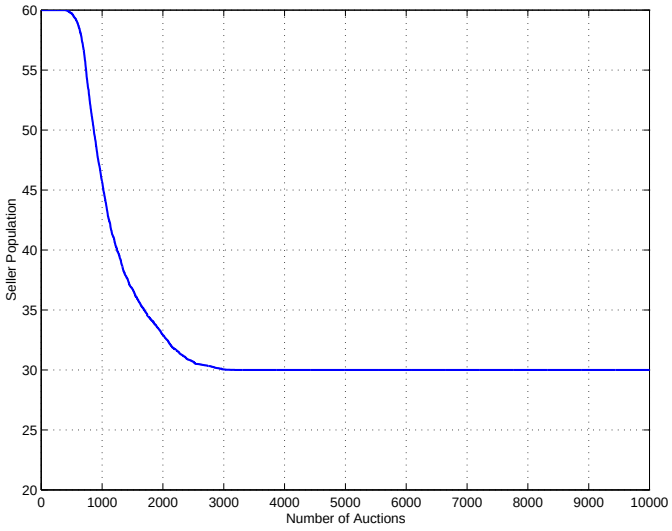

Figure 4.13: Seller population reaches equilibrium state.

Such an equilibrium is beneficial for the buyers because after the establishment of the equilibrium, there are no sellers in the market offering goods under the buyers' demanded value (of 105). It should also be beneficial for the sellers since there are now fewer competitors in the market, resulting in more opportunities for the sellers to make their sales. Figure 4.14 displays the true product values obtained by a buyer over the tested number of auctions. We can notice clearly that the buyer receives goods with unsatisfactory values during the first 1000 auctions, before the equilibrium is reached. After the equilibrium, the product values received are substantially improved. They are, in fact, asymptotically close to the highest possible value of 110 that a seller in this market is able to offer[5].

[5]Because the highest quality that a seller in our market offers is $q_2 = 44.0$, and because we assume that cost equals quality, the highest possible value a seller can offer in case it has to sell at cost is $v^b = 3.5(44.0) - 44.0 = 110.0$

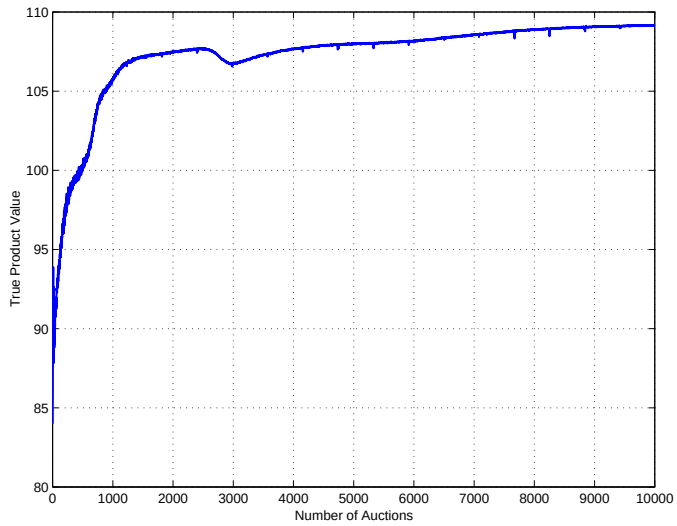

Figure 4.14: True product values obtained by a buyer.

Figure 4.15 displays the price paid by a buyer at each auction during the experimented number of auctions. Although the sellers adjusting quality and price prevents the price from reaching a single equilibrium value, the price tends to decrease towards value 44.0, after the seller population has reached an equilibrium state. This is because the remaining sellers in the market compete with one another by increasing product quality (up to $q_2 = 44.0$) and lowering the product price (down to the cost $c = q_2 = 44.0$) in order to attract the buyers.

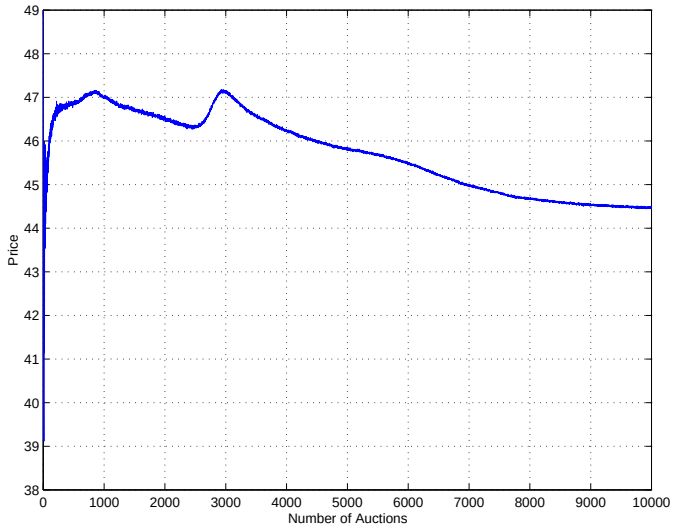

Figure 4.15: Prices paid by a buyer.

4.3 Lessons Learned

Several lessons can be discerned from our experimentation:

- Agents equipped with learning methods such as reinforcement learning to guide their behaviours perform much better than agents not making use of any learning methods.

- In both modest and large sized marketplaces where there are sellers altering the quality of their goods, buyers following our proposed buying algorithm obtain greater satisfaction than buyers not modelling sellers' reputation. Modelling the reputation of sellers allow the buyers to quickly identify reputable sellers who offer goods with satisfactory values, as well as disreputable sellers who offer low, unsatisfactory value goods. Buyers' satisfaction can therefore be obtained by focussing on doing business with the reputable sellers and

avoiding the disreputable ones. Interestingly enough, our experiments show that dishonest sellers cannot make profit from buyers following our proposed algorithm in the long run.

- In both modest and large sized marketplaces where buyers make use of some learning strategy, sellers following the proposed selling algorithm fare better than sellers who do not consider adjusting the quality of their goods. Adjusting product quality in addition to adjusting product price allows the sellers to provide value-added services that are tailored to meet the particular needs of their customers, hence allowing them to make more sales and accordingly obtaining better satisfaction.

- An equilibrium of agent population can be reached in a market populated with buyers and sellers following the proposed algorithms. The establishment of this equilibrium only allows those sellers whose goods satisfy the buyers' needs to remain in the market. Those sellers whose goods do not meet the buyers' needs are not able to make sales in the long run, and therefore are forced to leave the market.

- Such an equilibrium, once obtained, is beneficial for the market as a whole. On the sellers' side, they will have fewer competitors and therefore more chances to get buyers' attention. On the buyers' side, there are no sellers in the market that offer goods under their demanded value. In addition, competition among the remaining sellers tends to raise the value of goods towards the highest possible one, and to lower the price down to the production cost, which is clearly desirable for the buyers.

4.4 Chapter Summary

In this chapter, we experimentally evaluate our model by simulating several electronic marketplaces populated with buying and selling agents.

We first investigate the micro behaviours of buying and selling agents that participate in electronic marketplaces. Our experiments show that in both modest and large sized marketplaces, buyers and sellers following the proposed algorithms will fare better than buyers and sellers who only use reinforcement learning, but the buyers do not model sellers' reputation and the sellers do not consider adjusting the quality of their goods.

We then study the macro behaviours of markets populated with our buyers and sellers. Our experimental results confirm that such markets would reach an equilibrium state where the agent population remains stable (as some sellers who repeatedly fail to sell their goods may decide to leave the market), and the establishment of this equilibrium is beneficial for the agent society as a whole.

Chapter 5

Discussion

In Chapter 3 we described our agent market model and presented the proposed algorithms for buying and selling agents, based on reputation modelling and reinforcement learning. We experimentally evaluated our model in Chapter 4 by simulating different electronic marketplaces populated with trading agents. In this chapter we offer a detailed discussion on the value of our model. In particular, Section 5.1 contrasts our model with other related e-commerce agent models, and experimentally compares the performance of our agents with those proposed in the most related research. Section 5.2 discusses the merits of our model, including its potential advantages and the value of certain design decisions that we have made within the model. Section 5.3 comments on the reputation mechanism used in the model. Finally, Section 5.4 ends the chapter with a summary of the discussed issues.

5.1 Compare and Contrast

In this section we first contrast our model with other related e-commerce agent models. We then experimentally compare the performance of our trading agents with those proposed in the most related work [77, 78].

5.1.1 Contrast with Other Models

We presented in Chapter 2 several e-commerce agent models related to ours. In this subsection we briefly analyze the operations of these models in contrast to ours.

BargainFinder [3] and Jango [14, 29] are shopping agents that, given a specific good by a customer, will simultaneously query multiple online sellers for the price and related information of the good. The collected information is then displayed to the customer who will compare the sellers to select a suitable one. Although the information provided by BargainFinder and Jango are useful for sellers' comparison, these shopping agents are not autonomous: They leave the task of analyzing the collected information and choosing the appropriate seller completely to the customer. In addition, the algorithms underlying these shopping agents' operations do not capture important product information such as product quality or value added services, which are also necessary for comparing sellers. More importantly, BargainFinder and Jango do not make use of any AI learning techniques in order to help improve their performance over time.

Agents in the Kasbah marketplace [9] are more advantageous than BargainFinder and Jango in that they are autonomous in seeking out potential traders, therefore freeing customers from this tedious task. However, the weak point of Kasbah agents is that they are unable to learn. Hence, these agents would not

be smart enough to deal with environmental changes and especially with dishonest agents.

In contrast with the above agents, our buying (and selling) agents are designed to be both autonomous and possess learning capabilities. They pro-actively search and select appropriate trading partners in their operations. They also learn to optimize their expected product values (and expected profits) using reinforcement learning. Our choice of reinforcement learning as a learning technique for buying and selling agents is justified in Section 5.2.2.

The economy of shopbots and pricebots explored in [21, 22, 23] consists of B buyers, who are interested in a single homogeneous good g offered by S sellers. This work is concerned with the dynamics of interaction among different pricing algorithms. Its ultimate aim is to identify those pricing algorithms that are most likely to be profitable, from both an individual and a collective standpoint. The economy of shopbots and pricebots seems similar to our agent marketplace at the first glance; however, there are significant differences between the two models: The authors of [21, 22, 23] assume that the number of buyers in their market is infinitely greater than the number of sellers ($B >> S$). This assumption allows them to reduce their attention to only sellers' algorithms, because the small number of sellers implies that the behaviours of individual sellers will significantly influence the marketplace. Consequently, their buyers only follow one of the two simple strategies, namely *Bargain Hunter* and *Any Seller*. In Bargain Hunter, a buyer will choose a seller with the lowest price and purchase the good from that seller if the price is less than the buyer's evaluation of the good. In Any Seller, a buyer will select a seller at random and buy the good if the price is less than the buyer's evaluation of the good. In addition, the authors of [21, 22, 23] assume that the production cost of all sellers in their market is identical and unchanged by the

sellers.

In contrast, we think that the above assumptions are not always realistic and may be inflexible. A general marketplace should not have any restrictions either on the relationship of the buyer and the seller populations or on the buyer strategies. Also, the cost of production should not be considered identical and unchanged among sellers, not only because multiple sellers may produce the same good with different qualities using different production costs, but also because a seller may decide to alter the quality of its goods by probably changing its production cost. As a result, we design a market model that allows for any combinations of buyer and seller populations. We assume that the production cost of multiple sellers may not be the same and that a seller may alter the quality of its goods by changing its production cost. Finally, buyers of our marketplace are free to follow any strategy. In particular, our proposed buying strategy, using the combination of reinforcement learning and reputation modelling, goes beyond the consideration of price alone as in the Bargain Hunter strategy, or the random selection of sellers as in the Any Seller strategy.

Perhaps, the most related to our research is the work of Vidal and Durfee [77, 78] in which they develop strategies for trading agents using a recursive modelling approach. Their agents are divided into different classes depending on the agents' capabilities of modelling other agents. For instance, agents with 0-level models base their actions on the inputs and rewards they receive. Agents with 1-level models are those agents that model other agents as 0-level agents. Agents with 2-level models are those that model others as 1-level agents. In theory, agents with higher level models should fare better and could be recursively defined in the same manner. However, as pointed out in [77, 78], the agents with deeper recursive models of others suffer from the computational costs associated with maintaining

these deep models. In fact, due to the infeasible complexity in implementing agents with deeper models of others, the experimentation reported in [77, 78] is limited to only 1-level buyers and 2-level sellers. Moreover, the marketplace considered in [77, 78] does not allow for the sellers to alter the quality of their goods, nor does it address how to cope with dishonest sellers.

