e-Marketplace Development and Trading Agent Design for Supply Chain Management

Kanghua Zhao

A thesis submitted for the degree of Master of Science at The University of Western Sydney

January 2006
© Kanghua Zhao

Typeset in Palatino by \TeX\ and \LaTeX\ 2ε.
I declare that, to the best of my knowledge, the work presented in this thesis is original and my own work, except where otherwise indicated, and that no part of this thesis has been submitted as part of any other degree at this or any other university. Permission is herewith granted to the University of Western Sydney to circulate and to reproduce the above title for non-commercial purposes, at its discretion, upon request of individuals or institutions.

Kanghua Zhao

6 January 2006
To my wife Jian.
Acknowledgements

I would like to thank my supervisor, Dr. Dongmo Zhang. Without his enthusiastic support, tender guidance and sincere encouragement, this work would not have been possible. Most of the theories, approaches and results included in this thesis were spawned from our discussions and joint work. Some of the results in this work have been presented in our joint research papers published at international conferences and in international journals (see Appendix A). Dr. Zhang is the leader of the “Jackaroo Trading Agent Design and Development” team and is the main contributor to the “Jackaroo” software.

I would like to thank Associate Professor Yan Zhang, the director of the Intelligent Systems Laboratory (ISL), for encouraging me to pursue scientific research and giving me enormous support throughout my research study.

I also would like to thank the rest of my research colleagues in the ISL for creating an educational, enjoyable, and challenging atmosphere. Meetings with members of the “Jackaroo” team led to the development of many new ideas presented in this thesis. I would especially like to thank Chia-Ming Liang, who implemented a substantial portion of our software agent and Golenur Begum, who put great effort into data analysis. Without their hard work, the experimental results presented in this thesis would not have
been possible. I would also like to thank Vino Fernando Crescini, Yulin Ding and Sai Kiran Lakkaraju. Their friendship over the last one and half years has been invaluable, both socially and intellectually. I especially appreciate their willingness to listen to my problems and to help me develop, refine, and clarify solutions to these problems.

I am especially grateful for the support of my wife Jian Huang, my parents, my sisters and my brothers-in-law. I cannot express how important their dedication and love have been throughout my life.

Many of the works presented in this thesis are the result of intensive discussions and joint efforts between myself, Dr. Dongmo Zhang and other team members. For this reason it is better to use the pronoun “we” instead of “I” to present some ideas, especially when it is difficult to identify who are the initial proposers of those ideas.

This work was financially supported through a thesis completion scholarship from UWS and a research grant provided by Dr. Dongmo Zhang.
Abstract

Supply Chain Management (SCM) deals with the planning and coordination of resource procuring, product marketing, production scheduling, and end-product delivering across multiple organizations. With the emergence of e-Marketplaces, it is natural to seek automated solutions that are capable of responsively coping with dynamic market conditions and rapidly evaluating/re-evaluating a large number of bidding and procurement options in real time while taking into account strategies employed by market competitors. It is therefore necessary to use automated programs or software trading agents to perform and support SCM.

This thesis considers the problems posed by e-Marketplace development and the design of intelligent trading agents for supply chain management in e-Marketplaces. These problems are interesting because good solutions are thought to require a synthesis of several fields: information technology, economics, industrial operations research, organization management, game theory, artificial intelligence, machine learning, multi-agent systems and statistics.

The primary contribution of this thesis is its detailed analysis of the characteristics and properties of a typical e-Marketplace environment, its exploration of optimal trading agent strategies and its discussions of possible enhance-
ments for the state-of-the-art e-Marketplace development and trading agent design.

This research is stimulated by the recently introduced Trading Agent Competition for Supply Chain Management (TAC SCM) game scenario, which is a representation of typical e-Marketplaces. To the best of our knowledge, TAC SCM is the only competition-based e-Marketplace simulation system that captures the real challenges present in an integrated procurement, production and customer-bidding environment. The game was designed jointly by a team made up of researchers from the e-Supply Chain Management Lab at Carnegie Mellon University, the University of Minnesota, and the Swedish Institute of Computer Science (SICS), with tremendous inputs from the research community. To be successful, an agent needs to perform the following interrelated tasks on each simulated time unit, taking into account its limited production capacity:

1. Bid for customer orders
2. Negotiate supply contracts for components
3. Schedule daily assembly activities and deliveries
4. Manage costs and storage

This thesis starts with a detailed analysis of TAC SCM e-Marketplaces. We present an abstract model of the game, which consists of the market demand
and market supply models. Based on these models and related theoretical results, we then describe the architecture and the main strategies employed by our very own TAC SCM agent, the “Jackaroo” trading agent. A series of experimental results that indicate the correctness and effectiveness of the strategies will then be presented, followed by the presentation of a new negotiation mechanism for the TAC SCM component market and the presentation of a new supplier model to demonstrate how negotiation strategies can be applied in automated negotiation. Then, we will explore the areas where the current TAC SCM scenario can be improved, followed by an outline of proposed enhancements to the TAC SCM e-Marketplace mechanism. Finally the thesis concludes with a discussion of future research directions.
# Contents

Acknowledgements vii

Abstract ix

1 Introduction 1

1.1 Background and Motivation ............................... 2

1.1.1 e-Commerce and e-Marketplace ...................... 2

1.1.2 Supply Chain Management Challenges in e-Marketplaces 5

1.1.3 Agent-Enabled SCM Automation in e-Marketplaces ... 8

1.1.4 TAC SCM ............................................ 12

1.2 "Jackaroo" TAC SCM Team and The Author’s Contributions . 16

1.3 Overview of this Thesis .................................. 17

2 e-Market Modelling: Economic Models for TAC SCM Markets 19

2.1 Introduction ............................................. 19

2.2 Product Market Models: Quantity Competition ............ 22

2.2.1 Basic Assumptions .................................. 23

2.2.2 Nash Equilibrium .................................... 25

2.2.3 Varying Customer Demands ......................... 29

2.3 Product Market Models: Price Competition ............... 31

xiii
2.4 Component Market Model ........................................... 34
2.5 Discussion and Related Work ................................. 36
2.6 Summary ....................................................... 37

3 TAC SCM Trading Agent Design and Analysis .......... 39
3.1 Introduction ..................................................... 39
3.2 Agent Architecture ............................................. 41
3.3 Strategizing “Jackaroo” ......................................... 44
   3.3.1 Component Procuring .................................... 44
   3.3.2 Product Pricing ............................................ 47
   3.3.3 Price Coefficient Learning ............................... 48
   3.3.4 Production Scheduling .................................... 51
   3.3.5 Market Price Forecasting ................................. 53
      3.3.5.1 Correlation between Market Supply and Market-
               Clearing Price ........................................... 53
      3.3.5.2 Smooth-Regression for Price Forecasting .......... 55
      3.3.5.3 Estimation of Price Parameters .................... 60
3.4 Discussion and Related Work ................................. 62
3.5 Summary ....................................................... 64

4 Negotiation Mechanism for the TAC SCM Component Market 65
4.1 Introduction ..................................................... 65
4.2 Negotiation Protocol for the Component Market .......... 68
## Contents

4.3 Supplier Model ........................................... 69  
4.3.1 Assumptions ......................................... 71  
4.3.2 Daily Production Capacity ............................. 72  
4.3.3 Guaranteed Production Capacity(GPC) ................. 74  
4.4 Production Scheduling ................................. 76  
4.5 Supplier Pricing Policy ................................. 78  
4.5.1 Making Offers ......................................... 78  
4.5.2 Counter Offer Processing ............................. 80  
4.6 Properties of the Suppliers Model ......................... 82  
4.6.1 Orders vs. Counter Offers ............................. 82  
4.6.2 Large Procurement .................................. 84  
4.6.3 Day 0 Procurement .................................. 85  
4.6.4 Bargaining Range .................................... 88  
4.7 Summary .................................................. 90  

5 Enhancements of the TAC SCM Game ..................... 91  
5.1 Introduction ............................................... 91  
5.2 A Customer Behavior Model with Linear Reserve Price Function ........................................... 91  
5.3 Extendible Production Capacity of Manufacturing Agents ........................................... 94  
5.4 Agent Cooperation via an Information Sharing Mechanism ........................................... 96  
5.4.1 An Information Sharing Mechanism .................. 97  
5.4.2 Impact of Sharing Private Inventory Information ........................................... 99
## Contents

5.4.2.1 The Model .............................................. 100  
5.4.2.2 Best Response Function ............................... 102  
5.4.2.3 Supply Chain Configurations ......................... 105  
5.4.2.4 The Simulation Experiment and Results .......... 106  

5.5 Discussion and Related Work ............................... 108  
5.6 Summary ..................................................... 109  

6 Conclusion and Future Work ................................... 111  

A Publications .................................................... 115  

Bibliography ....................................................... 117
List of Tables

3.1 Correlation between market-clearing price and market supply in the product market. ........................................... 55
3.2 Market price forecasting by using linear regression ........ 56
3.3 Market price forecasting by using smooth regression. ....... 60
3.4 Market price forecasting by using smooth regression (continued). ................................................................. 61
5.1 Domain policies used in the simulation experiment ........ 106
5.2 Sharing vs. concealing private inventory information ....... 107
List of Figures

1.1 TAC SCM scenario ........................................ 14

2.1 TAC SCM component market and product market ................. 22

2.2 Equilibrium under large market demands ......................... 30

2.3 Equilibrium under small market demands ......................... 31

3.1 Architecture of “Jackaroo” .................................. 42

3.2 TAC SCM decision problem ................................... 43

3.3 An example of component procurement scheduling (data from
TAC SCM-03 game 418 on server tac6.sic.se) .................... 45

3.4 Correlation between market price and market supply of PC prod-
     ucts (data from TAC SCM-03 game 427 on server tac5.sics.se) . 49

3.5 Fitting curve of $b_0$ over market demand ..................... 58

3.6 Fitting curve of $b_1$ over market demand ..................... 59

4.1 Negotiation protocol between agent and supplier ............... 70

4.2 Agent’s bargaining range for component price (Pintel CPU 2.0GHz) 89

4.3 Agent’s bargaining range for component due date (Pintel CPU 2.0GHz) .... 89
5.1 Achieve extendable production capacity by adding Manufacturing Equipment Supplier (MES) ................. 95
5.2 An information sharing mechanism ................................. 98
Introduction

E-business has been embraced by many as the new frontier of the information revolution. A major step in e-Business innovation is the automation of business activities. Much of the economic value of e-commerce arises from this kind of automation. The boom of agent technology has resulted in a plethora of agent-based systems that automate business activities such as product brokering, merchant bordering, auction bidding and trade negotiation[33][51][57]. More ambitious attempts have been made by a number of companies seeking to create automated programs, called trading agents that are capable of doing business autonomously in real marketplaces including e-Marketplaces. Since only a very small fraction of these businesses exist today, it is obvious that there is still a lot to be learned before we can design an effective trading agent to manage highly dynamic market situations.

In this chapter, we will briefly present some background knowledge and our motivation for the studies in e-Marketplace development and the e-Trading agent design project.
1.1 Background and Motivation

1.1.1 e-Commerce and e-Marketplace

A market is an institution or mechanism that brings together suppliers (sellers) and demanders (buyers) of particular goods and services[9]. Normally neither supply nor demand by itself can determine the price or the quantity of a good or service in a competitive market. Supply and demand are like two wings of the same bird - both are required to make the system work and each depends on the other. Under increasing pressure from a changing global market, traditional relationships in market supply and demand have inevitably gone through a number of radical alterations. Of these alterations the most prominent one is the emergence and proliferation of electronic commerce (“e-Commerce”). The term electronic commerce generally denotes an advanced step of modern commerce in which the figures of buyer and seller are replaced by electronic entities[46].

One driving force behind this trend towards e-Commerce is the development of information technology (IT). Whilst consistently producing performance enhancements, IT has also brought about a corresponding rapid decline in costs. This, combined with the continuing expansion of global communications networks (especially the Internet), has allowed the major players of markets to create extensive and sustainable competitive advantage[58]. The advancement of the IT and communication networks also provided the frame-
work upon which electronic commerce could subsequently develop. In the past few years alone, we have witnessed a surge of business-to-consumer (B2C) and business-to-business (B2B) electronic commerce operated on the Internet. According to the United States Bureau of Census, retail e-commerce sales increased from 15 billion dollars in 1999 to 44 billion dollars in 2003. Studies have shown that e-commerce has become very significant in certain product categories. For example, in 2003, 32 percent of computer hardware and software sales transactions were performed online, 17 percent of ticket sales were done over the Internet, and 12 percent of book sales were completed electronically, according to a study by the National Retail Federation and Forecaster Research[49].