In contrast, we model a marketplace where the quality of a good offered by different sellers may not be the same, sellers may alter the quality of their goods, and there is a possibility of having dishonest sellers in the market. Furthermore, we would like to avoid heavy computational costs (such as those incurred by the agents with recursive models) for our trading agents and therefore take a different approach. The algorithm we propose for buying agents makes use of a combination of reinforcement learning and reputation modelling techniques. Modelling sellers' reputation plays the role of a pre-screening process, which partitions the set of sellers into three disjoint subsets, namely the reputable sellers, the disreputable sellers (including the dishonest sellers), and the neither reputable nor disreputable ones. Reinforcement learning is then applied to the set of reputable sellers (instead of all sellers) in exploitation steps, and to the non-disreputable sellers in exploration steps. This process helps buying agents to enhance their opportunity to purchase high value goods from the reputable sellers, and reduce the risk of purchasing low value goods from the disreputable sellers. In other words, reinforcement learning and reputation modelling work together as two layers of learning to improve the performance of buying agents. The algorithm we propose for selling agents enables them to learn to maximize their expected profits by not only adjusting product prices using reinforcement learning, but also by adjusting product quality in order to meet the buyers' specific needs. Since quality and price are the two most important factors based on which buying agents determine the value of the goods

they purchase, the proposed selling algorithm obviously gives more opportunities for selling agents to make successful sales.

5.1.2 Experimental Comparison

It is of interest to compare the performance of our agents with those proposed in [77, 78], given the above-said different characteristics of the two approaches. Thus, we first discuss the types of agents we choose for comparison and describe how these agents work. Then, we report the experimental comparative results showing that our proposed agents are able to achieve better performance, in terms of satisfaction and computational time.

Selecting Agents for Comparison

We experimentally compare our buyers and sellers with 1-level buyers and 0-level sellers proposed in [77, 78], respectively. We choose these specific agents for comparison because of the following reasons:

- As explained in [77, 78], since a buyer receives bids from the sellers, there is no need for the buyer to try to out-guess or predict what the sellers will bid. The buyer is not concerned with what other buyers are doing either, because it is assumed that there will be enough supply in the market[1]. Thus, buyers do not need to keep models of others deeper than level 1. In other

[1]We believe that this assumption is reasonable for a common marketplace. Marketplaces where buyers compete with one another for goods are usually those of traditional auctions, in which buyers are bidders and sellers are auctioneers. These auctions are discussed in Section 6.3.4 of Future Work.

words, 1-level buyers are the buyers with deepest models of others. We were therefore interested in challenging our buyers with 1-level buyers.

- We would like to compare our sellers with 0-level sellers because both our sellers and 0-level sellers learn from the observations they make about the environment and from any environmental rewards they receive. In addition, it is not relevant to compare our sellers with sellers of deeper levels (i.e., 1 or 2-level sellers), because these levels of sellers make use of two assumptions which, we think, are unrealistic and therefore do not implement in our market mechanism. These two assumptions are

  (i) The bid submitted by a seller to a buyer is known by other sellers in the market. This assumption is unrealistic because the bid submitted by a seller to a buyer should be treated as private information between that seller and buyer; and therefore, should not be made known to other sellers in the marketplace. Moreover, a seller would not have any incentive or interest to broadcast the bid it is submitting to a buyer to all other sellers in the market.

  (ii) The price accepted by a buyer at each auction is known by all sellers in the market. This assumption is also unrealistic because the buyer would not want everybody know the price at which it purchases the good from a particular seller, and neither would the seller involved in the transaction; otherwise, its behaviours would be modelled and exploited by other sellers in the market.

How 1-Level Buyers and 0-Level Sellers Work

Let us have a brief look at how 1-level buyers and 0-level sellers work, as described in [77, 78].

- 1-level buyers model sellers in the marketplace by keeping a history of the qualities of the goods they purchased from each seller. In particular, a 1-level buyer b remembers the last N qualities offered by a seller s for the good g that it purchases from s. It then defines a probability density function $q_s^g(x)$ over the quality x offered by seller s for good g. Function $q_s^g(x)$ returns the probability that seller s will offer an instance of good g that has quality x. Buyer b then uses the expected value of this probability density function to calculate which seller will offer good g with highest expected product value:

$$s^* = \arg\max_{s \in S} E(V_b^g(p_s^g, q_s^g(x))) \tag{5.1}$$

$$= \arg\max_{s \in S} \frac{1}{|Q|} \sum_{x \in Q} q_s^g(x) V_b^g(p_s^g, x) \tag{5.2}$$

  where Q is a finite set of values representing product qualities.

- A 0-level seller s, when requested by some buyer b for the price of some good g, will choose a price p_s^* greater than or equal to its cost c_s^g to produce g such that its expected profit is maximized:

$$p_s^* = \arg\max_{p \in P} h_s^g(p) \tag{5.3}$$

  where $h_s^g(p)$ returns the profit seller s expects to get if it offers good g at price p. Depending on the success of the transaction, $h_s^g(p)$ is learned as follows:

$$h_s^g(p) \leftarrow (1 - \alpha) h_s^g(p) + \alpha Profit_s^g(p) \tag{5.4}$$

where α is the learning rate ($0 \leq \alpha \leq 1$) and $Profit_s^g(p)$ is the actual profit:

$$Profit_s^g(p) = \begin{cases} p - c_s^g & \text{if } s \text{ is able to sell } g, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

Buyers' Comparison

We experimentally compare the performance of our proposed buyers with 1-level buyers. We consider two comparison criteria, namely satisfaction and computational costs. In particular, we simulate a marketplace populated with 32 sellers and 40 buyers, using Java 2. The seller population is equally divided into four groups (each having 8 sellers):

- Group A offers goods with quality chosen randomly from interval [32, 42].
- Group B consists of dishonest sellers who attract buyers with high quality goods ($q = 45$) and then cheat them with really low quality ones ($q = 1$).
- Sellers in group C offer goods with fixed quality $q = 40$. These sellers do not consider adjusting the quality of their goods.
- Sellers in group D offer goods with relatively lower starting quality $q = 38$, compared to sellers in group C. However, these sellers will consider improving product quality up to value 45 in order to meet the buyers' needs, according to our proposed selling algorithm.

The buyer population is equally divided into 2 groups:

- Group I consists of the 1-level buyers.
- Group II consists of our proposed buyers.

Other parameters are set as follows:

- The number of qualities N offered by a seller s for some good g that a 1-level buyer remembers is 50.
- The quality q of a good is chosen to be equal to the cost for producing that good. This supports the common assumption that it costs more to produce high quality goods.
- The true product value function $v^b(g, p, q) = 3q - p$, where p and q represent the price and quality of the good g purchased, respectively.
- The reputation threshold $\Theta = 0.5$ and the disreputation threshold $\theta = -0.9$.
- The demanded product value $\vartheta^b(g) = 80$. Thus, even when a seller sells at cost, it must offer goods with quality of at least 40 in order to meet the buyers' requirement[2].
- The cooperation factor μ is defined as in equation (3.12). That is, if $v^b - \vartheta^b \geq 0$,

$$\mu = \begin{cases} \dfrac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} & \text{if} \quad \frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}} > \mu_{min}, \\ \mu_{min} & \text{otherwise,} \end{cases} \tag{5.6}$$

  where $\mu_{min} = 0.005$, $v^b_{max} = 3q_{max} - p_{min}$, $v^b_{min} = 3q_{min} - p_{max}$, $q_{max} = p_{max} = 49.0$, and $q_{min} = p_{min} = 1.0$. As mentioned in Section 3.2.1, we vary μ as an increasing function of v^b to reflect the idea that the reputation rating of a seller that offers goods with higher product value should be better increased. Also, we prevent μ from becoming zero when $v^b = \vartheta^b$ by using value μ_{min}.

[2]Because $3(40) - 40 = 80$.

- The non-cooperation ν is defined as in equation (3.13). Hence, if $v^b - \vartheta^b < 0$,

$$\nu = \lambda(\frac{v^b - \vartheta^b}{v^b_{max} - v^b_{min}}) \tag{5.7}$$

  where we choose $\lambda = 3 \approx \frac{1}{1-\frac{\vartheta - v_{min}}{v^b_{max} - v^b_{min}}}$ according to (3.41). As explained in Section 3.2.1, we vary ν as an increasing function of v^b to reflect the idea that the lower product value a seller offers, the more its reputation rating should be decreased. The use of factor $\lambda > 1$ indicates that a buyer will penalize a non-cooperative seller λ times greater than it will award a cooperative seller. This implements the traditional assumption that reputation should be difficult to build up, but easy to tear down.

- The exploration probability ρ and the learning rate α are both set to 1 initially, and decreased over time (by factor 0.998) down to $\rho_{min} = 0.1$ and $\alpha_{min} = 0.1$.

- The number of consecutive unsuccessful auctions (after which a seller following our proposed algorithm may consider improving the quality of its goods) $m = 10$, and the number of consecutive successful auctions (after which a seller following our proposed algorithm may consider reducing the quality of its goods) $n = 10$.

- The quality increasing factor $Inc = 0.05$, and the quality decreasing factor $Dec = 0.05$.

The results we report here are based on the average of 100 runs each of which has 5000 auctions.

Because the higher product value a buyer receives, the better satisfied it is, we record and present in Figure 5.1 the true product values obtained by a 1-level buyer (graph (i)) and by a buyer following our proposed algorithm (graph (ii)).

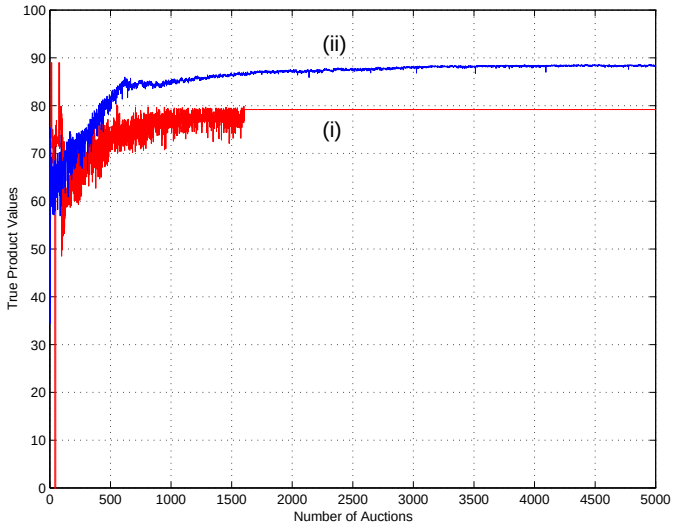

Figure 5.1: Comparison of true product values obtained by a 1-level buyer (graph (i)), and by a buyer following our proposed algorithm (graph (ii)).

As shown in the figure, the buyer following the proposed algorithm receives goods with higher true product values and is therefore more greatly satisfied. In fact, the product values this buyer obtains are reaching the highest possible value (90) that could be offered in the marketplace[3]. The highest product value obtained by the 1-level buyer is about 80 only, indicating that it selects sellers in group C as its favourite sellers[4]. Clearly, the 1-level buyer is not able to discover that sellers in group D are actually the best sellers to purchase from. The reason is that although the 1-level buyer may try these sellers with their improved quality products, the history of low initial quality products offered by these sellers earlier keeps the buyer

[3]Since the highest quality offered (by sellers in group D) in our marketplace is 45 and since we assume cost equals quality, the highest possible product value offered in our market (by sellers in group D if they sell at cost) would be 3(45) - 45 = 90.

[4]These sellers offer goods with fixed quality 40. So, the highest product value they could offer if they sell at cost is 3(40) - 40 = 80.

from selecting them as sellers with maximum expected value, according to the buyer's probability density function model shown in equations (5.1) and (5.2).

We are also interested in investigating the performance of the two buyers in terms of computational time. The run time needed for a buyer to complete an auction is composed of communication time and computational time. The communication time accounts for the time needed for communication between the buyer and sellers (e.g., the buyer broadcasting its request to sellers, the sellers responding with their bids, etc.). The computational time accounts for the time needed by the buyer to compute the seller that it will purchase the good from, according to its buying algorithm. Clearly, the communication time depends on the specific network underlying the marketplace and is therefore not relevant for comparison. The computational time, however, depends on the complexity of the buying algorithm and can be compared between agents using different algorithms. Obviously, the shorter the computational time the better the algorithm. This is especially important in application domains where the buyer is required to calculate a suitable seller within a constrained time frame. For instance, if the buyer serves some user as a personal assistant, then it must respond to the user within an allowable time period. Figure 5.2 shows the computational time over the number of auctions taken by a 1-level buyer (graph (i)) and by a buyer using our proposed algorithm (graph (ii)).