From a computer science point of view, an e-Marketplace can be described as a virtual online market place where customers, retailers, wholesalers, distributors, manufactures and suppliers conduct trade, find and exchange information and collaborate with each other via an aggregation of information portals, trading exchanges and collaboration mechanisms. e-Trading Mechanisms then deal with the activities of marketing and procuring commodities between two or more parties through electronic transaction systems in e-Marketplaces. Transactions can be Business-to-Consumer or Business-to-Business. Besides the trading entities involved, an electronic commerce framework generally consists of one or more e-Marketplaces, each with a set of e-Trading mechanisms.

There are many advantages in using e-Marketplaces for international trade
as a buyer or seller[27]:

- Trading in e-Marketplaces can provide market transparency since the availability, prices, stocks, delivery times and payment terms of products can be shown in the e-Marketplace. Companies are able to make business decisions based on more information, provided that sufficient buyers and suppliers are involved in the e-Marketplace.

- The use of e-Marketplaces may simplify international business, as e-Marketplaces make it easier to effectively buy and sell products locally and internationally. e-Marketplaces make it easier for companies to reach their target groups and decrease marketing costs. The Internet can easily be used as a marketing channel in order to disseminate product and company information to a large number of potential buyers.

- E-marketplaces can remove time and space limitations, as trade is possible 24 hours a day, 7 days a week. Trading partners do not have to travel, as they have the ability to negotiate from their own office, which means considerable time savings.

As e-Commerce transactions have grown in number and value, a number of unresolved issues have unfolded. Among these issues are the development of healthy e-Marketplaces with robust e-Trading mechanisms and the exploration of automated supply chain management (SCM) in e-Marketplaces - two major concerns of this thesis.
1.1.2 Supply Chain Management Challenges in e-Marketplaces

An e-Marketplace provides the base from which companies are able to coordinate their everyday transactions with their business partners, with the end result that these increasingly effective and efficient e-Trading processes create the need to constantly redefine traditional supply chains[58]. The traditional view of supply chains as only physical distribution is too limited. The supply chain in an e-Marketplace is more than the physical movement of goods. It may also be the exchange of information and money, and the creation and deployment of intellectual capital. Ayers[6] defined a supply chain as follows:

“A Supply Chain consists of the life cycle processes supporting physical, information and financial and knowledge flows for moving products and services from suppliers and end-users.”

Traditionally, physical, information and financial flows are frequently cited dimensions of the supply chain. Less frequently mentioned is the role of knowledge inputs into supply-chain processes. Knowledge is as important as, if not more than, physical and other types of inputs[6]. A good example is new product development; this supply chain process requires close coordination of intellectual input with physical inputs.

Profitability of organizations in manufacturing, wholesaling and retailing industries depends on the efficiency and effectiveness of their management of supply chains. Supply chain management is vital to today’s global economy,
leading to trillions of dollars in annual transactions worldwide. Understanding the role of supply chain management is increasingly important to those involved in e-Marketplaces.

Following Ayers’ definition, the Supply Chain Management (SCM) deals with the design, maintenance and operation of supply chain processes for satisfaction of end user needs.[6]. This definition reflects the idea that SCM extends to both the supply chain formulation and its subsequent maintenance and operation[6]. More specifically, supply chain management deals with the planning and coordination of bidding/marketing, production/delivery scheduling and procurement activities across the multiple organizations involved in the delivery of one or more products or services to end consumers.

For years, it was thought that it was enough for manufacturers to have a Material Requirement Planning (MRP) or Enterprise Resource Planning (ERP) system that could help answer fundamental SCM questions such as: What are we going to make? What do we need to make the products? What do we have now? What materials do we need, and when do we need them? What resources/capacity do we need and when do we need them? However, most systems are not capable of handling all the variables up and down the supply chain. The complexities of getting material ordered, manufactured and delivered overload most supply chain management systems. A recent study[10] has found that companies lose between 9% and 20% of their value over a six-month period due to supply chain problems. These problems include part shortages, excessive fin-
ished good inventories, under-utilized plant capacity, unnecessary warehousing costs, and inefficient transportation of supplies and finished goods. Because supply chains involve independent suppliers and manufacturers who must maintain the integrity and confidentiality of their information systems and operations for business advantages, these problems are exacerbated.

There are a number of fundamental weaknesses in the old SCM logic. Many planning and scheduling systems assume that lead times are fixed, queues must exist and are static, capacity is infinite and backward scheduling logic will produce valid load profiles and good shop floor schedules. These assumptions are totally illogical, and following them causes many schedule compliance problems[25]. Organizations in e-Marketplaces need to develop flexible supply chain processes that can responsively adapt to the needs of various customer segments. They must also develop supply chain strategy, processes and supporting systems that conform to current and future requirements.

In conclusion, SCM is a discipline worthy of distinct identity. This identity puts it on the level with disciplines like finance, operations or marketing[6]. SCM creates new challenges for managers, especially those involved in e-Marketplaces. Old missions must be achieved in new ways. Generally, an effective SCM approach must focus on:

- Flexible bidding and production processes that can respond quickly to changing customer demand and market supply
• A short-cycle, demand-driven and build-to-order production process

• Accurate, relevant information that is available on demand throughout the supply chain

1.1.3 Agent-Enabled SCM Automation in e-Marketplaces

Supply chains are a highly dynamic, stochastic, and strategic environment. To operate in such an environment requires coping with dynamic market conditions by constantly evaluating and re-evaluating a large number of options in real time. It is important to take into account changing market conditions and also strategies employed by competitors. With the emergence and development of electronic marketplaces, it is only natural to seek automated SCM solutions that are capable of rapidly evaluating/re-evaluating a large number of bidding, production, delivery and procurement options[4].

Consider a typical situation faced by a manufacturer in a supply chain: A stream of customer RFQs (Request-for-Quotes) regarding various finished products, and offers by potential component suppliers promising components for the production of the finished products, arrive in a not-so-predictable fashion. Determining the combination of customer requests and which supplier offers to accept, so as to maximize profit is a challenging problem. On the one hand, if one accepts too many requests (orders) there could be a risk of defaulting on some orders, as key components may be unavailable or the production
capacity to manufacture them may simply be inadequate. On the other hand, accepting too few customer requests (orders) may result in missing out on potentially profitable deals. In addition, erroneous forecasts of supply/demand or the late arrival of components may mean re-evaluating and generating fresh plans[4]. Therefore, an automated supply chain management system is required throughout the supply chain to accurately and quickly answer the following critical questions:

- How many orders should be sought or received?
- When should an order be completed?
- If an order has been delayed, what are the reasons for the delay?
- What are the specific problems that delayed the production schedule?
- What are the best production/delivery schedules that can be executed?

It seems that the best approach to automating supply chain management is to gather companies into e-Marketplaces, where they can negotiate for goods and services. However, such centralization does not necessarily foster collaborations, alliances, and long-term relationships, which are the more significant drivers of improved efficiency in supply chains. A distributed architecture is thus preferable, but computer applications that can automate supply chains require a number of important properties beyond traditional software approaches[35], as follows:
• **Disintermediation** (the direct association between users and their software). Providing a participant with seamless access to and interaction with remote information, application, and human resources requires a distributed, active-object architecture.

• **Dynamic composability and execution.** A system should execute as a set of distributed parts, but the resources required will be mostly unknown until run-time. This requires an infrastructure to enable their discovery and composition as needed.

• **Interaction.** There might be subtle and critical patterns of interaction among supply-chain participants, but the specific interactions may be unknown until run-time, and may vary: this requires that the patterns of interaction to be explicitly represented and reasoned about.

• **Error tolerance and exploitation.** As deployed systems become increasingly complex, they should anticipate and compensate for errors in their components and interaction protocols.

Recent advances in *software agent* architecture and languages can address the above requirements. A software agent is considered as an entity with goals, capable of actions endowed with domain knowledge and situated in an environment[59].

Software agents differ from traditional software because they exhibit three distinct traits[22]: 
• the ability to personalize

• the ability to be proactive

• the ability to be adaptive

Personalization is the capturing of the unique goals of the user and acting accordingly on his/her behalf. Personalization is apparent for example in a user’s ability to customize informational searches. A software agent is proactive if it has the ability to complete its task independent of the user’s activities. An agent that finds new information for the user, without the user requesting a response is an example of a proactive agent. The ability to be adaptive is described as the software agent’s ability to learn from its past actions and from its environment. Learning from past actions is characteristic of the artificial intelligence (AI) technology which preceded much of today’s agent technology[22].

Software agents are well suited to dynamic supply chain problems in which frequent and timely adjustments to the flow of goods and production schedules are needed in order to leverage conditions in the marketplace or to take advantage of opportunities as they arise.

Agent-based technologies are identified as new software-development approaches with a potential significant impact on supply chain management solutions. Multi-agent systems (MAS) are suitable for the domains that involve interactions between different people or organizations with different (possibly
conflicting) goals and proprietary information. The MAS paradigm is a natural fit to certain classes of dynamic supply chain problems because the paradigm focuses on coordinating the activities of loosely coupled distributed entities, e.g., raw material suppliers, shippers, manufacturers, distribution centers, and retailers (where each of these is represented by an agent). One goal of the paradigm is to enable agents to meet deadlines and resource constraints but also to be flexible, robust, responsive, and adaptive. With agents, these behaviors are obtained without centralization and without assuming complete static knowledge a priori[59].

We will see an example of MAS-based e-Marketplace simulation with a focus on supply chain management in the following section.

1.1.4 TAC SCM

Much of the past work in the area of MAS-based e-Marketplace and supply chain simulation has focused on subsets of the problem. For instance, looking at only the production aspects[21][76] or only procurement decisions whilst ignoring customer demand[32][44]. Furthermore, key temporal and capacity constraints have often been ignored or simplified[7][8][67]. To the best of our knowledge, TAC SCM is the only competition-based e-Marketplace simulation system that captures the real challenges of an integrated procurement, production and customer-bidding environment, while keeping the rules of the game simple enough to entice a large number of competitors to submit entries.
TAC (Trading Agent Competition) is a virtual arena where software agents can compete in different realistic scenarios. TAC has been successfully run for four years since it was introduced by Wellman and Wurman[71]. This annual activity offers “an international forum designed to promote and encourage high quality research into the trading agent problem”\(^1\).

The first three games proceeded with a travel agency scenario (TAC Classic). In 2003, a new game scenario of Supply Chain Management (SCM) was introduced by CMU and SICS[3]. This scenario specifies a supply chain integration of a Personal Computer (PC) marketplace. Here, the agents\(^2\) run virtual companies assembling and selling PCs. They interact with different markets, ordering the components needed to build PCs and selling the assembled product to customers. The TAC SCM game provides a competitive environment to stimulate solutions to the problems involved in supply chain integration and multiple market e-Trading. Participants are required to design a trading agent capable of procuring components, manufacturing PCs and competing for customer orders, while trying to maintain a high profit margin. About thirty teams from different universities and research institutes around the world were attracted to the challenge and competed against each other in 2003.

---

\(^1\)See http://www.sics.se/tac for more details.

\(^2\)In TAC SCM game, suppliers and customers are implemented as built-in agents on the server side. Therefore, the word “agents” normally refers to manufacturing agents designed by individual competitors unless otherwise indicated.
Specifically, TAC SCM depicts a scenario (Figure 1.1) where six PC (personal computer) assembly agents, limited by the capacity of their assembly lines, compete for customer orders and for procurement of components from eight potential suppliers, over a period of 220 simulated days. Each agent can manufacture 16 different types of PC products, characterized by different stock keeping units (SKUs). Components are acquired from a common pool of suppliers at costs that vary as a function of demand. After assembly, each agent can sell its PCs to a common pool of customers by outbidding the other agents with the lowest selling price[3].

Daily customer demand comes in the form of requests for quotes (RFQs). Each request contains a SKU, a due day, a penalty rate, and a reserve price - the highest price the customer is willing to pay. Each day, a customer issues
RFQs and receives offers from agents. The customer then selects the agent with the lowest-priced (and valid) offer and awards that agent with an order.

Meanwhile, the agents themselves are sending RFQs to suppliers, requesting a specific quantity of a component to arrive on a particular day.

At the end of each day, each agent converts the components it acquired from suppliers into SKUs according to a production schedule it generates for its factory. It also reports a delivery schedule assigning the SKUs in its inventory to customer orders.

In other words, on each simulated day, each agent performs the following highly inter-related tasks, taking into account its firmly restricted production capacity:

- Choose RFQs and Bid for customer orders
- Negotiate supply contracts for components
- Schedule daily assembly activities and deliveries

At the end of the game, agents are ranked by their total profits.

In conclusion, TAC SCM game simulates a 3-tier dynamic supply chain environment in which software trading agents tackle complex problems with highly imperfect information, and high-dimensional strategy spaces[28]. The games, or simulations, are of significant importance for many reasons: They constitute a controlled environment in which one can study complex economical processes and even evaluate the validity of different economical models.
They further supply computer scientists, especially those in the field of autonomous agents, with a framework for empirical evaluation of algorithms and theories.[61].

A more detailed description of the game can be found at TAC Website:

http://www.sics.se/tac.

1.2 "Jackaroo" TAC SCM Team and The Author’s Contributions

TAC SCM has attracted more than 30 research groups from universities and institutes all over the world. As one of the participants in the TAC SCM, the “Jackaroo” team (led by Dr. Dongmo Zhang), representing the University of Western Sydney (UWS), contributed a trading agent to the annual competitions. In the TAC SCM-03 tournament, the “Jackaroo” agent finished in third place in the qualifying round and fourth place in the seeding round. Unfortunately, the agent was unable to proceed in the final round due to a network problem at the conference venue. In the TAC SCM-04 tournament, the “Jackaroo” agent finished first place in the qualifying round, fifth place in the seeding round, first place in the quarter-final round and fifth place in the semi-final round. Although the “Jackaroo” agent was eliminated before the final round, it nevertheless was recognized by the research community as one of the most inventive and influential agents because its demonstration of several leading
as a core member of “Jackaroo” team, my contribution includes the design, development, testing and refinement of the pricing, bidding, procuring and information collection modules.

1.3 Overview of this Thesis

This thesis focuses on the analysis of the characteristics of e-Marketplaces, together with the discussion of optimal trading agent strategies and possible enhancements of the state-of-the-art e-Marketplace architecture. We chose the TAC SCM game scenario as a representation of typical e-Marketplaces and a test-bed to examine our theories and algorithms.

The rest of this thesis is structured as follows: In chapter 2 we introduce an abstract model of TAC SCM based on micro-economic theories. With the models and theoretical results, we describe in chapter 3 the key strategies that have been applied in the design and implementation of our “Jackaroo” agent, one of the most inventive and influential agents in the TAC SCM tournaments. In chapter 4, we present a new negotiation mechanism for TAC SCM component market and a new supplier model to demonstrate how negotiation strategies can be applied in automated negotiation. In chapter 5 we address some problems existing in current TAC SCM scenario and explore several enhancements to improve TAC SCM e-Market mechanism. Finally, we conclude this thesis.
with a discussion of future research directions.
Chapter 2

e-Market Modelling: Economic Models for TAC SCM Markets

2.1 Introduction

A typical supply chain normally consists of at least three components: *raw material suppliers, product manufacturers* and *end-product consumers*. These three components are linked together by two typical markets: *a raw material or component market* and *an end-product market*. As we have mentioned in the previous chapter, the TAC SCM game devises such a typical three-tier supply chain scenario. Besides, both the component market and the end-product market are dominated by a small\(^1\) number of sellers of a particular type of good. Such markets are called oligopolistic markets, and are more interesting than perfect competition markets, from a micro-economics point of view.

The key feature of any oligopolistic market is strategic interdependence. Strategic interdependence exists whenever the payoffs of a firm’s choices de-

\(^{1}\)The number of firms is regarded as “small” when the actions of any one firm have a significant impact on the market (or industry) as a whole.
epend on the choices of other firms. Interdependence arises from the existence of a small number of rival firms in the market[58]. Assumptions on whether firms use price or quantity as a decision variable leads to the two well known Cournot and Bertrand models, respectively[34][32].

A Cournot model is a generalization of the Cournot game in Game Theory to describe industry structure. Let’s consider an oligopoly market with \( N \) participants (firms), each of which chooses a quantity of \( q_i \) as its output. Price is a commonly-known decreasing function of total output. All firms know \( N \) and take the output of the others as given. Each firm has a cost function \( c_i(q_i) \). Usually the cost functions are treated as common knowledge. Often, the cost functions are assumed to be the same for all firms. The prediction of the model is that the firms will choose the Nash equilibrium output levels[38].

A Bertrand game is a model of a bidding war between firms, each of which can offer to sell a certain good (say, widgets). Each firm may choose a price to sell widgets, and must then supply as many widgets as demanded. Consumers are assumed to buy the cheaper one, or to purchase equal amount of widgets from each if the prices are the same. It is best for the firms (both collectively and individually) to cooperate, charge monopoly price, and split the profits. Each firm could then seize the whole market by lowering price slightly. However, the outcome of non-cooperative Nash equilibrium of a Bertrand game is that both agents charge a zero-profit price.

In this chapter, we will present an abstract economic model of the TAC
SCM markets to show how micro-economic theory can be applied to the analysis of e-Marketplaces.

To make the presentation simple, our analysis will be based on TAC SCM-03 specification. Some minor changes (e.g. storage cost, customer demand variation and supplier pricing formula) have been introduced in TAC SCM-04. These changes aimed to refine the game scenario and overcome some shortcomings of the game but achieved no noticeable improvement. A new specification for TAC SCM-05 is still under discussion and we hope our suggested enhancements of TAC SCM in chapter 5 can make some contributions to that. Nevertheless, those changes affect neither the nature of TAC SCM nor the generality of the models and results discussed in this chapter.

As a typical supply chain, the TAC SCM game simulates an economy with three types of entities (see Figure 2.1): component suppliers, PC manufacturers and end customers, who are linked with two marketplaces: component market and PC product market.

The downstream marketplace (i.e. product market) is characterized by the combination of quantity competition and price competition between manufacturers (agents). In 2.2, we will specify the quantity competition by a variation of the classic Cournot model and view price competition as an extension of the Bertrand game.

Different from the product market, the market of components is a typical constant-supply market. This is because the market supply is fixed with only
a small reverting random walk. Therefore, in 2.4, we will specify the upstream market with a constant-supply model.

As will be shown in chapter 3, the models and related results we obtained in this chapter will provide solutions to some critical decision-making problems in the TAC SCM trading agent design, including daily production, product pricing and component procuring.

### 2.2 Product Market Models: Quantity Competition

The TAC SCM product market is a typical oligopoly\footnote{A market for a product where a few major suppliers account for a large majority of sales.} where the manufacturers (agents) can choose quantities of products to supply within their production capacity. The market price of each product relies on the aggregate quantity
of all manufacturers. Since each manufacturer’s pay-off structure is common knowledge to each agent, the quantity competition of the market can be easily specified with a classic Cournot model with a slight variation\[48\]. To maximize the use of the traditional micro-economic theories, we temporarily assume that all PC products are homogeneous, with the same market price and the same production costs for each manufacturer (agent). In the later sections, we will relax some of these assumptions.

2.2.1 Basic Assumptions

In general, we assume that there are \( n \) manufacturers competing in the market. Let \( q_i \) denote the quantity of PCs produced by manufacturer \( i \) and \( Q \) the aggregate quantity of PCs in the market, that is, \( Q = \sum_{i=1}^{n} q_i \). Assume that market demand (expressed by customer’s request for quotes (RFQs)) for all products is \( D \). If \( Q \leq D \), we assume the market clearing-price of the product is constant at \( P_0 \). If \( Q > D \), the market clearing-price decreases with the level of products on the market until the price drops to 0. Let \( P(Q) \) denote the market clearing-price over \( Q \). Then,

\[
P(Q) = \begin{cases} 
P_0, & \text{if } Q \leq D; \\
\Gamma(Q), & \text{otherwise.} 
\end{cases}
\]  

(2.2.1)
where $\Gamma(Q)$ is a monotonous decreasing function, called the *price depreciation function*. Specially, if the price decline is linear, the price function can be further simplified:

$$
P(Q) = \begin{cases} 
P_0, & \text{if } Q \leq D; \\
P_0 - \gamma(Q - D), & \text{if } D < Q \leq D + \frac{P_0}{\gamma}; \\
0, & \text{otherwise.} \end{cases}
$$

(2.2.2)

where $\gamma$ is called the *depreciation coefficient* ($\gamma > 0$).

Let $\delta$ be the marginal cost of the product and $c_0$ be the fixed cost of each manufacturer. The cost for each manufacturer to produce $q$ products is then:

$$
C(q) = \delta q + c_0, \text{ where } \delta \text{ and } c_0 \text{ are non-negative.}
$$

Following the terminology of game theory, a *strategy*, $q_i$, of manufacturer $i$ here means the quantity that the manufacturer chooses to produce.

A *strategy profile*, $S$, is then a decision of production by all manufacturers: $(q_1, \cdots, q_n)$, where $q_i \geq 0$ for any $i$.

The profit of each manufacturer $i$ (payoff function) can then be written as:

$$
\pi_i(S) = q_i P(Q) - C(q_i) = q_i P(Q) - \delta q_i - c_0
$$

If we assume that the price depreciation function is linear, the payoff function can be further specified as:

$$
\pi_i(S) = \begin{cases} 
(P_0 - \delta)q_i - c_0, & \text{if } Q \leq D; \\
(P_0 - \delta)q_i - \gamma q_i (Q - D) - c_0, & \text{if } D < Q \leq D + \frac{P_0}{\gamma}; \\
-\delta q_i - c_0, & \text{otherwise.} \end{cases}
$$

(2.2.3)
2.2.2 Nash Equilibrium

Now the strategy of daily production to each manufacturer can be stated as a profit-maximization technique: for each player \( i \), \( q_i^* \) solves the following optimization problem:

\[
\max_{0 \leq q_i < \infty} \pi_i(q_1^*, \ldots, q_{i-1}^*, q_i, q_{i+1}^*, \ldots, q_n^*)
\]

It is well-known that such a strategy profile \( (q_1^*, \ldots, q_n^*) \) is called a Nash equilibrium.

Based on the linear assumption of price function, we have the following result:

**Theorem 2.2.1** Assume that the product depreciation function \( \Gamma \) is linear. If \( p_0 > \delta \), there exists a unique Nash equilibrium \( (q_1^*, \ldots, q_n^*) \) to the problem where if \( D \geq n \frac{p_0 - \delta}{\gamma} \),

\[
q_i^* = \frac{1}{n} D; \text{ if } D < n \frac{p_0 - \delta}{\gamma},
\]

\[
q_i^* = \frac{1}{n+1} \left( D + \frac{p_0 - \delta}{\gamma} \right) \text{ for each } i.
\]

**Proof:** Let \( Q = \sum_{i=1}^{n} q_i \). According to the linear assumption, the payoff function can be rewritten as follows:

\[
\pi_i(S) = \begin{cases} 
(P_0 - \delta)q_i - c_0, & \text{if } q_i \leq D - \sum_{k \neq i} q_k; \\
-\gamma q_i^2 + (P_0 - \delta - \gamma \sum_{k \neq i} q_k + \gamma D)q_i - c_0, & \text{if } D - \sum_{k \neq i} q_k < q_i \leq D + \frac{p_0}{\gamma} - \sum_{k \neq i} q_k; \\
-\delta q_i - c_0, & \text{otherwise.}
\end{cases}
\]

(2.2.4)

Therefore we have
\[
\frac{\partial \pi_i}{\partial q_i} = \begin{cases}
  P_0 - \delta, & \text{if } q_i \leq D - \sum_{k \neq i} q_k; \\
  -2\gamma q_i + P_0 - \delta - \gamma \sum_{k \neq i} q_k + \gamma D, & \text{if } D - \sum_{k \neq i} q_k < q_i \leq D + \frac{P_0}{\gamma} - \sum_{k \neq i} q_k; \\
  -\delta, & \text{otherwise}.
\end{cases}
\]

First we consider the case that \( D > \frac{n}{n+1} \frac{P_0 - \delta}{\gamma} \). We prove that \( S^* = (q_1^*, \ldots, q_n^*) \) is a Nash equilibrium if \( q_i^* = \frac{1}{n} D \).

For any \( i \), let \( S = (q_1^*, \ldots, q_i^*, q_i, q_i^*, \ldots, q_n^*) \). If \( q_i < D - \sum_{k \neq i} q_k^* \), then
\[
\pi_i(S) = (P_0 - \delta)q_i - c_0 < (P_0 - \delta)(D - \sum_{k \neq i} q_k^*) - c_0 = (P_0 - \delta)(D - \frac{n-1}{n} D) - c_0 = (P_0 - \delta)(\frac{1}{n} D) - c_0 = \pi_i(S^*)
\]

If \( D - \sum_{k \neq i} q_k^* < q_i \leq D + \frac{P_0}{\gamma} - \sum_{k \neq i} q_k^* \), then
\[
\frac{\partial \pi_i}{\partial q_i} = -2\gamma q_i + P_0 - \delta - \gamma \sum_{k \neq i} q_k^* + \gamma D < -2\gamma(D - \sum_{k \neq i} q_k^*) + P_0 - \delta + \frac{1}{n} \gamma D = -2\gamma(D - \frac{n-1}{n} D) + P_0 - \delta + \frac{1}{n} \gamma D = -\frac{2}{n} \gamma D + P_0 - \delta + \frac{1}{n} \gamma D = -\frac{1}{n} \gamma D + P_0 - \delta
\]

Since \( D \geq n \frac{P_0 - \delta}{\gamma} \), we obtain \( \frac{\partial \pi_i}{\partial q_i} \leq 0 \). Thus \( \pi_i(S^*) \geq \pi_i(S) \).