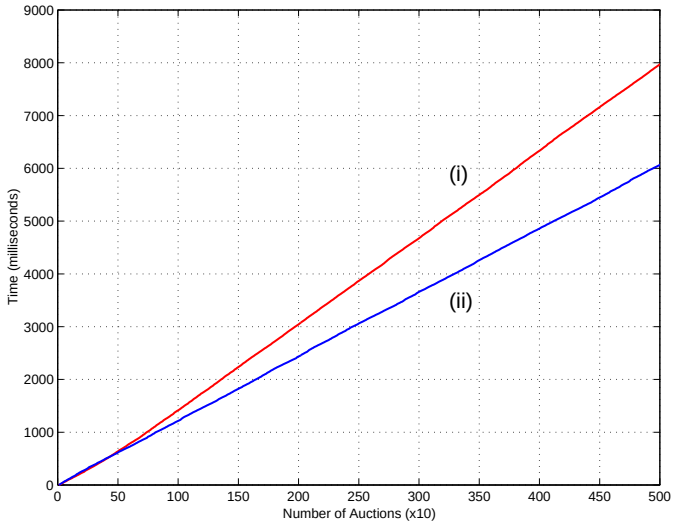

Figure 5.2: Comparison of computational time over the number of auctions taken by a 1-level buyer (graph (i)), and by a buyer using our proposed algorithm (graph (ii)).

The figure indicates that the buyer following our proposed algorithm outperforms the 1-level buyer. This is argued even more convincingly by looking at the respective algorithms governing the behaviours of these two buyers. In order to calculate the seller with highest expected value, the 1-level buyer has to examine every seller in the market (equation (5.1)). Moreover, for each seller s, the 1-level buyer also needs to calculate the product of the expected value and the probability of having that value at each quality q (equation (5.2)). Thus, the order of growth of the algorithm underlying the 1-level buyer is $O(|S||Q|)$, where $|S|$ and $|Q|$ are the cardinalities (sizes) of the set of sellers and the set of quality values, respectively. The order of growth of our proposed algorithm is $O(|S|)$, since a buyer b only needs to examine the set of sellers to compute a suitable seller. In the long run when every seller may be placed into either the set of reputable sellers or the

set of disreputable sellers, this order will be reduced to $O(|S_r^b|)$, where S_r^b is the set of reputable sellers to buyer b and $|S_r^b|$ should be a lot smaller than $|S|$. Obviously, $O(|S_r^b|)$ beats $O(|S||Q|)$, especially when $|S|$ and $|Q|$ are sufficiently large.

Sellers' Comparison

We also experimentally compare the performance of our proposed sellers with 0-level sellers, in terms of satisfaction level and computational costs. Towards this objective, we simulate a marketplace populated with 20 sellers and 40 buyers. We let half of the sellers be the 0-level sellers, who offer goods with fixed quality of 40. The other half are our proposed sellers, who provide goods with lower initial quality of 38 but consider adjusting product quality to meet the buyers' needs, according to the proposed algorithm. All buyers follow a simplified learning version of our proposed buying algorithm, that is, they only use reinforcement learning and do not model sellers' reputation. Other parameters such as $v^b(g, p, q)$, ρ, α, m, n, Inc, and Dec are chosen as in the previous experiment. The following reported results are based on the average taken over the buyer population.

Since the higher profit a seller makes the more greatly satisfied it is, we show in Figure 5.3 the profits made over the number of auctions from a buyer by the 0-level sellers (graph (i)), and by the sellers following our proposed algorithm (graph (ii)). We notice from the figure that, at the beginning the 0-level sellers are often chosen by the buyer because they offer goods with higher quality. However, as sellers of the other group improve the quality of their goods, the 0-level sellers lose more and more sales in the long run. This is indicated by a sharp decline in the profit graph, reaching the mean of approximately 0.5 after about 1000 auctions. In contrast, as the sellers following our proposed algorithm improve their product quality, they are selected more and more often by the buyer, resulting in their improved profit.

In fact, they outperform the 0-level sellers after 1000 auctions with their profit reaching the mean of about 2.25, which is 4.5 times greater than that of the 0-level sellers.

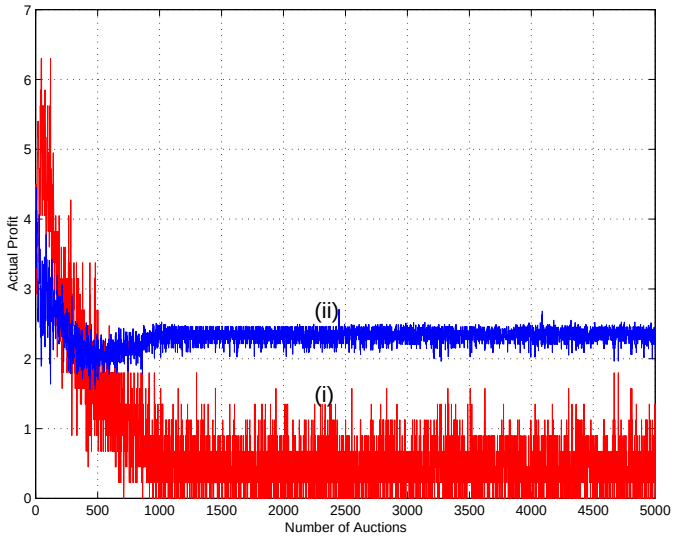

Figure 5.3: Actual profits made from a buyer by the group of 0-level sellers (graph (i)), and by the group of sellers following our proposed algorithm (graph (ii)).

Although the sellers following our proposed algorithm achieve better satisfaction than the 0-level sellers, they do not incur more computational time. This is because both algorithms underlying these two seller types have the same order of growth $O(|P|)$ where $|P|$ is the cardinality of the set of prices, as the sellers of both types search this set for the price that maximizes their expected profits. Indeed, Figure 5.4 shows that the difference in computational time spent by the two types of sellers is negligible.

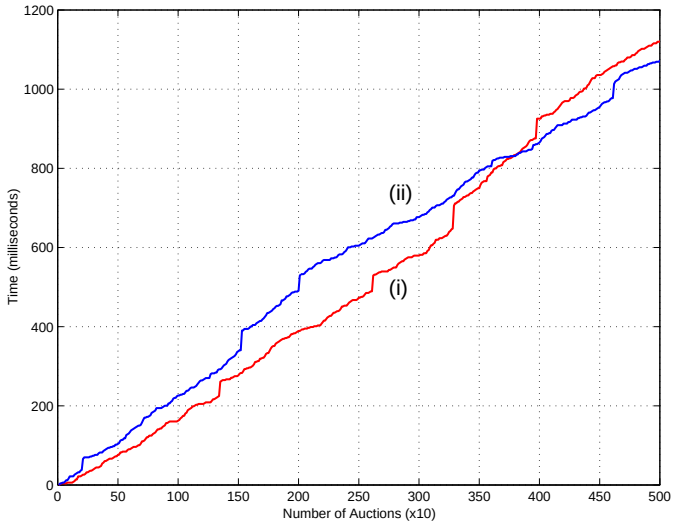

Figure 5.4: Computational times spent by a 0-level seller (graph (i)), and by a seller following the proposed algorithm (graph (ii)).

5.2 Merits of Model

In this section we discuss the potential advantages of our model. We also discuss the value of certain design decisions that we have made in the development of the model.

5.2.1 Potential Advantages

Work in the area of software agents has been focusing on how agents should interact or cooperate to provide valuable information services to one another [33, 35, 36]. However, answering the question of why agents should interact with one another at all is also of equal importance [34]. Therefore, modelling an agent environment as a marketplace, where agents are motivated by economic incentives to provide

goods and services to each other, has practical value especially for e-commerce applications.

We believe that the agent market model that we proposed can be used in designing electronic marketplaces and their trading agents. There are several reasons that support our belief.

- First, our model is built on a set of general and flexible assumptions that account for the open, dynamic, uncertain and untrusted natures of a real electronic marketplace. These assumptions include:

  (i) Trading agents can freely enter or leave the market (open market).

  (ii) The quality of a good offered by multiple sellers may not be the same and a seller may alter the quality (and certainly, the price) of its goods (dynamic market).

  (iii) A buyer would not know the quality of the good it purchases until after it receives the good from the seller (uncertain market).

  (iv) There may be dishonest sellers (untrusted market).

- Secondly, our market mechanism with its three elementary phases for buying and selling goods is simple and hence implementable, given the networking capabilities of modern programming languages.

- Thirdly, our proposed learning algorithms for trading agents are feasible and effective (in terms of the agents' satisfaction level), as experimentally demonstrated in Chapter 4.

It is possible that there are dishonest sellers in a marketplace. Buyers in our approach use a reputation mechanism in addition to reinforcement learning as a means

to protect themselves from dishonest sellers. They each dynamically maintain sets of reputable and disreputable sellers, and learn to maximize their expected value of goods by selecting appropriate sellers among the reputable sellers while avoiding the disreputable ones. This strategy should increase a buyer's chance of purchasing high value goods and reduce its risk of receiving low value ones, and therefore bring better satisfaction to the buyer. In addition, the fact that the marketplace is open will allow new reputable sellers to enter the market. Also, since sellers may be learning to improve their profits, some non-reputable sellers may have reasonably adjusted the prices and greatly improved the quality of their goods, and thus should be reconsidered as reputable sellers. Our proposed buying algorithm accounts for these possibilities by letting a buyer b explore the marketplace with probability ρ to discover new reputable sellers. The experimental results reported in Chapter 4 confirm these advantages of the proposed buying algorithm.

A proposed buyer, in its exploitation time, will choose a seller from the set of reputable sellers. Since this reputable set is usually quite smaller (in terms of cardinality) than the set of all sellers in the market, the proposed buying algorithm should reduce computational time, and accordingly results in improved time-performance for the buyer (compared to the case where the buyer has to consider all possible sellers in the market). This is especially important in those application domains where the buyer is required to calculate a suitable seller within a constrained time frame.

There are two important reasons why a seller may not be able to win an auction (i.e., to sell a good to a buyer): *(i)* It may set the price too high, and *(ii)* the quality of its good may not meet the buyer's demanded level. Our proposed selling algorithm considers both of these factors by allowing a seller to not only adjust the price (equation (3.15)), but also optionally adjust the quality to its good (equation

(3.17)). This strategy should provide the seller with more opportunities to win an auction and consequently bring greater satisfaction to the seller. The value of our proposed selling algorithm is demonstrated via experimentation in Chapter 4.

The underlying mechanism that facilitates our agents in buying and selling goods is actually a form of the contract-net protocol [13, 63], where buyers announce their requests for goods to all sellers via multi-cast or possibly broadcast. This mechanism works well in small and moderate-sized environments. However, as the problem size (i.e., the number of communicating agents and the number of requested goods) increases, it may run into difficulties due to the slow and expensive communication. The proposed buying algorithm provides a potential solution to this problem: A buyer may send some volume of its requests for goods to its reputable sellers instead of all sellers, thus reducing the communication load and increasing the overall system performance.

In our proposed buying algorithm, a buyer selects a seller based on its own experience and doesn't communicate with other buyers for its decision[5]. We believe that this type of learning has certain advantages: Buyers can act independently and autonomously without being affected by communication delays (due to other buyers being busy), the failure of some key-buyer (whose buying policy influences other buyers), or the reliability of the information (the information received from other buyers may not be reliable). The resultant system, therefore, should be robust [62].

[5]See Section 6.3.1 for a discussion of the challenges in designing a marketplace that allows for such communication.

5.2.2 Design Decisions

This subsection discusses the value of several design decisions that we have made in order to improve our model.

Why Reinforcement Learning

We first provide justification for why we chose reinforcement as a learning method for our agents. Clearly, an electronic marketplace is an uncertain environment where buying and selling agents know little about one another and the environment may change any time (e.g., new agents may enter the market, existing agents may leave the market, information such as prices, the availability of products, product quality etc. may be altered). As defined in Section 2.2.1, reinforcement learning explicitly considers the problem of an agent that learns from interaction with an uncertain environment in order to achieve a goal. The agent must learn what to do (i.e., how to map situations to actions) so as to maximize a numerical reward signal. The learner is not told which actions to take as in most forms of machine learning, but instead must discover which actions yield the most reward via a trial-and-error search. It is this special characteristic of reinforcement learning that makes it a naturally suitable learning method for trading agents in market environments.

The suitability of reinforcement learning can also be seen if we take a close look at the activities of a buyer and a seller in an electronic marketplace. The buyer observes the bids submitted by sellers, selects a seller, and receives the good from that seller. The seller observes the request for good from a buyer, selects a price to sell that good, and receives profit (if any) from the transaction. In general, these agents get some input, take an action, and receive some reward. Indeed, this framework is the same framework used in reinforcement learning, which explains

why we chose reinforcement learning as a learning method for our proposed agents. The experimental results reported in Section 4.1 confirm that in an electronic market environment, a reinforcement learning agent fares significantly better than a non-learning agent.

The fact that reinforcement learning is adopted by the 0-level agents presented in [77, 78] also serves as a motivation for our consideration of this learning approach. However, the expected value function learned by a 0-level buyer and the expected profit function learned by a 0-level seller only take the price as their single variable. This is problematic in marketplaces where various sellers may offer different product qualities, and where multiple buyers may use different functions to evaluate the goods they purchase. We overcome this problem by introducing variable s (representing sellers) and variable b (representing buyers) into the expected value function and the expected profit function, respectively. More discussion of this issue is provided in the subsection labelled "Buyers Keep Track of Sellers' Behaviours and vice versa" under this section.

Disreputable Sellers

Reinforcement learning would allow a buyer to learn the best seller to purchase from, so as to maximize its expected product value, if all sellers in the market offer their goods with fixed quality. However, this is probably not the case, especially in a market environment where some sellers may deliberately cheat by repeatedly offering a good of high quality, followed by really low quality ones. This reality suggests us to let a buyer model sellers' reputation, and consequently put more trust on the sellers that have had a good history in doing business with the buyer. The reputation mechanism then pre-screens sellers in the market to form a set of trustworthy sellers to which reinforcement learning will be applied. The motivation

behind the use of reputation modelling is simply an analogy to human society in which shoppers often prefer to purchase goods from a list of merchants that are trustable (or reputable) to them. An early version of our model therefore was built with the reputation threshold Θ, used to form the set of reputable sellers. However, the model did not have the disreputation threshold θ and the corresponding set of disreputable sellers at that time. In an experiment later on we discovered that our proposed algorithm did not protect buyers well enough from dishonest sellers. Figure 5.5(a) shows the result of an early experimentation with large sized marketplaces, namely the profit values made by the dishonest sellers from a buyer using the combination of reinforcement learning and reputation modelling without the implementation of the set of disreputable sellers.

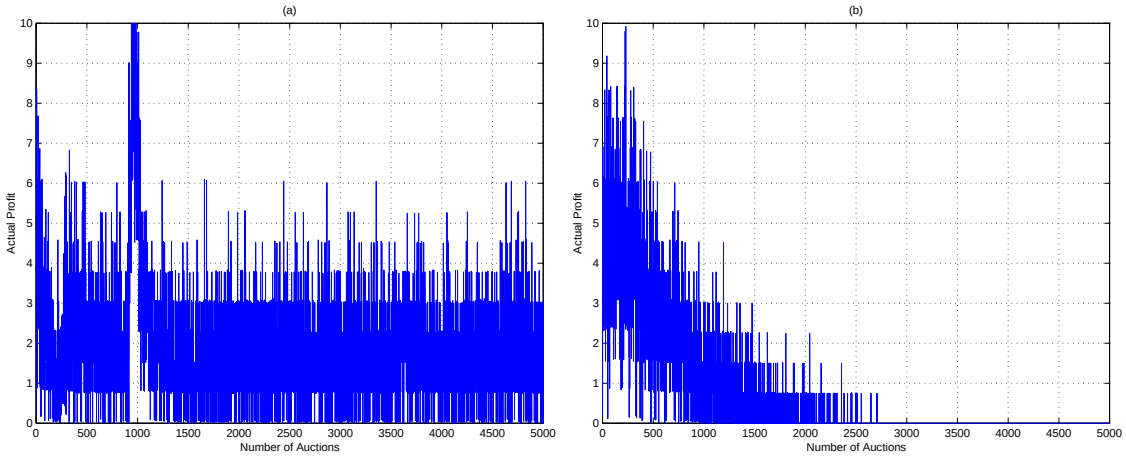

Figure 5.5: Profit values made by the dishonest sellers from a buyer using an early version of our model which did not implement the set of disreputable sellers (a), and from a buyer using the current version of the model (b).

As we can clearly notice, the dishonest sellers were able to make profit from the buyer throughout the number of auctions tested. In other words, our early buyer

could not avoid interaction with the dishonest sellers in the long run. The main reason is that although the buyer gave priority to considering reputable sellers in exploitation steps, it still chose the dishonest sellers in exploration steps, despite the fact that it had been cheated repeatedly by these sellers in many times. To eliminate this undesirable situation, we introduced into our model the disreputation threshold θ to form the set of disreputable sellers, with whom the buyer would not interact even in exploration steps. Of course, θ should be set low enough so that the buyer would not mistakenly place any "innocent" seller in the disreputable set. We provide suggestions on reasonable values for θ in Section 3.4. Figure 5.5(b) shows the profit made by the dishonest sellers from a buyer using the current version of our model, in the same experimental settings as that of Figure 5.5(a). Indeed, the implementation of the disreputable threshold θ and its corresponding disreputable set has done what we desire: The reputation mechanism provides much better protection for buyers. The dishonest sellers are no longer able to make profit from our proposed buyers in the long run.

Demanded Product Value

In the early version of our model, a buyer b did not maintain the demanded product value $\vartheta^b(g)$ for the good g that it purchased. The reputation rating $r^b(s)$ of a seller s was then updated based on the comparison of the true product value v^b with the expected product value f^b. That is, $r^b(s)$ would be increased if $v^b \geq f^b$; otherwise, it would be decreased. We later found that these updating conditions were inappropriate, since it would be possible for some seller s^*, who offered good g with very low product value, to get into the set of reputable sellers if it continually offered g with that low value. This could happen because the updating scheme would increase the reputation rating of s^* whenever the product value that s^*

offered was equal to the value that it offered before, regardless whether or not that value would satisfy the buyer's need. Obviously, it would be inappropriate for a buyer b to place in its reputable set some seller s^* that had always offered to it a good with unsatisfactory value. We solved this problem by introducing into our model the demanded product value ϑ^b, which serves as buyer b's threshold for the true value v^b of good g. For a good g, buyer b can determine $\vartheta^b(g)$ based on the lowest quality that it would like g at least to have, the highest price that it would be willing to pay for that quality, and the relative importance between these two factors, as discussed in Section 3.4. This makes sense because naturally for a particular good g, buyer b should have in its mind the lowest quality that it would demand for g, and the highest price that it would agree to pay for that quality. As a result, in our current model, a buyer b updates the reputation rating $r^b(s)$ of a seller s based on whether or not the true product value meets its demanded product value (equations (3.7) and (3.8)). We believe that these updating conditions are appropriate, resulting in reasonable assignment of reputation ratings to sellers.

Buyers Keep Track of Sellers' Behaviours and vice versa

In our proposed buying algorithm (Section 3.2.1), we let a buyer b keep track of sellers' behaviours by introducing the variable s in the buyer's expected value function $f^b(g, p, s)$. This is essential in marketplaces where sellers may alter the quality of their goods. In such market environments, some good g may be offered at the same price p but with greatly different qualities (and accordingly greatly different product values) by different sellers. Buyer b therefore needs to keep track of which seller offering which good g, in order to appropriately learn the best seller from which to purchase g. Otherwise, an undesirable experience in buying g at price p from some bad seller will mistakenly prevent buyer b from purchasing g at

that price from other sellers, even though they are good sellers and providing g with great value that meets the buyer b's demanded product value.

Similarly, in the proposed selling algorithm (Section 3.2.2), a seller s keeps track of buyers' behaviours by using variable b in its expected profit function $h^s(g, p, b)$. Since multiple buyers may have different opinions in evaluating the goods they purchase (by adopting different true product value functions v^b), seller s obviously needs to track the behaviours of buyers. If s does not do so, then a failure in selling good g at price p to some buyer b will falsely keep it from making the same offer to other buyers, including those buyers who value g in a different way than b does, and therefore may be happy to purchase g at price p from s.

It should be inferred from the above discussion that in a marketplace where sellers offer goods with the same quality, buyers simply need to select the seller with the lowest price, and therefore do not need to track sellers' behaviours (and accordingly, sellers' reputation). Similarly, in a marketplace where different buyers have the same point of view in evaluating the same good (by making use of the same true product value function), sellers do not need to track buyers' behaviours, and therefore their profit function is reduced to only $h^s(g, p)$. We discuss as part of our future work the case where sellers, instead of tracking individual buyers' behaviours, may divide buyers into groups and keep track of groups of buyers' behaviours (Section 6.3.2).

Quality as Cost

In our model, the quality q of a product is represented by a single numerical value, but it could be a multi-faceted concept. As we discussed in Section 3.4, a buyer b may consider q as a combination of several factors, e.g., physical product char-

acteristics, technical support, delivery punctuality etc. In this case, buyer b may calculate q as a weighted sum of these factors.

In the experiments reported in Chapter 4, we chose the quality of a good to be equal to the cost for producing that good. Our choice supports the common assumption that it costs more to produce high quality goods. This assumption means that in general, quality should have a direct proportional relation to cost. Without loss of generality, we chose this relation to be the simplest linear form, for the purpose of simplicity and ease in implementation.

5.3 Reputation Mechanisms

We discuss in this section some possible advantages of the reputation mechanism used in our proposed buying algorithm.

In an open market environment, new sellers may enter the market anytime. Moreover, if the market is large enough, there may be some sellers that a buyer b has not yet interacted with. Obviously, we need some reasonable way for buyer b to assign its initial reputation ratings to these sellers. As opposed to [16], our reputation function maps the set of sellers to the range $(-1, 1)$, which allows buyer b to naturally assign the neutral value of zero to new sellers or to sellers that b has not yet had experience about, hence initiating the reputation models of these sellers.

In contrast to [16] and [55], by introducing the reputation and disreputation thresholds (Θ and θ), our reputation mechanism allows a buyer to partition the set of all sellers in the market into three disjoint subsets, namely the reputable sellers, the disreputable sellers, and the neither reputable nor disreputable sellers

(i.e., those sellers that the buyer has not yet had enough information to decide on their reputation). As a result, the buyer can quickly select a reputable seller while avoiding the disreputable ones, whenever a purchase decision needs to be made. Also, it can have easy access to the sellers whose reputation it has not yet decided on in order to discover new reputable sellers. We believe that this design is especially suitable for electronic marketplaces where the rate of transactions between trading agents is usually high.

One of the most important issues for any model of reputation is how to adjust it. A good adjusting (or updating) scheme should base its updating conditions on appropriate factors in order to achieve reasonable modelling of reputation; yet the updating scheme should not be too complicated to implement. The proposed reputation mechanism uses a feasible updating scheme (equations (3.7) and (3.8)) the operation of which is based on the comparison of the true value of the good purchased by a buyer and the product value demanded by that buyer. As mentioned in Section 5.2.2, these updating conditions depend on three factors, namely *(i)* the price of the good, *(ii)* the quality of the good, and *(iii)* the relative importance between price and quality as viewed by the buyer. Since these are the most significant factors for the buyer to determine the value of the good purchased, we believe our updating scheme is appropriate for market settings and therefore provides reasonably accurate modelling of sellers' reputation.

Our reputation mechanism is motivated by [84]. As in [84], our updating scheme encourages sellers to offer high value goods and discourages them from offering low value ones, by making use of the cooperation and non-cooperation factors (μ and ν), respectively. However, in contrast with [84], we do not keep these factors fixed, but take one step further to vary them as increasing functions of the true product value (see equations (3.12) and (3.13)) in order to implement the common idea that

a transaction with higher value should be more appreciated than a lower one and vice versa. In other words, the reputation rating of a seller who offers higher value goods should be better increased, and the reputation rating of a seller who greatly disappoints should be more seriously decreased. We also implement the traditional assumption that reputation should be difficult to build up but easy to tear down by introducing the penalty factor $\lambda > 1$, by which we penalize a non-cooperative seller λ times greater than we would award a cooperative one. Section 3.3 provides a theoretical exploration of desirable values for λ.