If \( q_i \geq D + \frac{P_0}{\gamma} - \sum_{k \neq i} q_k^* \), then \( q_i \geq q_i^* \). It follows that \( \pi_i(S) = -\delta q_i - c_0 \leq -\delta q_i^* - \cdots \).
\( c_0 \leq (P_0 - \delta)q_i^* - c_0 = \pi_i(S^*) \). So we have proved that \( S^* \) is a Nash equilibrium.

Next, we consider the case when \( D \leq n \frac{P_0 - \delta}{\gamma} \). Since the derivation of the function \(-\gamma q_i^2 + (P_0 - \delta - \gamma \sum q_k^* + \gamma D)q_i - c_0 \) at point \( D - \sum q_k^* \) is non-negative, it is easy to see that the maximum of the payoff function \( \phi_i(S) \) lies in the interval \([D - \sum q_k^*, D + \frac{P_0 - \delta}{\gamma} - \sum q_k^*]\). The Nash equilibrium of the problem is then the solution of the following set of linear equations (first-order conditions):

\[
q_i^* + \frac{1}{2} \sum_{k \neq i} q_k^* = \frac{1}{2} D + \frac{(P_0 - \delta)}{2\gamma}, \quad i = 1, \ldots, n, \tag{2.2.5}
\]

solving the set of equations yields

\[
q_i^* = \frac{1}{n+1} (D + \frac{P_0 - \delta}{\gamma}) \text{ for each } i.
\]

Now we prove the uniqueness. First we prove that a strategy profile \( S^* = (q_1^*, \ldots, q_n^*) \) is a Nash equilibrium only if it satisfies the following condition:

\[
q_i^* \geq D - \sum_{k \neq i} q_k^*, \text{ for all } i \leq n. \tag{2.2.6}
\]

Suppose that there exist \( i_0 \) such that \( q_{i_0}^* < D - \sum_{k \neq i} q_k^* \). Let \( q_{i_0} = D - \sum_{k \neq i} q_k^* \). Since \( P_0 - \delta > 0 \), it is obvious that \( \pi(S^*) < \pi(q_1^*, \ldots, q_{i_0-1}^*, q_{i_0}, q_{i_0+1}^*, \ldots, q_n^*) \). Therefore, \( S^* \) is not a Nash equilibrium. According to equation 2.2.6, if there is a \( j \) such that \( q_j^* \geq D - \sum_{k \neq i} q_k^* \), then for all \( i \), \( q_i^* = \frac{1}{n} D \). Otherwise, \( S^* \) must satisfy the first-order condition 2.2.5. Therefore, \( q_i^* = \frac{1}{n+1} (D + \frac{P_0 - \delta}{\gamma}) \) for each \( i \).
The following example illustrates a simple application of Theorem 2.2.1:

**Example 2.2.2** Consider a PC marketplace with \( n \) suppliers. Assume that \( p_0 = 1800 \), \( \gamma = 0.25 \), \( \delta = 1600 \), \( c_0 = 0 \) and \( D = 2000 \). Then the price function is

\[
P(Q) = \begin{cases} 
1800, & \text{if } Q \leq 2000; \\
2300 - 0.25Q, & \text{if } 2000 < Q \leq 7200; \\
0, & \text{otherwise}. 
\end{cases}
\]

For each manufacturer \( i \), its profit is decided by the following function:

\[
\pi_i(S) = \begin{cases} 
200q_i, & \text{if } q_i \leq 2000 - \sum_{k \neq i} q_k; \\
-0.25q_i^2 + 200q_i - 0.25q_i \sum_{k \neq i} q_k, & \text{if } 2000 - \sum_{k \neq i} q_k < q_i \leq 7200 - \sum_{k \neq i} q_k; \\
-1600q_i, & \text{otherwise}. 
\end{cases}
\]

The following table lists the equilibrium production in the case of no more than 6 manufacturers.

<table>
<thead>
<tr>
<th>Number of Manufacturers</th>
<th>Equilibrium Production</th>
<th>Aggregate Production</th>
<th>Market Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
<td>2000</td>
<td>1800</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>2000</td>
<td>1800</td>
</tr>
<tr>
<td>3</td>
<td>700</td>
<td>2100</td>
<td>1775</td>
</tr>
<tr>
<td>4</td>
<td>560</td>
<td>2240</td>
<td>1740</td>
</tr>
<tr>
<td>5</td>
<td>467</td>
<td>2333</td>
<td>1717</td>
</tr>
<tr>
<td>6</td>
<td>400</td>
<td>2400</td>
<td>1700</td>
</tr>
</tbody>
</table>
According to the TAC SCM specification, the production capacity of each PC manufacturer is 2000 cycles per day. This allows each manufacturer to produce up to 400 PCs per day (364 on average). The last row in the table shows that if the market situation follows the setting of the example, a TAC SCM agent should maximize its use of assembly line capacity, which coincides with the real situation in TAC SCM.

**Corollary 2.2.3** Let $S^*$ be the Nash equilibrium in Theorem 2.2.1, then for any $i \leq n$,

$$\pi_i(S^*) = \frac{\gamma}{(n+1)^2} (D + \frac{p_0 - \delta}{\gamma})^2 - c_0.$$

The result shows that the changing of the component price has a quadratic effect on agent profit. This explains why reducing component cost is one of the dominant strategies in the TAC SCM game.

### 2.2.3 Varying Customer Demands

It is easy to see that our model above is different from the classical Cournot model. Since market prices of each product are capped with customers’ RFQ reserve prices, the manufacturers cannot fully control the market with their production. The variation results in the difference of Nash equilibrium: if market demand is so big that a small reduction of price would result in a significant drop of manufacturer’s profit, no manufacturer has the incentive to produce more than its market share. This can be illustrated by the following example.
Example 2.2.4 Consider a duopoly where two manufacturers compete a market of one PC product. Assume that $p_0 = 2000$, $\gamma = 0.5$, $\delta = 1000$, $c_0 = 0$.

Suppose that the current market demand $D = 3000$. Then $D > n\frac{p_0 - \delta}{\gamma}$. According to Theorem 2.2.1, the equilibrium quantity for each manufacturer is 1500, which coincides with the amount of its market share. Therefore, no manufacturer has the incentive to produce more than it should. Figure 2.2 depicts the payoff curve in this situation where the peak of payoff function of a manufacturer is located at exactly its market portion (1500).

Figure 2.3 illustrates a small market demand situation where $D = 1000$, which is less than $n\frac{p_0 - \delta}{\gamma}$. It can be seen that a manufacturer would benefit from producing an extra 167 units.
2.3 Product Market Models: Price Competition

It is to be noted that in the real TAC SCM game, the depreciation coefficient $\gamma$ could vary with market demands. However the fluctuation of market demand is non-linear. In section 3.3.5.3 we will introduce a smooth-regression approach to reduce the non-linear factors of market demands on the market-clearing price and derive the correlation between the depreciation coefficient and market demand.

2.3 Product Market Models: Price Competition

In the last section, we made daily production decisions based on a comprehensive perspective of the market. In other words, we assumed that all products in the market are homogeneous with a uniform price for every product and every agent. Additionally, we did not consider the limitation of production capacity.
of each agent. In the real TAC SCM game, however, prices of products result from a first-price sealed-bid auction. Agents bid for customer orders with their own prices. The quantity an agent supplies is limited by its nominal capacity of production and its component supply[3].

Once the production amount has been decided, the produced products will be put on to the market to participate in the price competition. Economists have found that price competition is normally more severe than quantity competition[34]. In an oligopoly, where more than one firm competes on price, the market equilibrium would lead to a situation where all firms earn zero profits. This extreme situation is known as the Bertrand paradox[29]. Fortunately, the TAC SCM product market does not fit into the simple Bertrand model of price competition since the market contains multiple product selling and the production capacity of each manufacturer is limited. Additionally, production costs of each manufacturer are different. Due to the complexity of these features, the analysis of price competition becomes much harder.

To simplify the exploration, we still keep the assumption of homogeneous products but relax the assumption of uniform prices and production costs. Formally, let $q_i$ be the scheduled daily production of agent $i$ and $\delta_i$ the marginal cost of the products. Assume that $o_i$ is the total ordered amount the agent receives on a day at an average price of $p_i$ (different orders may have different prices). Additionally, we assume that $\eta_i$ is the average inventory cost of the product. Note that this cost is not necessarily the real inventory carrying cost
but is composed of the cost of unsold products at the end of the game, the revenue of the products when they are sold in the future (referred to as a negative value) and bank interest. Therefore, the value of $\eta_i$ can be negative if the revenue can eventually offset the cost. Using these assumptions, the one-day profit of agent $i$ will be:

$$\pi_i = o_i p_i - \delta_i q_i - \eta_i (q_i - o_i)$$  \hspace{1cm} (2.3.8)

Putting this aside, we consider how much an average agent can earn in one day. Again we use $D$ to denote the customer demand for the product per day. Let $p$ and $\delta$ be the average market price and average component cost, respectively. An average agent can then earn on the day:

$$\pi_{\text{average}} = (p - \delta) \frac{D}{n}$$  \hspace{1cm} (2.3.9)

where $n$ is the total number of agents.

If agent $i$ expects its profit to be no worse than an average, i.e.,

$$o_i p_i - \delta_i q_i - \eta_i (q_i - o_i) \geq (p - \delta) \frac{D}{n}$$

then the total ordered amount it receives should satisfy:

$$o_i \geq \frac{(p - \delta) \frac{D}{n} + (\delta_i + \eta_i) q_i}{p_i + \eta_i}$$  \hspace{1cm} (2.3.10)
If the actual order an agent receives is less than it expects (does not satisfy the condition (2.3.10)), the agent will need to lower it’s bidding price; otherwise it should keep or raise its product price.

2.4 Component Market Model

Different from the product market, the market of components is a typical constant-supply market. Market supply is fixed with only a small reverting random walk. Each supplier of components has an allocated output quota for each component type it supplies, called nominal capacity and denoted by $C_{nominal}$. A supplier accepts orders assuming that it will have $C_{nominal}$ products available every day for each component. The price of a component is determined by its market demand capped with a base price, which is specified in the game specification.

Consider a single component: Let $C_{ordered}(d,d')$ denote the total ordered amount of the component up to day $d$ which are requested for delivery on day $d'$ (where $d' > d$). Specifically, $C_{ordered}(0,d') = 0$ for any $d'$ since no component order can be made on day 0 according to the TAC SCM specification. Then, the total available capacity from day $d$ to day $d'$ will be:

$$C_{available}(d,d') = \sum_{j=d}^{d'-1} (C_{nominal} - C_{ordered}(d,j))$$

Note that the nominal capacity on day $d'$ is not included since it is assumed that products can only be shipped on the next day after they are produced.
With all the information, the component supplier decides its product price by using the following formula:

\[
p(d, d') = p_{\text{base}} (1 - \delta p \frac{C_{\text{available}}(d, d')}{(d' - d)C_{\text{normal}}})
\]  

(2.4.11)

where \(p(d, d')\) denotes the day \(d\)'s price for the product requested for delivery on day \(d'\); \(p_{\text{base}}\) is its base price and \(\delta p\) is a price discount factor, which is assumed to be a fixed value at 50% in the TAC SCM specification. The following proposition is straightforward from the equation 2.4.11:

**Proposition 2.4.1** Let \(d_1 > d\) and \(d_2 > d\). If \(d_1 < d_2\), then \(p(d, d_1) \geq p(d, d_2)\).

From the pricing function of components, we can easily see that the earlier an order for a component is placed, the more available production capacity for the component is left, and the cheaper the component will be. Since it is not allowed \(C_{\text{available}}\) to be negative, the discount of component price is always between 0 to 50% on the base price. Notably, ordering components on day 0 can results in receiving the top discount of 50%, i.e., \(p(0, d') = 0.5p_{\text{base}}\) for any \(d' > 0\). This gives every agent a great bargain. Unfortunately, this setting led the component market into an unintended situation in the TAC SCM competition (see more analysis in section 3.3.1).
2.5 Discussion and Related Work

We are probably the first to formally model TAC SCM markets based on classic economic theories and to use the models as a basis of strategy development throughout our trading agent design. The following two chapters will show how the market modelling can be applied to an e-Marketplace development and trading agent design. We will show the effectiveness of our modelling approaches in solving key decision-making problems that an agent may face in TAC SCM. While most approaches and principles we used are not restricted to TAC SCM but can be applied to any typical e-Markets, it does not mean that our market models can provide solutions to all practical problems. The reason lies in some inherent limitations and pitfalls of TAC SCM design economy.