Our modelling of reputation focuses on the individual dimension of reputation which allows an agent to compute reputation ratings of others using its own experience. This is sufficient for our market model where buyers learn to select sellers based on their own experience and do not communicate with other buyers in the market[6]. We discuss, as a future research direction in Section 6.3.1, a possible market environment where the social dimension of reputation is needed to allow an agent to compute reputation ratings based on not only its own but also others' experience. We also describe in Section 6.3.5 some possible extensions of our reputation mechanism for multi-agent systems other than market environments.

5.4 Chapter Summary

This chapter provides a detailed discussion of the value of our model. Section 5.1 contrasts and compares our model with other related e-commerce agent models. In particular, Section 5.1.1 reviews several models in contrast with our model, namely BargainFinder [3], Jango [14, 29], Kasbah [9], the model of shopbots and pricebots [21, 22, 23], and the nested agent model [77, 78]. By analyzing these related models

[6]Section 5.2.1 discusses some advantages of this approach.

we argue that the proposed trading agents of our model are autonomous, able to learn to adapt themselves to environmental changes, and able to avoid costly computation in making decisions. Section 5.1.2 respectively compares our buyers and sellers to 1-level buyers and 0-level sellers proposed in [77, 78], which we consider as the most related work to ours. Two comparison criteria are measured, namely satisfaction level and computational costs. The experimental results show that our proposed buyers and sellers outperform the 1-level buyers and 0-level sellers, respectively.

Section 5.2 discusses the merits of our model. It begins in Section 5.2.1 with a discussion on the following potential advantages that the proposed model may offer: First, the proposed model can be used in designing electronic marketplaces and effective participant trading agents. Secondly, the proposed buying algorithm should reduce computational costs and bring greater satisfaction to buyers. Thirdly, the proposed selling algorithm should give more opportunities for sellers to make sales and therefore increase their profits. Fourthly, the proposed buying algorithm provides a potential solution to reduce communication load and hence improve the overall system's performance. Finally, the fact that buyers do not rely on communication with one another should result in more robust systems. Section 5.2.2 discusses the value of certain design decisions that have been made within the model. These decisions include the choice of reinforcement learning as a suitable learning method for trading agents in market environments, the necessity of identifying the set of disreputable sellers, the need for introducing the demanded product value, and the need for letting buyers' keep track of sellers' behaviours and vice versa.

Section 5.3, the last section of the chapter, comments on the advantages of the reputation mechanism used in the proposed algorithm. First, it allows for a

natural initialization of neutral reputation rating on new sellers and those sellers that a buyer has not yet had experience with. Secondly, it enables the buyer to quickly select a reputable seller while staying away from the disreputable ones, and also to have easy access to the neither reputable nor disreputable sellers for discovering new reputable sellers. Thirdly, the proposed reputation mechanism provides a feasible and appropriate updating scheme for market settings. Finally, the proposed reputation mechanism implements two reasonable traditional ideas, namely *(i)* reputation should be difficult to build up but easy to tear down, and *(ii)* the extent to which the reputation rating of a seller is increased or decreased should be based on the value of the transaction that it offers.

Chapter 6

Conclusions

In this chapter we provide a summary of the thesis followed by its contributions. We end the chapter with a number of future research directions.

6.1 Thesis Summary

This thesis addresses the problem of how to develop learning algorithms that guide the behaviours of buying and selling agents participating in electronic market environments. Section 3.1 presents an agent market model suitable for e-commerce applications. This agent market model is equipped with a feasible contract-net like buying and selling process that allows trading agents in the market to exchange goods with one another. It also takes into account the open, dynamic, and unpredictable natures of a multi-agent market environment. Section 3.2 proposes our reputation-oriented reinforcement learning based algorithms for buyers and sellers, respectively. Our buyers learn to maximize their expected values of goods using reinforcement learning. In addition, they model sellers' reputation by dynamically

maintaining sets of reputable and disreputable sellers, and consider selecting suitable reputable sellers while avoiding the disreputable ones. This strategy enhances the buyers' chance of purchasing high value goods and reduces their risk of receiving low value goods, and therefore brings better satisfaction to the buyers. Our sellers learn to maximize their expected profits by using reinforcement learning to adjust prices, and also by altering product quality in order to provide better customized goods to meet the buyers' specific demands. Section 3.3 theoretically shows that a dishonest seller may not cause infinite loss to a buyer if the buyer properly sets its penalty factor λ while modelling sellers' reputation. Section 3.4 discusses the parameters used in the proposed algorithms and provides general guidelines for setting these parameters.

Chapter 4 presents the experimentation that we have performed to measure the value of our model on both microscopic and macroscopic levels. On the micro level, we were interested in examining the individual benefit of agents, particularly their level of satisfaction. Our experimental results confirm that in both modest and large-sized marketplaces, buyers and sellers following the proposed algorithms achieve better satisfaction than buyers and sellers who only use reinforcement learning, with the buyers not modelling sellers' reputation and the sellers not considering adjusting the quality of their goods. On the macro level, we studied how a market populated with our buyers and sellers would behave as a whole. Our results show that such a market can reach an equilibrium state where the agent population remains stable (as some sellers who repeatedly fail to sell their goods may decide to leave the market), and this equilibrium is beneficial for the participant agents.

Chapter 5 offers a detailed discussion on the value of the proposed model. In particular, Section 5.1 compares and contrasts our model with other related e-commerce agent models. This section also provides an experimental comparison

between our trading agents and those proposed in [77, 78], which we consider as the most related work to ours. The experimental results indicate that our proposed agents are able to achieve better performance in terms of satisfaction and computational costs. Section 5.2 discusses the merits of the proposed model. It presents a number of potential advantages that the model may offer, including the effectiveness and feasibility of the model, the greater satisfaction of agents following the proposed algorithms, the potential solution to reduce expensive and slow communication load, and the robustness of a system populated with the proposed agents. This section also discusses the value of several design decisions that we have made within the model. Section 5.3 provides detailed comments on the advantages of the reputation mechanism which our proposed buyers use to model sellers' reputation in the market: First, it allows a buyer to naturally assign neutral reputation rating to sellers, hence initiating models of entirely new sellers. Secondly, it enables a buyer to quickly access the reputable sellers while staying away from the disreputable ones, and also to explore the neither reputable nor disreputable sellers in discovering new reputable sellers. Thirdly, it provides a feasible updating scheme that allows for the reputation models of sellers to be adjusted appropriately over time. Finally, the proposed reputation mechanism implements two reasonable and traditional assumptions about reputation, namely reputation should be difficult to build up but easy to tear down, and a transaction with higher value should be more appreciated than a lower one.

6.2 Contributions

The contributions of this thesis are summarized in the following bullet points:

- We present a feasible agent market model which is suitable for e-commerce

applications. Our market model takes into consideration several important factors of a multi-agent market environment, namely its open, dynamic, uncertain and untrusted natures.

- We propose a learning algorithm for buying agents in electronic marketplaces that uses a combination of reinforcement learning and reputation modelling. We have theoretically shown that by properly setting the penalty factor, a proposed buyer can avoid suffering infinite loss caused by a dishonest seller (Section 3.3). We have experimentally demonstrated the value of our proposed buying algorithm in both modest and large sized marketplaces: Buyers following the proposed algorithm obtain better satisfaction level than buyers using reinforcement learning alone. In particular, buyers following the proposed algorithm are able to avoid interaction with dishonest sellers (Section 4.1).

- We also propose a learning algorithm that enables selling agents to learn to maximize their expected profits by using reinforcement learning to adjust prices for and by providing more customized value to their goods. The value of adjusting product quality to provide more customized product value (in addition to using reinforcement learning) is confirmed in Section 4.1, where sellers following the proposed algorithm have been experimentally demonstrated to achieve better satisfaction than sellers using reinforcement learning alone.

- By proposing a feasible agent market model as well as effective learning algorithms that govern how buying and selling agents in an electronic marketplace should act and make decisions, this thesis should provide an example for AI-system designers in developing effective trading agents and desirable market environments.

- The thesis demonstrates that reputation modelling can be used as an effective means to deal with dishonest, unreliable agents in a marketplace. It also presents a specific framework for representing and updating the reputation of sellers in electronic marketplaces. Further, the thesis shows that reputation modelling can be incorporated with reinforcement learning techniques to design intelligent agents that participate in an open, dynamic, uncertain and untrusted market environment.

- Finally, the work in this thesis shows that market-based multi-agent environments provide a useful test-bed for the application of AI-learning techniques, especially reinforcement learning.

6.3 Future Work

This thesis present a number of possibilities for future research which are listed below.

6.3.1 Buyers Forming Neighbourhoods

We would like to investigate more sophisticated learning algorithms that allow agents to cooperate with one another to better achieve their goals. Specifically, we are interested in considering the case where buyers in the market form neighbourhoods such that within a neighbourhood they inform one another of their knowledge about sellers. The buyers then use their own knowledge combined with the informed knowledge to make purchase decisions. Interesting issues to explore may include the following:

How Neighbourhoods Should Be Formed

At this time we foresee two possibilities based on which neighbourhoods of buyers may be formed, namely geography and similarity. The first one suggests that buyers that are geographically close to one another may form a neighbourhood. Examples are buyers that represent human users in the same company, organization, community etc. The advantage of geography-based neighbourhoods is that members of a neighbourhood should enjoy fast and inexpensive communication. However, they may have different ways to value the goods they purchase, and accordingly different views on sellers' reputation. In this case, a method for interpolating and hence making use of the knowledge exchanged by members of different views is obviously needed. The second possibility suggests that buyers that have similar views on evaluating the goods they purchase may form a neighbourhood (e.g., buyers that utilize similar true product value functions). Since these buyers should also have similar views on the models of sellers' reputation, they should greatly make use of the knowledge about sellers that they exchange with one another. Nevertheless, the tradeoff is that they may incur slow and expensive communication, especially when they are scattered geographically far away from one another.

What Knowledge to Exchange and How to Use It

We think that buyers should exchange with each other their knowledge about sellers' reputation. As an example, consider a neighbourhood N consisting of n buyers, namely $N = \{b_1, b_2, ..., b_n\}$. A buyer b_i may exchange with its neighbours the reputation rating $r^{b_i}(s)$ of every seller s in the market. In return, b_i receives the reputation ratings $r^{b_1}(s)$, $r^{b_2}(s)$, ..., $r^{b_{i-1}}(s)$, $r^{b_{i+1}}(s)$, ..., $r^{b_n}(s)$ from its neighbours, namely buyers b_1, b_2, ..., b_{i-1}, b_{i+1}, ..., b_n. Assume that buyer b_i equally trusts its

neighbours, and that all members of the neighbourhood utilize similar functions to evaluate the goods they purchase. Thus, one possible way for buyer b_i to make use of the exchanged information is to simply update its reputation rating with the average value of all ratings (including its own one):

$$r^{b_i}(s) \leftarrow \frac{r^{b_1}(s) + r^{b_2}(s) + \ldots + r^{b_i}(s) + \ldots + r^{b_n}(s)}{n} = \frac{\sum_{i=1}^{n} r^{b_i}(s)}{n} \tag{6.1}$$

This approach should work in marketplaces where there is sufficient supply of goods. In such markets, buyers do not compete with one another for goods and therefore should provide trustworthy information. However, in marketplaces where supply is lacking, some buyers may have the incentive to give misleading information due to competition. This situation initiates the need for a social trust (or reputation) model that allows a buyer, say b_1, to compute the trust (or reputation) rating of another buyer, say b_2, based on not only its own experience but also the experience of other buyers about b_2. This is especially essential when b_1 has not yet had enough experience about b_2. To us, the latter is an interesting case and deserves exploration. Some researchers have been working on modelling trust (or reputation) with this social aspect [16, 55, 84, 85]. An examination of the existing models is therefore required, with the possibility of developing a new model that is suitable for the task. In particular, Breban and Vassileva [6, 76] have examined the use of trust to determine coalition formation for improved buying activities. Their approach uses a trust evolution function, based on that of [30]. For future work, it would be useful to study how to begin with the model proposed in [6, 76] but extend it to include a richer model of trust, building on the framework for reputation modelling included in this thesis.

How Often Buyers Should Communicate

We predict that this form of transferring knowledge should be beneficial (in terms of speeding up the learning process) for new buyers who have just entered the market and who therefore can use the experience of existing buyers to make satisfactory purchase decisions, without having to undergo several trials to build up enough experience for themselves. In addition, for a marketplace where new sellers frequently join in, buyers may need to communicate with one another periodically so that those buyers who have had experience with the new sellers may share their knowledge with the buyers who have not. Furthermore, we need a clear specification that enumerates all situations in which a buyer may be necessarily triggered to communicate with its neighbours.

How Much a Communicating Buyer Gains

Intuitively, in a marketplace where buyers do not have incentives to provide false information and therefore can trust one another, a communicating agent should be able make more informed purchase decisions by using the knowledge shared by its neighbours. This probably results in the buyer's purchases of higher value goods and hence better satisfaction, compared to the case where the buyer does not communicate with its neighbours. However, the buyer must also incur a cost due to the necessary communication. A careful cost-benefit analysis is thus needed to justify the usefulness of the approach.

6.3.2 Sellers Modelling Groups of Buyers' Behaviours

We discussed in Section 5.2.2 that a seller essentially needs to track the behaviours of buyers when the buyers have different opinions in evaluating the good they pur-

chase, by adopting different true product value functions. However, the modelling of buyers' behaviours may not be necessary when buyers in the market have similar evaluation functions for the goods they purchase.

The above discussion suggests that it is possible to explore an additional version of the proposed selling algorithm in which a seller divides buyers into groups that use similar true product value functions and keeps track of groups of buyers' behaviours, instead of individual buyers' behaviours. Suppose, a seller s divides buyers into n groups, namely Γ_1, Γ_2, ..., Γ_n, then this additional version can be obtained from the proposed selling algorithm described in Section 3.2.2 by substituting b with Γ_i in the expected profit function $h^s(g, p, b)$, the production cost function $c^s(g, b)$, and the actual profit function $\phi^s(g, p, b)$, where Γ_i is the group that buyer b belongs to.

We believe that the approach for sellers to model groups of buyers should have at least the following advantages:

- Since the number of buyer groups may be considerably less than the number of all buyers in the market, this approach should reduce the size of a seller's internal database and accordingly the search space, resulting in improved performance when the seller searches for an optimal price to sell some good.

- More importantly, this approach allows a seller to significantly reduce the number of customized versions of a good to be maintained. That is, instead of producing so many versions of a good tailored to meet specific demands of all individual buyers in the market, the seller now only needs to produce a much smaller number of versions of the good, which are customized for the different groups of buyers.

However, this approach also presents several issues to be addressed.

- How to measure the similarity between buyers' true product value functions (especially when the buyers may not explicitly let sellers know their functions) in order to classify buyers into groups.
- How to update the models of groups of buyers appropriately over time.
- How to detect if any buyer has changed the way it evaluates the goods it purchases (i.e., changed its true product value function) and therefore should be removed from its current group and placed in another group, resulting in both groups to be updated.
- Finally, a formal analysis is needed to justify the approach, considering its advantages and the complexity in addressing the above issues.

6.3.3 Negotiation

It is possible to add negotiation into our proposed algorithms. Negotiation is a process in which two or more agents communicate their positions (which are probably conflicting) and then try to reach an agreement by making concessions or searching for alternatives (page 104 of [81]). For instance, a buyer in our market model, after deciding on a suitable seller, may start negotiating with the seller over the price and possibly the quality of the good. Obviously, implementing this approach requires that a negotiation mechanism (i.e., a set of negotiation rules) be built into the agent market model. This set of rules should clearly specify how the agents have to interact in order to come to a joint decision. The negotiation mechanism should be designed such that the agents in the market, regardless of their origin, capabilities or intentions, will interact fairly and efficiently. In other words, it should have at least the following properties:

- Efficiency: The agents should not waste resources in reaching an agreement.
- Simplicity: The negotiation process should not impose costly computational and bandwidth demands on the agents.
- Fairness: The mechanism should not be biased against any agent due to arbitrary or inappropriate reasons.

A number of efforts have been done in the field of agent negotiation [12, 13, 32, 39, 63]. We believe that a careful analysis should be performed on existing works to find an appropriate negotiation mechanism with the possibility of modifying or developing an entirely new one, in order to embed negotiation into the proposed algorithms. One promising approach is outlined in [44]. In this research, buyers and sellers negotiate based on the preferences of their users, incorporating a modelling of the user's risk attitude.

An appropriate contracting mechanism is also needed for the market model when implementing this negotiation approach. While the negotiation mechanism concerns the process of finding a joint agreement between agents, the contracting mechanism concerns how the agents should commit themselves to the agreement. In general, there are two standard types of contracts, namely unbreakable and breakable contracts. For an unbreakable contract, once the contract is made, it is binding: The agents can not back out but have to follow through with it, no matter how future events may occur [38, 52, 56]. A breakable contract, however, allows agents to de-commit (i.e., to be free from the obligations of the contract) upon certain conditions or events. Two common forms of breakable contracts are contingency contracts [51] where the obligations of a contract are made contingent on future events, and levelled commitment contracts [1, 58] where an agent that

wants to de-commit can do so by paying a de-commitment penalty to the other party.

Breakable contracts have a significant advantage over unbreakable ones: They allow agents participating in dynamic environments to flexibly adjust their decisions according to environmental changes, which may make existing contracts unprofitable or unfavorable. Nevertheless, they also present some disadvantages. Contingency contracts have at least three problems when used among self-interested agents [81]: First, they become awkward as the number of relevant future events to be monitored increases. Further, the value of the contract may depend on combinations of events, resulting in a potential combinatorial explosion of items to be conditioned on. Secondly, it is often impossible to enumerate all possible relevant future events in advance. Thirdly, a relevant event sometimes is only observable by one of the agents, which may give this agent an incentive to lie to the other party of the contract. Levelled commitment contracts, even though they look intuitively more appealing, are not obviously beneficial. First, the gain of the agent that breaks the contract may be smaller than the loss of the other party, resulting in an overall loss of the society. Secondly, the agent may commit insincerely: A sincere agent would de-commit whenever the difference between the gain from de-commitment (by contracting with another party) and the de-commitment penalty (that it has to pay to the current party) is greater than the value of the current contract. However, a self-interested agent may be more reluctant to de-commit if it takes into account the possibility that the other party may also want to de-commit, in which case the former agent would be free from the contract obligations, would not have to pay the penalty to the latter but instead collect the penalty from the latter. Due to such reluctant de-committing situations, a contract may end up being kept, while breaking it would be better from the social welfare perspective.

Essentially, a formal analysis of existing contracting mechanisms or the design of new ones needs to be done if we would like to equip our proposed algorithms with the negotiation feature.

6.3.4 Auctions

A possible research path to which our work may be extendible is the design of versatile trading agents that can participate not only in common marketplaces, but also in various auction environments. Auctions have been known as popular mechanisms for allocating resources among agents in multi-agent environments [59]. There are several auction types which require different market mechanisms accordingly. In an English auction, each bidder is free to raise her bid. The auction ends when no bidder is willing to raise her bid anymore, and the highest bidder wins the item at the price of her bid. In a Dutch auction, the auctioneer continuously lowers the price of the item until one of the bidders agrees to take the item at the current price. In a Vickrey auction, each bidder submits one bid without knowing the others' bids. The highest bidder wins, but at the price of the second highest bid. In a continuous double auction, sellers continuously decrease their asks while buyers continuously increase their bids until the highest bid is not less than the lowest ask, which results in a transaction. In a parallel auction, items are open for auction simultaneously, and a bidder may place her bids for different items at the same time. In a combinatorial auction, bidders are allowed to place bids for arbitrary combinations of items.

Combinatorial auctions are particularly of interest to us because of their appropriateness in situations where bidders have preferences over combinations of items. For example, an agent a may want two goods, say g_1 and g_2, such that obtaining

one good is useless without the other. Bidding for g_1 and g_2 individually (in sequence or in parallel) is obviously not preferable since it exposes the agent to the risk of obtaining one without the other. In a combinatorial auction, the auctioneer has to decide on how best to allocate individual items to those combinations for which bids are placed, with the aim of maximizing the auctioneer's revenue. This problem is known as the *optimal winner determination problem*, which turns out to be NP-complete [59]. It is therefore interesting and challenging for researchers to find algorithmic solutions to this problem that are feasible and useful in practice.

A number of efforts have been invested in exploring solutions to the optimal winner determination problem. To our knowledge, these efforts can be represented by four lines of work, namely the Dynamic Programming Algorithm (DPA) by Rothkopf et al. [53], the Optimal Search Algorithm (OSA) by Sandholm [59], the Structured Search Algorithm (SSA) by Fujishima et al. [19], and the Stochastic Local Search Algorithm (named Casanova) by Hoos and Boutilier [25].

DPA solves the optimal winner determination problem by evaluating all possible allocations. It has running time $O(3^n)$, where n is the number of items to be auctioned [53]. It executes the same algorithmic steps regardless of which bids have actually been submitted. As pointed out in [59], this algorithm will quickly become infeasible when the number of items is increased above 25. Thus, the algorithm is only useful when the number of items is sufficiently small so that $O(3^n)$ operations can be carried out in a reasonable amount of time.

Unlike DPA, OSA makes use of various preprocessing and pruning techniques to exploit the bid structure, and therefore restrict the search. Hence, if the number of bids received is relatively sparse compared to the space of possible bids, OSA will achieve a much better performance than DPA. Furthermore, OSA is a *complete* algorithm, which means that if given enough time it will find an optimal solution.

OSA also has the useful *anytime* feature; that is, if the algorithm does not finish execution in the desired amount of time, it can be terminated prematurely and still guarantee a feasible solution that improves monotonically over time. Although OSA has been shown to perform reasonably well on problems of moderate size, it may become inadequate or even infeasible when problem instances are large, or when solutions are needed quickly. The reason for this is that OSA necessarily spends considerable amount of time "proving" that the solution it produces is optimal, at the expense of quickly providing high quality (though perhaps suboptimal) solutions.

SSA is also a complete algorithm with the anytime feature. It uses depth-first search (DFS) to find optimal solutions but cleverly structures the search space using preprocessing, caching, and pruning techniques to allow the search to find optimal solutions rather effectively. Although SSA seems similar to OSA in terms of exploiting the structure of bids to restrict the search, there is an important difference between the two approaches: OSA performs a secondary DFS to identify non-conflicting bids (i.e., bids whose combinations do not share items), whereas SSA's structured approach allows it to avoid considering most conflicting bids. For modest-sized problems, SSA demonstrates good performance in both finding optimal allocations, and as an anytime algorithm (providing good allocations prior to finding optimal allocations). However, as the problem size increases, it may become infeasible as in the case of OSA.

Casanova applies stochastic local search techniques to tackle the optimal winner determination problem. It has been experimentally shown in a number of test cases to work faster than SSA, especially in large problem instances where it is able produce relatively high quality solutions to the problems [25]. However, Casanova is an incomplete algorithm. That means, in practice, if given enough time, it may

find an optimal solution; but it can not be used to prove the optimality of any solutions it finds nor can it guarantee the quality of the solutions.

Due to the computational complexity of the optimal winner determination problem, we believe that there is still sufficient room for researchers to improve existing algorithms or to design entirely new ones. We are interested in taking part in this interesting task as one of our future research paths. Since the market mechanism of combinatorial auctions is completely different from that of common marketplaces that we consider in this thesis, we do not hope to re-use or modify our proposed algorithms and fit them into the combinatorial auction setting. In fact, different auctions require different market mechanisms, and accordingly different algorithms for their participant agents. Thus, to pursue this line of research, we need to carefully study the problem, analyze the advantages and drawbacks of existing solutions in order to invent new solutions that have more advantages than and avoid the drawbacks of the existing ones. In addition, two topics worth exploring as an extension of our current research are: *(i)* Whether it is possible to allow sellers to vary the quality of their goods in combinatorial auctions, and *(ii)* whether it is possible for buyers to include the modelling of reputation in parallel auctions.

6.3.5 Reputation Modelling

Electronic market-based multi-agent systems are typically untrusted environments where, as in human society, there may exist unreliable or dishonest trading agents. In such environments, it is very important for an agent a to know the reputation (or trustworthiness) of an agent b before a can initiate a commercial transaction with b. Consequently, the notion of reputation (or trust) in human society should be modelled in multi-agent electronic marketplaces.