Similarly, Wellman(1995) presented a market model for a well-defined class of distributed configuration design problems. Given a design problem, the model defines a computational economy to allocate basic resources to agents participating in the design. The result of running these “design economies” constitutes the market solution to the original problem. For some simple examples, the market model offers useful conceptual viewpoints for analyzing distributed design problems. However, analysis shows that the design economies, due to their inherent restrictions and shortcomings, are not guaranteed to find optimal designs[72]. Enhancements thus become important ongoing issues in e-Market development.
We will discuss the limitations and pitfalls of TAC SCM e-Market in chapter 5, together with our enhancement efforts.

### 2.6 Summary

This chapter presented theoretical models of TAC SCM, based on existing economic models.

Since the downstream market in TAC SCM (i.e. product market) is characterized by the combination of quantity competition and price competition between manufacturers (agents), we specified the quantity competition by a variation of Cournot model and viewed the price competition as an extension of Bertrand game. Since the market supply is fixed with only a small reverting random walk, we specified upstream market with a constant-supply model.

The aim of these models is to provide solutions to several key problems in the TAC SCM trading agent design, including decision-making of daily production, product pricing and component procuring. We have already seen some intuitive results derived from the models. More applications of these models and results will be discussed in the next chapter.

The TAC SCM game is a typical representation of a broad range of real supply chain situations. The game not only provides a competitive environment to evaluate different trading strategies and different structures of agents, but also act as a test-bed for examining artificial e-Market rules. There is no
doubt that the modelling approaches introduced here can also be applied to
modelling and analyzing other e-Trading systems.
Chapter 3

TAC SCM Trading Agent Design

and Analysis

3.1 Introduction

Unlike the previous chapter, this chapter will concentrate on the agent-side modelling and analysis. We will present some key strategies that have been applied in the design of the “Jackaroo” agent.

TAC SCM has attracted more than 30 research groups from universities and institutes all over the world. As one of the participants in the TAC SCM, the “Jackaroo” team, representing the University of Western Sydney (UWS), contributed a trading agent to the annual competitions. In the TAC SCM-03 tournament, the “Jackaroo” agent finished in third place in the qualifying round and fourth place in the seeding round. Unfortunately, the agent was unable to proceed in the final round due to a network problem at the conference venue. In the TAC SCM-04 tournament, the “Jackaroo” agent finished first place in the

---

As a core member of “Jackaroo” team, my contribution includes the design, development, testing and refinement of the pricing, bidding, procuring and information collection modules.
qualifying round, fifth place in the seeding round, first place in the quarter-final round and fifth place in the semi-final round. Although the “Jackaroo” agent was eliminated before the final round, it nevertheless was recognized by the research community as one of the most inventive and influential agents because its demonstration of several leading strategies in the tournaments.

After giving an overview of its architecture, this chapter reviews the key strategies used by the “Jackaroo” agent. We focus on the problems of component procuring, product pricing, production scheduling and price forecasting, which are known as the most important issues involved in TAC SCM trading agent design.

Most of the strategies are rooted in theoretical models (as described in the previous chapter) and statistical analysis of the game. By differentiating quantity competition and price competition in the TAC SCM product market, we can solve the problem of daily production in a variation of a Cournot model and the product pricing problem in an extension of Bertrand game. By using a constant-supply model in the TAC SCM component market, “Jackaroo” became the first agent to use the “aggressive day-0 procurement” approach, which was later adopted by most of agents. To cope with the non-linear fluctuation of product price, we introduced an approach of smooth-regression to reduce the non-linear factors of market demands on the market-clearing price. The market-clearing price thus can be predicted. All the solutions to the problems have been implemented and tested in our “Jackaroo” agent.
3.2 Agent Architecture

Each simulated TAC day represents a decision cycle for an SCM agent. During each time unit the agents must solve the following highly-related problems:

- procuring components
- marketing products
- scheduling production
- scheduling delivery
- managing storage

The architecture of “Jackaroo” was designed with these tasks and their relationships in mind (Figure 3.1).

The component procuring module communicates with suppliers via RFQs and selects supplier offers to accept, among those which are received in response to these RFQs. The product marketing module decides how to assign offer prices to each customer RFQ. The production scheduling module decides how many of each SKU to assemble each day. The delivery scheduling module decides which orders to ship to which customers, using product inventory. The storage managing module monitors both production inventory and component inventory. It gives warnings when any product or component is running short, so as to maintain balance in both inventories. The bookkeeper module
Figure 3.1: Architecture of “Jackaroo”
### TAC SCM Decision Problem

Objective:
- Maximize Expected Profits

Inputs:
- Model of Quantity Competition in Product Market
- Model of Price Competition in Product Market
- Model of Component Market
- Product Inventory
- Component Inventory
- Customer RFQs
- Supplier Offers
- Customer Orders
- Game Data (Historical and Real-time)

Intermediate Tasks:
- Predicting Market-clearing Price
- Planning and Scheduling
- Determining Pricing Policy
- Determining Procuring Strategy

Outputs:
- Bidding Decision (Selection of RFQs and Pricing)
- Procurement Decision
- Production Decision
- Delivery Decision

---

**Figure 3.2**: TAC SCM decision problem

records all the transactions and run-time information to provide statistical data for decision-making. The coordinator module coordinates the activities of all other modules and facilitates communication between modules. The objective of all these modules is to maximize expected profits. A high-level description of the TAC SCM decision problem is presented in Figure 3.2.

The following sections present some strategies employed in “Jackaroo” to support key decision-making processes.
3.3 Strategizing “Jackaroo”

3.3.1 Component Procuring

The component market model (discussed in section 2.4) has suggested that ordering components earlier would reduce cost by up to 50%. One successful strategy first adopted by the “Jackaroo” agent is to order all components for the whole game at the very beginning of the game. We call this approach “aggressive day-0 procurement”.

As we know, each manufacturer is capable of producing around 360 PCs per day, which requires 90 units of each type of CPUs and 180 units of each type of the other components. Since the number of RFQs an agent can send to each supplier is limited (maximum ten RFQs per supplier per day), we packaged several days component usage into one RFQ (about 270 to 540 for CPU and 540 to 1080 for the he couldn’t have other components). With this approach, we can order all the components for a whole game in the first few days. Figure 3.3 shows the result of using this strategy during the early state of a TAC SCM-03 game (game 418 on tac6.sic.se).

In this game, we ordered 213 day’s worth of components in the first 8 days. The due date of each order was set a little bit earlier (5 days in advance) than was actually needed in order to avoid possible delay. By using this approach, we only paid 63.66% of full price for all components.

This “not-so-aggressive” strategy was only used in the qualifying round
Figure 3.3: An example of component procurement scheduling (data from TAC SCM-03 game 418 on server tac6.sic.se).
because we quickly realized that ordering components on day 0 would be rewarded by receiving the maximal discount of 50%. From the seeding round, we changed our strategy and ordered all components on day 0. As expected, “Jackaroo” demonstrated an overwhelming advantage in the tournament by using this aggressive strategy. As a result, most of the other competitors applied similar strategies as well. A severe and inherent problem in the design market of TAC SCM then appeared.

From each agent’s perspective, the main effect of being aggressive is on its own component procurement profile. If every agent is aggressive, however, it can significantly change the characteristics of the game environment and cause the component market to malfunction. Games in the rest of the TAC SCM-03 tournament showed that due to huge demands of components in day-0 and the limitation of supplier’s production capacity, component delay became the biggest headache of every agent. The result of a game heavily depended on the random arrival of components (often referred to as “lottery effect” in the TAC SCM competition). From the competition’s perspective, the game became less fair and thus less interesting.

Many contingent remedies have been proposed by researchers to overcome the “aggressive day-0 procurement” problem. Some turn out to have no noticeable effect and some come with negative “side-effects”. In Chapter 4, we will discuss this issue in more detail. We will develop a new supplier behavior model based on a four-round negotiation mechanism in the component mar-
ket and prove that it is a not only a solution to the problem, but it also brings many interesting elements into the original TAC SCM scenario.

### 3.3.2 Product Pricing

Recall that the price competition model (discussed in section 2.3) indicates that if agent $i$ expects its profit to be no worse than the average, i.e.

$$o_ip_i - \delta_i q_i - \eta_i(q_i - o_i) \geq (p - \delta) \frac{D}{n}$$

then the total ordered amount it receives should satisfy:

$$o_i \geq \frac{(p - \delta) \frac{D}{n} + (\delta_i + \eta_i)q_i}{p_i + \eta_i} \quad (3.3.1)$$

If the actual order that an agent receives is less than what is expected (i.e. does not satisfy the condition (3.3.1)), the agent will need to lower its bidding price; otherwise it should keep or raise its product price.

On the other hand, the condition (3.3.1) also provides a solution to the problem of product pricing. If we view $o_i$ as the average ordered amount an agent receives, then the agent should set its product price as

$$p_i = \frac{(p - \delta) \frac{D}{n} + (\delta_i + \eta_i)q_i}{o_i} - \eta_i \quad (3.3.2)$$

We remark that the actual pricing algorithm we used in our agent is much
more complicated than simply calculating the above expression. Application of this expression relies on the estimation of the current market price, component cost and the inventory cost. In section 3.3.5.2, we will explore the forecasting approaches of market product price.

3.3.3 Price Coefficient Learning

The decision of daily production is one of the biggest challenges faced by every SCM agent designer, especially if most components are ordered at the very beginning. The product model provides a guideline for such decision-making. Once we have a price function $P(Q)$, Theorem 2.2.1 would suggest a production quantity for each manufacturer. However, several problems need to be solved before it can be actually used:

- How many factors affect the market-clearing price?

- Is the market clearing-price of each product linearly correlated to its redundant supply in the PC market?

- If the correlation is linear, how is the price descent coefficient $\gamma$ determined?

Besides market demand and aggregate quantity of products in the market, there are several other factors influencing the market product price: agent’s pricing strategy, stages of a game, component supply, and so on. Nevertheless,
market demand and market supply are still the most significant factors affecting market prices. Figure 3.4 shows a statistical result produced by an off-line learning program:

The data are taken from TAC SCM-03 game 427 on server tac5.sics.se. To reduce the influence of other factors (e.g. the lead time for the ordered components to arrive, and the dumping strategy many agents adopt at the end of the games), we ignore the data of the first and last 50 simulation days, then average the statistical results of every five days. Instead of using multiple linear regression, we simply consider the correlation between the following two variables:
- **price descent** (the difference between customer’s reserve price $p_0$ and market-clearing price $P(Q)$),

- **redundant market supply** (the difference between product supply $Q$ and market demand $D_m$).

The statistical result shows a highly linear correlation between these two variables. The coefficient of correlation is up to 0.8357. The associated price decline coefficient $\gamma$ is 2.0496. This encourages us to use linear method to approximate the price function. With the linear regression method, we can easily estimate the price decline coefficient $\gamma$, which could range from 0.1 to 10, depending on market demands, different competitors and stages of a game.

Interestingly, the product market model (discussed in section 2.2) implies that an agent should maximize its use of its production capacity if the market demand is above the average level of 2000 PCs per day. Considering that there are some unproductive days, this level can be even lower. According to our observation, if the average market demand is higher than 1500, maximizing production capacity is always profitable if components can be obtained at a good price.

Note that a linear approximation$^2$ of the price function does not imply the price descent coefficient to be constant during a whole game. In fact, the price descent coefficient $\gamma$ is a function of time and market demand. Normally, the

---

$^2$Recall that in the above example we ignored the data of the first and last 50 simulation days, then averaged the statistical results of every five days.
abstract value of the coefficient would be much smaller than normal in the early stage of a game because most agents would wait for components (less market supply) and much higher at the very end since most agents would tend to dump their goods before the game ends. Also note that the linear correlation between market supply and market price varies with the market demand. In 3.3.5, we will discuss how to cope with the non-linear fluctuation of product price. We will use a smooth-regression approach to reduce the non-linear factors of market demands on market-clearing price.

### 3.3.4 Production Scheduling

There are many tradeoffs faced by TAC SCM agent designers. One of them is to decide which types of products should be produced for inventory\(^3\). On the one hand, one should produce more profitable products in order to keep a good inventory level of these products. On the other hand, a balance of all products in inventory should be maintained to maximize the ability of bidding for customer orders. In the implementation of our agent, we have adopted a dynamic weighting approach to support decision-making for production scheduling.