Reputation and trust are two inter-related concepts and at times may be used interchangeably: If agent b is considered reputable by an agent a, it is trusted by a. Conversely, if agent b is considered disreputable by agent a, it is not trusted by a. A good modelling of reputation should help honest agents to find reliable trading partners, to avoid interaction with dishonest agents, and to cooperate with one another to weed the dishonest, antisocial agents out of the environment. Further, a desirable modelling of reputation should have at least the following properties:

(i) Individualized: The model should allow an agent to decide the reputation of another agent based on its own experience with that agent. Moreover, if should take into account personal opinions of agents (e.g., high value transactions should be regarded as more important than low value ones, reputation should be difficult to build up but easy to tear down etc.).

(ii) Socialized: The model should consider the social aspect of reputation. In particular, it should provide a way for an individual agent to use the experience of others to complement its own experience in deciding the reputation rating of a particular agent.

(iii) Ontologically-structured: Reputation may be considered as a multi-faceted concept. For instance, the reputation of being a good Internet provider is composed of the reputations of offering low price, high speed, and friendly customer service etc.

(iv) Computable: The model should provide a feasible method to combine several aspects of reputation (e.g., individual, social, and ontological) and quantify them into a unified, comparable measure. The computing process should not be too complicated and computationally expensive.

(v) Adjustable or evolutional: The model should be equipped with an effective updating scheme based on which models of reputation can be appropriately adjusted as the agents' experiences evolve (or change) over time.

(vi) Distinctive: The model should make a clear distinction between being disreputable and lack of knowledge about reputation. Accordingly, it should provide an appropriate way for an agent a to initialize the reputation model of some agent b who is neither reputable nor disreputable to a. That is, a does not have sufficient knowledge to decide on the reputation of b.

Recently, there has been a growing interest in modelling reputation (and trust), mainly driven by the advent of e-commerce [15, 16, 30, 47, 55, 84]. In general, although existing models possess certain advantages, they also have a number of major drawbacks. The reputation models used by eBay [15] and Nextag [47] electronic marketplaces are not well individualized. In particular, these models do not differentiate private trading experiences: They increase the reputation rating of a trader by one after a satisfactory transaction with that trader, and similarly decrease the reputation rating of the trader by one after an unsatisfactory transaction, regardless of the transaction value. In other words, they unfairly consider high value transactions just as important as low value transactions. The model proposed in [16] does not clearly distinguish between distrust and lack of knowledge about trust. Moreover, it does not provide a method to combine different trust ratings into a unified value. The reputation model presented in [55] suggests a very complex method for calculating reputation rating without any guidelines for choosing the normalized factors involved in the calculation process. The method to compute social reputation ratings via propagation described in [84] may result in conflicting values. Also, its updating scheme using specific evidence factors leads to the similar

drawback as the cases of eBay and Nextag marketplaces. Section 2.3 provides a more detailed description and analysis of the advantages and shortcomings of the models presented in [16], [55], and [84].

We are interested in further exploring the issue of reputation in the context of electronic market-based multi-agent systems. The reputation mechanism that we propose in this thesis has a number of advantages as discussed in Section 5.3. It possesses almost all the above-mentioned desirable properties except for the social aspect. An obvious step for future work, therefore, is to try to incorporate a social dimension into our modelling of reputation, as we investigate the case where buying agents form neighbourhoods to exchange knowledge about sellers (Section 6.3.1). Our goal is to sharpen our reputation mechanism so that it will become more and more effective in protecting honest agents and accordingly in dealing with dishonest, unreliable, or anti-social agents.

One specific topic to examine more closely, as we extend our study of reputation mechanisms, is to determine the value of modelling the time of transaction with an agent. Jonker and Treur [30] propose a formal trust model in which their trust update function makes use of an inflation parameter to implement the fact that recent experiences are more important than older ones. Our current thought is that in market environments, it is the value of a transaction rather than its recency that should matter. That is, in our view, an agent a_1 should not forget the fact that an agent a_2 was not cooperative with it in a one-million-dollar transaction in the past, and trust agent a_2 just because a_2 has recently been cooperative with it in a few one-dollar transactions. However, we would like to study the pros and cons of this model more carefully in future work.

In addition, studying how best to integrate reputation models into more complex multi-agent system environments is another possible direction for future research.

Our modelling of reputation should be applicable for multi-agent systems (other than market environments), where unreliable or dishonest agents may exist. For example, in the predator-prey problem (described in Section 2.2.4), the information exchanged by the predators may not be reliable due to their competition to catch the prey. It is therefore obvious that a reputation (or trust) mechanism is needed to help the predators to measure how much they should trust one another, and accordingly to what extent they should follow one another's advice.

Appendix A

Example of an Auction for Information Goods

This appendix provides an example of how an auction is possibly carried out in our agent market model, to clarify terms such as *auction*, *announcing a need for a good* and *adjusting the quality of a good*.

Consider an electronic marketplace of information goods and services where a query-answering service is defined as a good, and all agents providing query-answering services are considered as sellers of the same good. Suppose that a buying agent b_i would like to know about economical vehicles. According to our buying and selling protocol (Section 3.1), b_i announces its need by electronically sending to all sellers in the market a query, say *most fuel efficient vehicles*. We view this action of buyer b_i as the initiation of an *auction* in which b_i is the auctioneer, and those sellers who will submit price bids to provide answers to the query are bidders.

Suppose that n sellers ($n > 1$), namely $s_1, s_2, \ldots, s_n$, send their price bids

$p_1, p_2, \ldots, p_n$ respectively, to buyer b_i to deliver their answers to b_i's query. Further, suppose that buyer b_i follows the proposed buying algorithm. It therefore chooses among its reputable sellers a suitable one, say s_j $(1 \leq j \leq n)$, in order to purchase the answer from (equation (3.3)). In our terminology, seller s_j is said to be *winning the auction.* Buyer b_i then pays seller s_j (probably by sending its credit card information or by direct electronic fund transfer) and receives from seller s_j answer a_j to its query. Answer a_j may look like the following[1]:

TOYOTA COROLLA $< Picture\ of\ vehicle >$

Engine: 1.8 L, 4 cylinders

Transmission: Manual 5 speed

Fuel Consumption:

City 7.1 L/100 km (40 mpg)

Hwy 5.3 L/100 km (53 mpg)

Annual Fuel Cost: $758

Annual Fuel Use: 1131 L

FORD FOCUS $< Picture\ of\ vehicle >$

Engine: 2.0 L, 4 cylinders

Transmission: Manual 5 speed

Fuel Consumption:

City 8.6 L/100 km (33 mpg)

Hwy 6.0 L/100 km (47 mpg)

[1]This is only an example and therefore should NOT be based on to make vehicle purchase decisions.

Annual Fuel Cost: $892

Annual Fuel Use: 1332 L

VOLKSWAGEN JETTA *< Picture of vehicle >*

Engine: 1.9 L, 4 cylinders

Transmission: Manual 5 speed

Fuel Consumption:

City 5.6 L/100 km (50 mpg)

Hwy 4.3 L/100 km (66 mpg)

Annual Fuel Cost: $605

Annual Fuel Use: 903 L

HYUNDAI ELANTRA *< Picture of vehicle >*

Engine: 2.0 L, 4 cylinders

Transmission: Manual 5 speed

Fuel Consumption:

City 9.6 L/100 km (29 mpg)

Hwy 6.5 L/100 km (43 mpg)

Annual Fuel Cost: $984

Annual Fuel Use: 1469 L

Continued ...

Buyer b_i may determine the quality of answer a_j based on *(i)* the number of vehicle models listed, *(ii)* the relevance of the information provided, and *(iii)* the time it

took for answer a_i to arrive. In particular, buyer b_i may compute the quality of answer a_j as a weighted sum of these factors. Buyer b_i then calculates the true value of answer a_j using its true product value function, and updates its expected value function using equations (3.5) and (3.6). Assume that buyer b_i is satisfied with answer a_j, it hence increases the reputation rating of seller s_j based on equation (3.7).

Consider a seller s_k ($k \neq j$) that was unsuccessful in selling its answer a_k at price p_k to buyer b_i during the above-mentioned auction. If s_k follows our proposed selling algorithm, it will update its expected profit function using equation (3.15). Since s_k made no profit in the auction, its actual profit function is assigned zero value (equation (3.16)), resulting in the expected profit function being updated with a smaller value than before. This implies that seller s_k will probably choose some lower price than p_k to sell its answers to buyer b_i in future auctions, according to equation (3.14).

Assume that seller s_k once succeeded in making a sale to buyer b_i, but has been unsuccessful for a number of consecutive auctions since then, despite its adjustment of the price. Seller s_k therefore may believe that the quality of its answers has not met buyer b_i's need. As a result, it may decide to improve the quality of its answers by, for instance, performing one or a combination of the following:

- Providing more items in its answers, e.g., including German and Korean vehicles in addition to American and Japanese vehicles.
- Adding more relevant information, e.g., consumers' statistical information such as annual fuel cost, annual maintenance cost etc.
- Making the format of its answers more graphical, e.g., adding pictures of vehicles etc.

In general, a quality improvement should result in an increase in production cost (equation (3.17)). This is especially obvious for the above-listed improvements: In order to provide more items, s_k will have to establish more networked links to other manufacturers. More statistical information means more costs to compute or generate the information. More graphics require more storage (to contain them), faster connection (to transfer them), and graphical technology built in the system (to handle them).

Appendix B

Glossary of Mathematical Symbols

This appendix provides a glossary of the mathematical symbols that we used in describing our proposed algorithms. In particular, we list these symbols together with their meanings and the pages where they were first defined.

Symbol	**Meaning**	**First Defined**
G	Finite set of goods	p. 52
S	Finite set of sellers	p. 52
B	Finite set of buyers	p. 57
P	Finite set of prices	p. 52
Q	Finite set of qualities	p. 53
$r^b(s)$	reputation rating assigned to seller s by buyer b	p. 52
Θ	Reputation threshold ($0 < \Theta < 1$)	p. 52
θ	Disreputation threshold ($-1 < \theta < 0$)	p. 52

Symbol	**Meaning**	**First Defined**
S_r^b	Set of reputable sellers to buyer b	p. 52
S_{dr}^b	Set of disreputable sellers to buyer b	p. 52
$f^b(g,p,s)$	Buyer b's expected value of buying good g at price p from seller s	p. 53
$v^b(g,p,q)$	True value of good g to buyer b, with p and q being the price and quality of g, respectively	p. 53
α	Learning rate ($0 \leq \alpha \leq 1$)	p. 54
$\vartheta^b(g)$	Product value demanded by buyer b for good g	p. 54
μ	Cooperation factor ($\mu > 0$)	p. 54
ν	Non-cooperation factor ($\nu < 0$)	p. 55
Δv^b	$\Delta v^b = v_{max}^b - v_{min}^b$, with v_{max}^b and v_{min}^b being the maximum and minimum value of $v^b(g,p,q)$	p. 56
μ_{min}	Minimum value of μ	p. 56
λ	Penalty factor ($\lambda > 1$)	p. 56
$h^s(g,p,b)$	Seller s's expected profit of selling good g at price p to buyer b	p. 57
$c^s(g,b)$	Cost of seller s to produce good g for buyer b	p. 57
$\phi^s(g,p,b)$	Actual profit of seller s if it sells good g at price p to buyer b	p. 57
Inc	Quality increasing factor	p. 58
Dec	Quality decreasing factor	p. 58

Table B.1: Glossary of the mathematical symbols used in the description of the proposed algorithms.

Bibliography

[1] M. R. Andersson and T. W. Sandholm. Leveled Commitment Contracts with Myopic and Strategic Agents. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 38-45, 1998.

[2] Michigan AuctionBot. http://auction.eecs.umich.edu/

[3] BargainFinder. http://bf.cstar.ac.com/bf

[4] A. Blomqvist, P. Wonnacott, and R. Wonnacott. Microeconomics (Fourth Edition). McGraw-Hill Ryerson, 1994.

[5] J. Boyan, A. Greenwald, R. M. Kirby, and J. Reiter. Bidding Algorithms for Simultaneous Auctions. In *Papers from the IJCAI-01 Workshop on Economic Agents, Models, and Mechanisms*, pages 1-11, 2001.

[6] S. Breban and J. Vassileva. Using Inter-Agent Trust Relationships for Efficient Coalition Formation. In *Proceedings of the Fifteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 221-236, 2002.

[7] D. Carmel and S. Markovitch. Learning and Using Opponent Models in Adversary Search. Technical Report 9609, Technion, 1996.