Let \(v(k)\) denote the current inventory of product \(k\) \((0 \leq k < 16)\) and \(v_{\text{max}}\) be their maximal value. We first calculate the inventory weight \(w_1(k)\):

\(^3\)This was not applicable for those agents who did not produce inventory.
\[ w_1(k) = \frac{(v_{\text{max}} - v(k))}{\sum_{i=0}^{16} (v_{\text{max}} - v(i))} \]

Next, we calculate the market profit weight. Let \( p_m(k) \) and \( c(k) \) be the current market price and component cost of product \( k \), respectively. The profit weight \( w_2(k) \) is defined as follows:

\[
w_2(k) = \begin{cases} 
0, & \text{if } p_m(k) - c(k) \leq 0; \\
\frac{(p_m(k) - c(k))}{\sum_{p_m(i) - c(i) > 0} (p_m(i) - c(i))}, & \text{otherwise}. 
\end{cases}
\]

Finally we combine these two weights with a balance coefficient \( \lambda \), which was learnt from previous games:

\[ w(k) = \lambda w_1(k) + (1 - \lambda)w_2(k), \ k = 0, \cdots, 16 \]

The actual quantity produced for inventory is then

\[ q(k) = \frac{w(k) \times r_{\text{cycles}}(k)}{u_{\text{cycles}}(k)} \]

where \( r_{\text{cycles}} \) represents the current remaining production cycles after producing all the outstanding orders and \( u_{\text{cycles}}(k) \) the required cycles per unit of product \( k \).
3.3.5 Market Price Forecasting

According to the TAC SCM-03 specification, a customer’s reserve price (i.e. the highest price that the customer is willing to pay) for a product is setup as 75% to 125% of its base cost. This means that there is not much margin between baseline product cost and its market value. Manufacturers have to try their best to reduce their component cost. Since there is no cost for inventory and bank interest is relatively low, one can stock a large amount of components while their prices are low. Therefore, finding an appropriate method to predict the market price of components is essential for agents to maintain low cost. Besides, we know from 3.3.2 that market price prediction plays an extremely important role in product pricing. Observations on the TAC SCM games showed that there were many factors which affected the market prices of PC products. Besides customer demands and market product supply, agents’ individual pricing strategies, stages of the game and component supply also played very important roles in the product market. In this section we investigate the relationship between market demand and product price, and use the relation as a way to predict the market-clearing price.

3.3.5.1 Correlation between Market Supply and Market-Clearing Price

One assumption that we have made in our product market model is that there would be a linear relationship between market supply and market-clearing
price. To verify this assumption, we conducted a statistical analysis on the TAC SCM-03 games. The data we used came from the last round of the TAC SCM-03 final competition (a total of 16 games on both tac5.sics.se and tac6.sics.se servers). The reason we chose these games is that all these games were played by exactly the same players. For this reason, we can minimize the effect of pricing strategies used by different players. For each game, we picked up all negotiation data for all simulated days of the game, including customer RFQ quantities, offered product quantities, customers’ reserve prices and market-clearing prices of all PC products. These data were classified and averaged according to the ranges of market demands (see Table 3.1). The coefficients of correlation between market supply and depreciation of market price (the gap between customer reserve price and market-clearing price) were then calculated. Finally, we averaged all the data from the 16 games. Table 3.1 shows the statistical result.

This table clearly shows a significant linear correlation between quantities of market supply and depreciations of market price in the middle range of market demand. The correlation, however, is weaker at the lower end and the higher end of market demand. The reason is: When market demand is low, competition in the product market becomes intense. Therefore, agents apply a non-linear price-cutting strategy to compete for customer orders. On the other hand, when market demand exceeds market supply, the market-clearing price will approach the customer reserve price. Product prices are then mostly
§3.3  Strategizing “Jackaroo”

<table>
<thead>
<tr>
<th>Market Demand</th>
<th>Average Market Demand</th>
<th>Average Market Supply</th>
<th>Average Reserve Price</th>
<th>Average Market Price</th>
<th>Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 800</td>
<td>728</td>
<td>2313</td>
<td>2003</td>
<td>957</td>
<td>0.53</td>
</tr>
<tr>
<td>800-1200</td>
<td>957</td>
<td>2802</td>
<td>2019</td>
<td>1093</td>
<td>0.55</td>
</tr>
<tr>
<td>1200-1600</td>
<td>1395</td>
<td>2178</td>
<td>2098</td>
<td>1622</td>
<td>0.83</td>
</tr>
<tr>
<td>1600-2000</td>
<td>1782</td>
<td>2040</td>
<td>2117</td>
<td>1807</td>
<td>0.81</td>
</tr>
<tr>
<td>2000-2400</td>
<td>2202</td>
<td>1325</td>
<td>2177</td>
<td>1988</td>
<td>0.68</td>
</tr>
<tr>
<td>2400-2800</td>
<td>2597</td>
<td>1631</td>
<td>2176</td>
<td>2006</td>
<td>0.52</td>
</tr>
<tr>
<td>2800-3200</td>
<td>3040</td>
<td>2027</td>
<td>2177</td>
<td>1982</td>
<td>0.24</td>
</tr>
<tr>
<td>&gt; 3200</td>
<td>3440</td>
<td>2360</td>
<td>2199</td>
<td>2000</td>
<td>0.20</td>
</tr>
</tbody>
</table>

The result is based on the TAC SCM-03 Final games 1264-1271@tac5.sics.se and 1423-1430@tac6.sics.se.

**Table 3.1:** Correlation between market-clearing price and market supply in the product market.

determined by agents’ pricing strategies rather than the quantity of market supply.

### 3.3.5.2 Smooth-Regression for Price Forecasting

We have seen that the price function of the product market may not simply be approximated by a linear function. This means that if we use a linear forecasting approach to predicting market price, the result may not be satisfying.

Table 3.2 shows the result of prediction by using linear regression on the data we used above (TAC SCM-03 Finals). Columns 2 and 3 in the table give sample estimates of regression coefficients $\beta_0$ and $\beta_1$, respectively, under different market demands (averaged values of 16 games). Columns 4 and 5 show the actual average market-clearing prices and their forecasted values. Column 6 shows the precisions of forecasting. Note that in the calculation of regression
The result is based on the TAC SCM-03 Final games 1264-1271@tac5.sics.se and 1423-1430@tac6.sics.se.

<table>
<thead>
<tr>
<th>Market Demand</th>
<th>$b_0$</th>
<th>$b_1$</th>
<th>Actual Market Price</th>
<th>Forecasted</th>
<th>Forecast Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 800</td>
<td>441.6</td>
<td>0.25</td>
<td>957</td>
<td>1380</td>
<td>55.8</td>
</tr>
<tr>
<td>800-1200</td>
<td>325.7</td>
<td>0.21</td>
<td>1093</td>
<td>1488</td>
<td>63.9</td>
</tr>
<tr>
<td>1200-1600</td>
<td>115.5</td>
<td>0.18</td>
<td>1622</td>
<td>1735</td>
<td>93.0</td>
</tr>
<tr>
<td>1600-2000</td>
<td>95.8</td>
<td>0.11</td>
<td>1807</td>
<td>1827</td>
<td>98.9</td>
</tr>
<tr>
<td>2000-2400</td>
<td>90.7</td>
<td>0.16</td>
<td>1988</td>
<td>1729</td>
<td>87.0</td>
</tr>
<tr>
<td>2400-2800</td>
<td>122.3</td>
<td>0.15</td>
<td>2006</td>
<td>1666</td>
<td>83.1</td>
</tr>
<tr>
<td>2800-3200</td>
<td>161.7</td>
<td>0.04</td>
<td>1982</td>
<td>1886</td>
<td>95.2</td>
</tr>
<tr>
<td>&gt; 3200</td>
<td>161.1</td>
<td>0.01</td>
<td>2000</td>
<td>1996</td>
<td>99.8</td>
</tr>
</tbody>
</table>

Table 3.2: Market price forecasting by using linear regression

coefficients, the dependent variable we used is not the market-clearing price, but the price difference between reserve price and market-clearing price (price depreciation). Therefore, the linear regression line gives us only an estimate of market price depreciation. We still need to recover the market-clearing prices by using the average reserve prices:

$$p_{\text{forecast}} = p_{\text{reserve}} - (b_0 + b_1Q) \quad (3.3.3)$$

where $p_{\text{reserve}}$ and $Q$ are the average reserve price and average quantity of market supply, respectively. $p_{\text{forecast}}$ is the forecasted value of market-clearing price.

Observing Table 3.2, we found that the forecasted results for the high range of market demands is not as bad as we expected. The reason for this is that the actual market prices in this range uniformly distribute around the regression
line (this can be observed from a scatter diagram of the raw data). Therefore, the average values of market-clearing price match the forecasted values very well. Nevertheless, the quality of prediction in the low range of market demands is quite poor. To solve the problem, instead of introducing a non-linear forecasting model, we adjusted the regression parameters so that the non-linear features can be recovered from the trend of parameter variations. We call this approach smooth-regression.

First, we calculate the estimated values of the regression parameter $b_0$ and $b_1$ against market demands on the previous game data (see Table 3.2 columns 2 and 3) by using a simple linear regression model. Next, we find the best fitting curve to match each regression parameter. Figures 3.5 and 3.6 depict the respective fitting curves of the regression parameters $b_0$ and $b_1$ varying with market demands. The sample data comes from columns 2 and 3 of Table 3.2.

Figure 3.5 shows that the trend of $\beta_0$ over market demand fits a quadratic curve. This results in a fitting function as follows:

$$\hat{\beta}_0 = 0.00012155 \cdot D^2 - 0.57967 \cdot D + 762.73 \quad (3.3.4)$$

However, the trend of $\beta_0$ fits very well in a straight line with the norm of
residuals as small as 0.08725:

$$\hat{b}_1 = -0.000076109 \times D + 0.29303$$  \hfill (3.3.5)$$

With these functions, we can easily yield the estimated values of the linear regression coefficients. This fitting process smoothed the stochastic factors from the agents’ pricing strategies. Table 3.3 shows the result of market price prediction by using the smoothed regression parameters, where the values of $\hat{b}_0$ and $\hat{b}_1$ were calculated by functions (3.3.4) and (3.3.5) respectively. We can see that the regression parameters in the lower end of market demand have been successfully adjusted by the smoothing process. The forecasting precision becomes more relevant to the market demand. The overall forecasting
quality has also been significantly improved.

We have shown that with smooth regression, we can keep the linear assumption for the purpose of market price forecasting. One may argue that the forecasting results shown in Table 3.3 do not represent the general online quantity of prediction since all the parameters were yielded via off-line learning. To show whether the smoothing functions can be used in online price forecasting, we chose 20 games from the TAC SCM-03 seeding round 2 (game 628-637 at tac5.sic.se and game 778-787 at tac6.sics.se) and analyzed the forecasted results. Table 3.4 shows a summary of the statistical results. Note that the estimated values of $b_0$ and $b_1$ are calculated by the smoothing functions (3.3.4) and (3.3.5) respectively, also we have used the average daily market demands as the values of independent variables. The forecasted values are

![Figure 3.6: Fitting curve of $b_1$ over market demand.](image-url)


<table>
<thead>
<tr>
<th>Market Demand</th>
<th>$b_0$</th>
<th>$b_1$</th>
<th>Actual Market Price</th>
<th>Forecasted Price</th>
<th>Forecast Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 800</td>
<td>405.1</td>
<td>0.24</td>
<td>957</td>
<td>1048</td>
<td>91.0</td>
</tr>
<tr>
<td>800-1200</td>
<td>319.3</td>
<td>0.22</td>
<td>1093</td>
<td>1083</td>
<td>99.1</td>
</tr>
<tr>
<td>1200-1600</td>
<td>190.6</td>
<td>0.19</td>
<td>1622</td>
<td>1500</td>
<td>92.1</td>
</tr>
<tr>
<td>1600-2000</td>
<td>115.7</td>
<td>0.16</td>
<td>1807</td>
<td>1680</td>
<td>92.7</td>
</tr>
<tr>
<td>2000-2400</td>
<td>75.7</td>
<td>0.13</td>
<td>1988</td>
<td>1935</td>
<td>97.0</td>
</tr>
<tr>
<td>2400-2800</td>
<td>77.1</td>
<td>0.10</td>
<td>2006</td>
<td>1943</td>
<td>96.5</td>
</tr>
<tr>
<td>2800-3200</td>
<td>123.8</td>
<td>0.06</td>
<td>1982</td>
<td>1928</td>
<td>97.5</td>
</tr>
<tr>
<td>&gt; 3200</td>
<td>207.0</td>
<td>0.03</td>
<td>2000</td>
<td>1918</td>
<td>96.1</td>
</tr>
</tbody>
</table>

*The result is based on the TAC SCM-03 Final games 1264-1271@tac5.sics.se and 1423-1430@tac6.sics.se.*

Table 3.3: Market price forecasting by using smooth regression.

given by Equation (3.3.3) (the average reserve prices are omitted from the table). The result shows that the smooth regression produces a reasonably high quality online price forecasting. In our agent implementation, the parameter can be dynamically adjusted through a feedback mechanism.

### 3.3.5.3 Estimation of Price Parameters

As we have seen, our solution to daily production is based on the linear assumption of price function. We have also shown that this assumption is not true when market demand is extremely low or high. By using the smooth regression function given by (3.3.4) and (3.3.5), we can adjust the price function to reduce the non-linear effects of market demand.