[8] D. Carmel and S. Markovitch. Incorporating Opponent Models in Adversary Search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 120-125, 1996.

[9] A. Chavez and P. Maes. Kasbah: An Agent Marketplace for Buying and Selling Goods. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1996.

[10] A. Chavez, D. Dreilinger, R. Guttman, and P. Maes. A Real-Life Experiment in Creating an Agent Marketplace. In *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1997.

[11] S. Choi and J. Liu. A Dynamic Mechanism for Time-Constrained Trading. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 568-575, 2001.

[12] S. E. Conry, K Kuwabara, V. R. Lesser, and R. A. Mayer. Multistage Negotiation for Distributed Constraint Satisfaction. In *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1462-1477, 1991.

[13] R. Davis and R. G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. In *Artificial Intelligence*, 20(1): 63-109, 1983.

[14] R. B. Doorenbos, O. Etzioni, and D. Weld. A Scalable Comparison-Shopping Agent for the World Wide Web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39-48, 1997.

[15] eBay. http://www.eBay.com

[16] B. Esfandiari and S. Chandrasekharan. On How Agents Make Friends: Mechanisms for Trust Acquisition. In *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 27-34, 2001.

[17] T. Finin and B. Grosof, Cochairs. Artificial Intelligence for Electronic Commerce. *Proceedings of the Sixteenth National Conference on Artificial Intelligence Workshop on Artificial Intelligence for Electronic Commerce*, 1999.

[18] T. Finin and B. Grosof, Cochairs. Knowledge-Based Electronic Markets. *Proceedings of the Seventeenth National Conference on Artificial Intelligence Workshop on Knowledge-Based Electronic Markets*, 2000.

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

[20] S. Gjerstad and J. Dickhaut. Price Formation in Double Auction. In *Games and Economic Behaviour*, 22:1-29, 1998.

[21] A. Greenwald and J. Kephart. Shopbots and Pricebots. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Vol. 1, pages 506-511, 1999.

[22] A. Greenwald, J. Kephart, and G. Tesauro. Strategic Pricebot Dynamics. In *Proceedings of the First ACM Conference on E-Commerce*, pages 58-67, 1999.

[23] A. Greenwald and J. Kephart. Probabilistic Pricebots. In *Proceedings of*

the Fifth International Conference on Autonomous Agents, pages 560-567, 2001.

[24] M. He and H. Leung. An Agent Bidding Strategy Based on Fuzzy Logic in a Continuous Double Auction. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 61-62, 2001.

[25] H. H. Hoos, and C. Boutilier. Solving Combinatorial Auctions Using Stochastic Local Search. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 22-29, 2000.

[26] J. Hu, D. Reeves, and H. S. Wong. Agent Service for Online Auctions. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence Workshop on Artificial Intelligence for Electronic Commerce*, pages 81-86, 1999.

[27] M. N. Huhns and M. P. Singh, Editors. Readings in Agents. Morgan Kaufmann Publishers, Inc., 1998.

[28] IEEE Computer Society. Proceedings of the Fourth International Conference on Multi-Agent Systems. Published by the IEEE Computer Society, 2000.

[29] Jango. http://www.jango.com/

[30] C. Jonker and J. Treur. Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *Proceedings of the Fourth International Conference on Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 81-94, 1999.

[31] K. L. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement Learning: A Survey. In *Journal of AI Research*, 4:237-285, 1996.

[32] R. Kakehi and M. Tokoro. A Negotiation Protocol for Conflict Resolution in Multi-Agent Environments. In *Proceedings of ICICIS*, pages 185-196, 1993.

[33] P. Kandzia and Klusch, Editors. Cooperative Information Agents I. Lecture Notes in Computer Science, Vol. 1202. Springer-Verlag, Berlin, 1997.

[34] J. O. Kephart. Economic Incentives for Information Agents. In *M. Klusch and L. Kerschberg, Editors, Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence*, Vol. 1860, pages 72-82. Springer-Verlag, Berlin, 2000.

[35] M. Klusch and G. Weiss, Editors. Cooperative Information II. Lecture Notes in Computer Science, Vol. 1435. Springer-Verlag, Berlin, 1998.

[36] M. Klusch and G. W. Onn Shehory, Editors. Cooperative Information Agents III. Lecture Notes in Computer Science, Vol. 1652. Springer-Verlag, Berlin, 1999.

[37] M. Klusch, editor. Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet. Springer-Verlag, Berlin, 1999.

[38] S. Kraus. Agents Contracting Tasks in Non-Collaborative Environments. In *Proceedings of the National Conference on Artificial Intelligence*, pages 243-248, 1993.

[39] S. E. Lander and V. R. Lesser. Negotiated Search: Organizing Cooperative Search among Heterogeneous Expert Agents. 1992.

[40] K. C. Laudon and J. P. Laudon. Management Information Systems (Sixth Edition). Prentice Hall, 2000.

[41] M. L. Littman. Markov Games as Framework for Multi-Agent Reinforcement Learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157-163, 1994.

[42] S. Marsh. Formalizing Trust as a Computational Concept. Ph.D. Thesis. Department of Computing Science and Mathematics, University of Stirling (Scotland), 1994.

[43] S. Minut and S. Mahadevan. A Reinforcement Learning Model of Selective Visual Attention. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 457-464, 2001.

[44] C. Mudgal and J. Vassileva. Bilateral Negotiation with Incomplete and Uncertain Information: A Decision Theoretic Approach Using a Model of the Opponent. In *M. Klusch and L. Kerschberg, Editors, Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence*, Vol. 1860, pages 107-118. Springer-Verlag, Berlin, 2000.

[45] Y. Nagayuki, S. Ishii, and K. Doya. Multi-Agent Reinforcement Learning: An Approach Based on the Other Agent's Internal Model. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 215-221, 2000.

[46] M. V. Nagendra Prasad, V. R. Lesser, and S. E. Lander. Learning Organizational Roles in a Heterogeneous Multi-Agent System. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 291-298, 1996.

[47] Nextag. http://www.Nextag.com

[48] T. Ohko, K. Hiraki, and Y. Anzai. Addressee Learning and Message Interception for Communication Load Reduction in Multiple Robot Environments. In *G. Weiss, Editor, Distributed Artificial Intelligence Meets Machine Learning, Lecture Notes in Artificial Intelligence*, Vol. 1221, pages 242-258. Springer-Verlag, Berlin, 1997.

[49] N. Ono and K. Fukumoto. Multi-Agent Reinforcement Learning: A Modular Approach. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 252-258, 1996.

[50] C. Preist, A. Byde, and C. Bartolini. Economic Dynamics of Agents in Multiple Auctions. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 545-551, 2001.

[51] H. Raiffa. The Art and Science of Negotiation. Harvard University Press, 1982.

[52] J. Rosenschein and G. Zlotkin. Rules of Encounter. The MIT Press, 1994.

[53] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally managable combinational auctions. In *Management Science*, 44(8):1131-1147, 1998.

[54] S. Russell and P. Norvig. Aritificial Intelligence: A Mordern Approach. Prentice Hall, 1995.

[55] J. Sabater and C. Sierra. REGRET: A Reputation Model for Gregarious Societies. In *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 61-69, 2001.

[56] T. W. Sandholm. An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 256-262, 1993.

[57] T. W. Sandholm and R. H. Crites. Multi-Agent Reinforcement in the Iterated Prisoner's Dilemma. In *Biosystems*, 37: 147-166, 1995.

[58] T. W. Sandholm and V. JR. Lesser. Advantages of a Leveled Commitment Contracting Protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 126-133, 1996.

[59] T. W. Sandholm. An Algorithm for Optimal Winner Determination in Combinatorial Auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 542-547, 1999.

[60] T. W. Sandholm. eMediator: A Next Generation Electronic Commerce Server. In *Papers from the AAAI Workshop on Artificial Intelligence for Electronic Commerce*, pages 46-55, 1999.

[61] T. W. Sandholm, and S. Suri. Improved Algorithms for Optimal Winner Determination in Combinatorial Auctions and Generalizations. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 90-97, 2000.

[62] S. Sen, M. Sekaran, and J. Hale. Learning to Coordinate without Sharing Information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426-431, 1994.

[63] R. G. Smith. The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. In *IEEE Transactions on Computers*, C-29(12): 1104-1113, 1980.

[64] H. Sung and S. T. Yuan. A Learning-Enabled Integrative Trust Model for E-Markets. In *Proceedings of the Fifth International Conference on Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 81-96, 2001.

[65] R. Sutton, and A. Barto. Reinforcement Learning: An Introduction. The MIT Press, 1998.

[66] M. Tan. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330-337, 1993.

[67] Thomas Tran and Robin Cohen. Hybrid Recommender Systems for Electronic Commerce. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence Workshop on Knowledge-Based Electronic Markets*, pages 78-84, 2000.

[68] Thomas Tran and Robin Cohen. A Learning Strategy for Economically-Motivated Agents in Market Environments. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence Workshop on Knowledge Discovery from Distributed, Dynamic, Heterogeneous, Autonomous Data and Knowledge Sources*, pages 51-56, 2001.

[69] Thomas Tran and Robin Cohen. A Reputation-Oriented Reinforcement Learning Strategy for Agents in Electronic Marketplaces. In *Computational Intelligence Journal, Special Issue on Agent Technology for Electronic Commerce*, Vol. 18, No. 4, pages 550-565, 2002.

[70] Thomas Tran and Robin Cohen. A Reputation-Oriented Reinforcement Learning Approach for Agents in Electronic Marketplaces. In *Proceedings*

of the Eighteenth National Conference on Artificial Intelligence (Doctoral Consortium), page 989, 2002.

[71] Thomas Tran and Robin Cohen. A Learning Algorithm for Agents in Electronic Marketplaces. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence Workshop on Multi-Agent Modelling and Simulation of Economic Systems*, pages 46-53, 2002.

[72] Thomas Tran and Robin Cohen. A Learning Algorithm for Buying and Selling Agents in Electronic Marketplaces. In *Proceedings of the Fifteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 31-43, 2002.

[73] Thomas Tran and Robin Cohen. Learning Algorithms for Software Agents in Uncertain and Untrusted Market Environments. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.

[74] Thomas Tran and Robin Cohen. Modelling Reputation in Agent-Based Marketplaces to Improve The Performance of Buying Agents. In *Proceedings of the Ninth International Conference on User Modelling*, pages 273-282, 2003. (One of the nominees for Best Student Paper Award)

[75] Thomas Tran and Robin Cohen. A Strategy for Improved Satisfaction of Selling Software Agents in E-Commerce. In *Proceedings of the Sixteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 434-446, 2003.

[76] J. Vassileva, S. Breban, and M. Horsch. Agent Reasoning Mechanism for Long-Term Coalitions Based on Decision Making and Trust. In *Computa-*

tional Intelligence Journal, Special Issue on Agent Technology for Electronic Commerce, Vol. 18, No. 4, pages 583-595, 2002.

[77] J. M. Vidal and E. H. Durfee. The Impact of Nested Agent Models in an Information Economy. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 377-384, 1996.

[78] J. M. Vidal. Computational Agents That Learn About Agents: Algorithms for Their Design and A Predictive Theory of Their Behavior. PhD Thesis, Department of Computer Science & Engineering, University of Michigan, 1998.

[79] G. Weiss. Learning to Coordinate Actions in Multi-Agent Systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 311-316, 1993.

[80] G. Weiss, editor. Distributed Artificial Intelligence Meets Machine Learning. *Lecture Notes in Artificial Intelligence*, Vol. 1221. Springer-Verlag, Berlin, 1997.

[81] G. Weiss, editor. Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 2000.

[82] M. Wooldridge and N. Jennings. Intelligent Agents: Theory and Practice. In *the Knowledge Engineering Review*, 10(2): 115-152, 1995.

[83] P. R. Wurman, M. P. Wellman, and W. E. Wash. The Michigan Internet AuctionBot: A Configurable Auction Server for Humans and Software Agents. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 301-308, 1998.

[84] B. Yu and M. P. Singh. A Social Mechanism of Reputation Management in Electronic Communities. In M. Klusch and L. Kerschberg, Editors, *Cooperative Information Agents IV*, Lecture Notes in Artificial Intelligence, Vol. 1860, pages 154-165. Springer-Verlag, Berlin, 2000.

[85] G. Zacharia and P. Maes. Trust Management through Reputation Mechanisms. In *Applied Artificial Intelligence*, 14: 881-907, 2000.