Let $Q$ be the aggregate quantity of the products and $\hat{P}(Q)$ the estimated value of market-clearing price. According to equation 3.3.3,

$$\hat{P}(Q) = p_{\text{reserve}} - (\hat{b}_0 + \hat{b}_1 Q)$$
§3.3 Strategizing “Jackaroo”

<table>
<thead>
<tr>
<th>Market Demand</th>
<th>$b_0$</th>
<th>$b_1$</th>
<th>Actual Market Price</th>
<th>Forecasted Price</th>
<th>Forecasted Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 800</td>
<td>402.8</td>
<td>0.2372</td>
<td>1209</td>
<td>1039</td>
<td>86.0</td>
</tr>
<tr>
<td>800-1200</td>
<td>317.6</td>
<td>0.2198</td>
<td>1339</td>
<td>1074</td>
<td>80.2</td>
</tr>
<tr>
<td>1200-1600</td>
<td>191.4</td>
<td>0.1871</td>
<td>1691</td>
<td>1362</td>
<td>80.5</td>
</tr>
<tr>
<td>1600-2000</td>
<td>115.2</td>
<td>0.1571</td>
<td>1782</td>
<td>1518</td>
<td>85.2</td>
</tr>
<tr>
<td>2000-2400</td>
<td>77.0</td>
<td>0.1275</td>
<td>1834</td>
<td>1696</td>
<td>92.5</td>
</tr>
<tr>
<td>2400-2800</td>
<td>76.8</td>
<td>0.0958</td>
<td>1934</td>
<td>1845</td>
<td>95.4</td>
</tr>
<tr>
<td>2800-3200</td>
<td>123.6</td>
<td>0.0618</td>
<td>1905</td>
<td>1831</td>
<td>96.1</td>
</tr>
<tr>
<td>&gt; 3200</td>
<td>206.3</td>
<td>0.0314</td>
<td>1937</td>
<td>1837</td>
<td>94.9</td>
</tr>
</tbody>
</table>

The result is based on the TAC SCM-03 Seeding round game 628-637 @tac5.sics.se and 778-787@tac6.sics.se.

Table 3.4: Market price forecasting by using smooth regression (continued).

Since $P(D) = p_0$, by using (3.3.4) and (3.3.5), we have:

$$
\hat{p}_0 = \hat{P}(D)
$$

$$
= p_{\text{reserve}} - (\hat{b}_0 + \hat{b}_1 D)
$$

$$
= p_{\text{reserve}} - 0.000045441D^2 + 0.28664D - 762.73
$$

Similarly, the estimate for depreciation coefficient is given by:

$$
\hat{\gamma} = \hat{b}_1 = -0.000076109D + 0.29303.
$$

Once the price function is ready, it is easy to decide the daily production.

For instance, suppose that the average customer reserve price is $p_{\text{reserve}} = 2000$, then

$$
\hat{p}_0 = -0.000045441D^2 + 0.28664D + 1237.27.
$$

$$
\hat{\gamma} = -0.000076109D + 0.29303.
$$

Assume that the average marginal cost of PC products is $\delta = 1378$.\footnote{This value is the average cost of PC products calculated, based on the TAC SCM final}
cording to Theorem 2.2.1, an agent should fully use its production capacity once the market demand is more than 1450 PCs per day.

3.4 Discussion and Related Work

Most strategies adopted by the TAC SCM agent designers are not based on formal economic models and are targeting specific problems in the TAC SCM game domain. Therefore, it is hard to adopt similar approaches in other market situations.

Botticelli[12] suggests to solve the stochastic problem of scheduling using a Sample Average Approximation (SAA) method in an online setting to find today’s optimal schedule, given predictive information about the future.

NaRC[17] proposes a technique for use in TAC SCM that assists the decision-making process for purchases of direct goods. Based on projections for future prices and demand, RFQs are constructed and quotes are accepted that optimize the level of inventory each day, while minimizing total cost. The problem is modeled as a Markov decision process (MDP), which allows for the computation of the utility of actions to be based on the utilities of consequential future states. Dynamic programming is then used to determine the optimal quote requests and accepts at each state in the MDP.

\[
\delta = \frac{\text{total material cost}}{\text{total PC sold}}
\]

Therefore, the “inventory cost” has been considered.
DeepMaize[11] suggests a heuristic for solving the supplier offer acceptance problem using a rollout method, following one or two stage approximations of the multi-stage stochastic program as the base policy during rollouts.

Nevertheless, most traditional agent design efforts have focused on either the autonomous behavior aspects of agency, or on interaction among agents. Bradshaw’s KAoS agents[16] are BDI agents in a CORBA environment. Agents have capabilities based on existing document management applications. JADE[50] is an agent framework that has been used to build trading agents, however its primary emphasis is on building multi-agent systems that comply with FIPA specifications for inter-agent communications, and with flexible deployment in a network environment. Vetsikas and Selman[65] show a method for studying design tradeoffs in a trading agent. RETSINA[62] suggests both a multi-agent architecture with a variety of agent roles, and an architecture for individual agents that provides communications, planning, scheduling, and execution monitoring. This architecture could probably be adapted to the TAC SCM domain, but its planning and communication capabilities would not likely be especially useful.

Ultimately the TAC SCM problem domain does not require the sort of flexible cognitive and social elements of the more traditional agent designs. In the next chapter, we will discuss some enhancements to TAC SCM designed to create a more general market scenario and to make use of those traditional agent design efforts.
3.5 Summary

This chapter presented the architecture and strategies of “Jackaroo”, one of the most influential agents in the TAC SCM tournaments. Most of the strategies are rooted in theoretical modelling (described in the previous chapter) and statistical analysis of TAC SCM games. We outlined the basic theory and algorithms dealing with essential decision-making problems including component procuring, product pricing, production scheduling and price forecasting.
Negotiation Mechanism for the TAC SCM Component Market

4.1 Introduction

The TAC SCM product market operates with the standard multi-item auction mechanism, which worked very well in previous games. Nevertheless, the component market exploited an artificial supply-dominant market mechanism, under which buyers have no influence on the market. With the TAC SCM-03 specification, the price of a component is determined by the following formula[3]:

\[
p(d, d + i) = p_{\text{base}}(1 - 0.5 \frac{C_{\text{available}}(d, d + i)}{iC_{\text{nominal}}})
\]  

(4.1.1)

where \(p(d, d + i)\) is the price of the component under consideration on day \(d\) for delivery on day \(d + i\); \(p_{\text{base}}\) is the pre-specified price of the component (base price); \(C_{\text{nominal}}\) is the nominal production capacity of the product line that produces the component (constant); \(C_{\text{available}}(d, d + i)\) is the total free capacity
of the product line from day $d$ to day $d + i$.

It is easy to see from the formula that the price of each component, at the beginning of a game is significantly lower than any of the following days because all production capacity is available at the beginning of a game. This feature encourages all agents to order their components at the beginning (referred to as “day-0-procurement”). Even worse, because the acceptance of component orders and production scheduled are based on the nominal capacity of a product line regardless of its actual production capacity, which varies day to day during a game, it frequently happens that a supplier cannot fulfill its component contracts when the actual production capacity is under the nominal capacity. The random arrival of components prevents manufacturers from scheduling their production and fulfilling their PC contracts. Most of the business strategies regarding production scheduling, price forecasting and cost management are not applicable in the agent design.

Another serious problem with the component market is that the supply of each component is independent from market demand and market price, and is determined by a random walking mechanism. This allows an agent to corner a whole market of a particular component (typically the cheapest component). A number of lottery effects have been seen, especially in the final round of the TAC SCM-04 games.

Many efforts to reduce the supply of “lottery effects” were made during previous games. For instance, the price policy (Formula 4.1.1) was changed
and a heavy storage cost was introduced in the TAC SCM-04 specification in order to discourage day-0-procurement. Nevertheless, the problems of the malfunctioning component market still exists.

We argue that the problem of the TAC SCM component market is not the improper setting of the formulas for component price or production capacity, but lies more deeply in its market mechanism. In fact, the current component market is not a market at all, in the economic sense, as no market mechanism is applied to the supply of components, suppliers cannot adjust their production according to market demands, and the exchange of information between suppliers and agents is limited. As a result, the behavior of the component market is far from realistic.

This chapter introduces a new negotiation mechanism into the component market by allowing agents (component buyers) to negotiate with suppliers on price, due date and quantity. Moreover, autonomy is given to each supplier to decide its production capacity and selling price. A supplier is allowed to vary its production capacity according to the law of supply and to allocate its products to buyers according to buyers offer price, due date and required quantity. A mathematical analysis is given to show that the new market model can effectively solve the existing problems of the component market while keeping

---

1The work presented here was done in collaboration with Dr. Dongmo Zhang and inspired by his initial idea of giving a new behavior model to suppliers. My portion of the work included several refinements of the initially proposed model, server/agentware implementation and testing.
the good features of the original proposal of the game (such as its simplicity and zero free inventory). A new TAC SCM server and AgentWare based on the existing system has been implemented and will be available for download from the Internet\textsuperscript{2}.

### 4.2 Negotiation Protocol for the Component Market

In the original TAC SCM specifications, the interaction between component suppliers and PC manufacturers is limited to three rounds: agents sending requests for quote (RFQ) to suppliers, suppliers making offers to agents RFQs and agents replying with orders for purchasing components. An agent, acting as a manufacturer, is not allowed to bargain over price or due date with suppliers. They can only accept or refuse the suppliers' offers. In our proposed component market model, we allow an agent to bargain with suppliers price, delivery date and quantity. The interaction between agents and suppliers consists of the following four rounds:

- An agent sends RFQs to suppliers, querying the price and availability of a component whenever the agent want to buy a component from a supplier;

\textsuperscript{2}The implemented system is based on SICS’s TAC Supply Chain Management Simulator, which is copyrighted by SICS.
• When a supplier receives an RFQ, the supplier can choose to either ignore the RFQ or send back an offer to the agent with its offer price, available date and available quantity;

• An agent has three choices when it receives an offer from a supplier: reject the offer by doing nothing; accept the offer by sending back an order; or bargain with the supplier over price, delivery date, and/or quantity by replying with a counter offer;

• If a supplier receives an order, the supplier will sign the contract with no further negotiation. A supplier can, however, reject a counter offer if the counter offer does not meet the supplier’s current reserve price for the component and availability of the product. If a supplier accepts a counter offer, it will reply with a confirmation.

The protocol is described by the following figure 4.1.

4.3 Supplier Model

To demonstrate how negotiation strategies can be used under the proposed negotiation mechanism, we revamp the original supplier model so that a supplier can negotiate with component buyers on price, due date and supply quantity. The open model of the supplier will help us to evaluate the business strategies developed for participating agents. To make the presentation
Figure 4.1: Negotiation protocol between agent and supplier

simple, we will focus on the changed features of the supplier model. The un-
changed parts will be referred to as the TAC SCM specification[3].

The major changes are as follows:

- An adjustment has been introduced to the suppliers’ decision on current
  production to reflect the law of supply[34].

- Production scheduling is based on so-called guaranteed production capac-
ity (see below) rather than nominal capacity to ensure the fulfillment of all component contracts.

- Different price policies are applied to demonstrate the negotiation strategies of suppliers.

### 4.3.1 Assumptions

Most assumptions that were made in the TAC SCM-03 and the TAC SCM-04 specifications will remain true under the new model. Some changes have been made to accommodate the new supplier model. We assume that:

1. At the beginning of a day each supplier calculates its remaining production capacity for each product line it owns.

2. All orders and counter offers received by a supplier on the last day are then processed. Confirmations are sent to successful counter offerers immediately. The orders and accepted counter offers are scheduled for production. Production scheduling is based on the guaranteed production capacity.

3. Production of any products is strictly based on a production schedule. End products are automatically transferred to inventory.

4. Orders are delivered to the manufacturers and charged exactly on their due dates. Only complete orders are shipped. No early payment or de-
5. At the end of the day each supplier processes all the RFQs received from agents on this day and makes offers to manufacturers.

### 4.3.2 Daily Production Capacity

In the original TAC SCM game scenario it is specified that each supplier has a dedicated product line, with a nominal capacity $C_0$ (the expected or mean production) for each component type it supplies. The actual production capacity $C(d)$ of each component line varies day to day and is determined by the following formula:

$$C(d) = \max(C_{\text{min}}, C(d-1) + \sigma_1 C_0 + \sigma_2 (C_0 - C(d-1)))$$

where $C_{\text{min}}$ is the minimal capacity the supplier of the component can commit. $\sigma_1$ is a random number, representing the unpredictable factor that affects the production of the product line and $\sigma_2$ is a constant which reverts the capacity back to nominal. The setting of the values are $C_{\text{min}} = 0$, $\sigma_1 = \text{random}(-0.05, +0.05)$ and $\sigma_2 = 0.01$ in the TAC SCM-03 and the TAC SCM-04 specifications. As we have pointed out before, this setting suffers from several severe problems. On the one hand, there is no lower limit in the production capacity for product lines. A supplier cannot fulfill its commitment to the accepted orders when capacity is down to a low level, because suppliers’
scheduling is based on the nominal capacity. This caused many lottery affects in the previous TAC SCM games. On the other hand, a supplier is unable to adjust its production when the demand of a product is increasing or decreasing. This allowed one agent to corner a component market and block other agents’ procurement. To solve this problem, we introduced the following changes in the determination of supplier production capacity:

- Allow a supplier to have a non-zero minimal capacity.
- Allow a supplier to vary its production to reflect the law of supply[34].

With the above ideas, each supplier’s production capacity is determined by the following formula:

$$C(d) = \max(C_{\text{min}}, C(d-1) + \sigma_1 C_0 + \sigma_2 (C_0 - C(d-1)) + C_{\text{adjust}})$$  \hspace{1cm} (4.3.2)

where

$$C_{\text{adjust}} = \sigma_3 \frac{p_{\text{average}} - p_{\text{expected}}}{p_{\text{base}}} C_0$$

$p_{\text{expected}}$ is the supplier’s expected price of the component; $p_{\text{average}}$ is the supplier’s last day average selling price of the component. $p_{\text{expected}}$ is currently set as $0.75 \times p_{\text{base}}$ and $\sigma_3 = 0.1$.

For other parameters, we simply keep the original setting in TAC SCM except for $C_{\text{min}}$. In our current implementation of the supplier agent, $C_{\text{min}}$ was set to a non-zero value, which was half of the nominal capacity $C_0$. With the
non-zero minimal production capacity, a supplier is able to guarantee its production without a breach of contract. This can be of great help to the supplier when making a decision to accept long term ordering and also reduces the lottery effects in the component market.

We remark that those factors that affect supplier’s production can be classified into two categories: objective factors and subjective factors. The random movement and reverting walk simulate the objective factors that a supplier cannot control, whereas the capacity variation with selling price is a decision by the supplier itself.

4.3.3 Guaranteed Production Capacity (GPC)

As we have mentioned, a supplier uses the minimal production capacity of a product line for its long-term production scheduling. For short-term scheduling, a supplier can expect more production than the minimum because the actual production capacity on each day is normally much higher than the minimal capacity and the variation of capacity is limited, as shown in the formula 4.3.2.

Suppose that the current production capacity of a product line is $C(d)$. The maximal reduction of the capacity that is out of the owner’s control can only be:

- Reduction due to random variation: $-\sigma_1 C_0$;
• Reduction due to back reverting: $\sigma_2(C_0 - C(d))$.

where $\sigma_1$ and $\sigma_2$ represent the boundaries of random variation.

**Definition 4.3.1** Let $C(d)$ be the production capacity of a component line on day $d$. For each $i \geq 0$, let

$$C_{GPC}(d, d + i) = \max(C_{\min}, (1 - \sigma_2)^i C(d) - (\sigma_1 - \sigma_2)C_0 \sum_{k=0}^{i-1} (1 - \sigma_2)^k).$$

We call $C_{GPC}$ the guaranteed minimal capacity (GPC) of the product line.

It is easy to show that if we ignore the capacity adjustment by the price factor in Formula 4.3.2, then $C_{GPC}(d, d + i)$ will always be less than the actual production capacity $C(d + i)^3$. Therefore a supplier can always guarantee its production with the quantity $C_{GPC}(d, d + i)$. According to the standard setting of $\sigma_1$ and $\sigma_2$,

$$C_{GPC}(d, d + i) = \max(C_{\min}, 0.99^i C(d) - 20 \sum_{k=0}^{i-1} 0.99^k).$$

For instance, if the current production capacity of a product line is 500 units, the $C_{GPC}$ will not reach the minimum until after 10 days. If the current capacity is 800, then it takes 23 days to reach minimum capacity. The total accumulated extra capacity that the supplier can use for scheduling purpose is 5999, which is equivalent to a 12 day nominal production capacity. This leaves a gap for agents to bargain over component price and due date.

---

3In the implementation of the supplier model, a supplier always checks the current production schedule to find any possible over-commitment when it adjusts the production capacity for a product line.
4.4 Production Scheduling

Before we describe a supplier’s offering procedure, we shall explain how a supplier schedules its production once a supply contract is signed.

Considering product line of a component type, let $S(d, d + i)$ denote the production quantity scheduled on day $d$ for day $d + i$. Initially, we have

$$S(0, d) = 0 \text{ for any } d \geq 0.$$  

Suppose that the day $d - 1$’s scheduling has been done, which is:

$$S(d - 1, d - 1 + i) \text{ for any } i \geq 0.$$  

For each order or accepted counter offer for the component, we need to reschedule the product line so that the order can be produced on time. Similar to the original TAC SCM supplier model, a supplier always tries to defer the production of an order until the due date approaching in order to minimize inventory cost and productive capital. Thus, a supplier will firstly try to fill an order (accepted counter offer) on the due date. If this fails, the supplier will then try the day before until all the free capacity between the current day and the due date is full. A significant difference between our scheduling mechanism and the original is that we use GMC for scheduling rather than the nominal capacity. This makes it possible for a supplier to guarantee the fulfillment of component contracts while maximizing the use of its production capacity.

Let $\{(q_k, d_k) : k = 1, \cdots, m\}$ be all the orders or counter offers a supplier ac-
cepts for a component on day $d$. We calculate the schedule of day $d$ recursively by putting these orders or counter offers into the existing schedule one by one:

For each $i \geq 0$, let

$$S^0(d, d + i) = S(d - 1, d + i).$$

Suppose that $S^{k-1}(d, d + i)$ is the result of scheduling after the first $k - 1$ orders or accepted counter offers have been processed. Now we put the order (accepted counter offer) $(q_k, d_k)$ into the schedule:

1. for any day $d'$ after $d$ ($d' > d_k$),

$$S^k(d, d') = S^{k-1}(d, d')$$

2. for the days between $d$ and $d_k$, we schedule the production backward from $d_k$ to $d$ recursively.

Let $q^{d_k} = q_k$. For each $t = d_k, d_k - 1, \ldots, d$,

(a) if $q' \geq C_{GPC}(d, t) - S^{k-1}(d, t),$

$$S^k(d, t) = C_{GPC}(d, t) \text{ and } q' - 1 = q' - C_{GPC}(d, t).$$

(b) if $q' < C_{GPC}(d, t) - S^{k-1}(d, t),$

$$S^k(d, t) = q' + S^{k-1}(d, t) \text{ and } q' - 1 = 0.$$

**Definition 4.4.1** An order or an accepted counter offer $(q_k, d_k)$ is scheduled if $q^{d-1} = 0$. 
We will prove in the next section that any order or counter offer accepted by a supplier can be scheduled for production by the supplier and is guaranteed to be produced and delivered on time under the new supplier model.

4.5 Supplier Pricing Policy

In this section, we focus on suppliers pricing policies in offer making and counter offer processing. We will see that a supplier uses different pricing policies in offer making and counter offer processing in order to maximize its negotiation power.

4.5.1 Making Offers

Each day, each supplier collects all RFQs it received on that day, calculate the total requested quantity and make offers to selected RFQs.

Let $R = \{(rq_j, rdd_j) : j = 1, \cdots, n\}$ be all the RFQs a supplier received for a component and $Q_{RFQ} = \sum_{j=1}^{n} rq_j$ be the total RFQed quantity. The procedure to process the RFQs is as follows:

1. Calculate the earliest completion date $ecd$ to produce the RFQed components:

   $ecd = \min_u (Q_{RFQ} + \sum_{k=d+1}^{u} S(d,k) \leq (u-d-1)C_{min})$
4.5 Supplier Pricing Policy

Note that the checking for free capacity starts from \( d + 1 \) because the actual orders won’t arrive until the next day.

2. Calculate offer due date for each RFQ \( r_j = (rq_j, rdd_j) \):

\[
odd_j = \begin{cases} 
  rdd_j & \text{if } rdd_j \geq ecd + 1; \\
  ecd + 1 & \text{otherwise.}
\end{cases}
\]

3. Calculate offer price \( op_j \) for the RFQ above:

\[
op_j = p_{base}(1 - discount)
\]

where

\[
discount = \bar{\delta}(1 - \frac{Q_{RFQ} + \sum_{k=d+1}^{odd_j-1} S(d,k)}{(odd_j - d - 1)c_{min}});
\]

\( p_{base} \) is the base price of the component;

\( \delta \) is the discount coefficient (standard value is 0.5).

In our implementation of the game server, not all RFQs are processed. Instead, we exploited a slightly complicated selection mechanism, based on the buyers’ previous order ratios (reputation) and current request quantities. An RFQ sent by an agent with a higher reputation and smaller order quantity has a higher processing priority. Once the total available minimal capacity of a product line for the whole game period has been reached, no further RFQs will be processed (a partial offer could be issued if a portion of the requested quantity can be met). Due to space limitations, we omit the details of such a
4.5.2 Counter Offer Processing

As we have described in the negotiation protocol, when an agent receives an offer from a supplier for a component, the agent can do one of two things: the agent can either directly order the component if it accepts all the conditions in the offer, or the agent can reply with a counter offer for a better price or an earlier delivery date.

On the supply side, whenever a supplier receives a valid order (referring to an offer that was sent by the supplier for the same component in the previous day), the supplier will simply accept the order and schedule its production immediately. For counter offers the process is slightly more complicated. Firstly, the supplier will separate those counter offers in which the requested quantity is no more than the offered quantity from the other counter offers. Secondly, based on the common rules implemented in most real marketplaces as well as reflected in the TAC SCM competition, these two types of counter offer are sorted according to the following criteria:

- the higher offer price receives higher priority;

- if the prices are equal, the earlier due date receives higher priority;

- if the prices and due dates are equal, the largest requested quantity receives higher priority;
• if the prices, due dates and requested quantities are equal, the RFQ that
arrived first receives higher priority.

These sorted lists are then put together with those with highest priority
in front. Let $coffer = \{(p_j, d_j, q_j) : j = 0, \cdots, m\}$ be the resulting list of all the
counter offers. If the supplier decides to accept a particular counter offer on the
list, the supplier will schedule its production immediately, otherwise, it will
move to the next one until all of the counter offers have been processed. Sup-
pose that the first $j - 1$ counter offers have been processed, we now consider
the counter offer $(p_j, d_j, q_j)$. Assume that the current production schedule is
$S(d,k)$. The supplier then accepts the counter offer if, and only if, it satisfies
the following conditions:

1. $q_j \leq \sum_{k=d}^{d_j-1} (C_{GPC}(d,k) - S(d,k))$

2. $p_j \geq p_{\text{base}}(1 - \text{discount})$, where

\[
\text{discount} = \delta(1 - \frac{q_j + \sum_{k=d}^{d_j-1} S(d,k)}{\sum_{k=d}^{d_j-1} C_{GPC}(d,k)})
\]

We call $p_{\text{reserve}} = p_{\text{base}}(1 - \text{discount})$ the reserved price of the supplier for
the counter offer.

Note that the production schedule is dynamic. It changes with order and
counter offer processing. The following proposition shows that any accepted
counter offer can be guaranteed to be produced and therefore be delivered on
time.
Proposition 4.5.1 For any counter offer \((p_j, d_j, q_j)\), if it is accepted, it can be scheduled for production.

4.6 Properties of the Suppliers Model

In this section, we present some important properties of the proposed negotiation mechanism and supplier model to show that the existing problems can be solved efficiently.

4.6.1 Orders vs. Counter Offers

As we have seen in the previous section, a supplier processes orders and counter offers separately. This makes it possible for any agent built on the original AgentWare to be used under the new game server without any changes. Since orders, however, are always processed before any counter offers, a question is raised: If an agent orders a component by sending a counter offer with exactly the same offer conditions, can the counter offer be accepted by the supplier as if an order was sent? The following theorem answers this question.

Theorem 4.6.1 Let \(o = (op, odd, oq)\) be an offer a supplier sent to an agent on day \(d\), where \(op, odd\) and \(oq\) are the offer price, offer due date and offer quantity, respectively. Let \(co = (cop, codd, coq)\) be the respective counter offer that was received by the supplier on day \(d + 1\). If \(cop = op\), \(odd = codd\) and \(oq = coq\), this counter offer will be accepted.
Proof: Let $Q_{RFQ}$ be the total quantity of all the accepted RFQs. According to the supplier’s offering procedure, we have

$$Q_{RFQ} + \sum_{k=d+1}^{odd-1} S(d,k) \leq (odd - d - 2)C_{\min}$$

On the other hand, before the counter offer is processed, the exiting production schedule should satisfy:

$$\sum_{k=d+1}^{odd-1} S(d+1,k) \leq Q_{RFQ} - oq + \sum_{k=d+1}^{odd-1} S(d,k)$$

Putting these two inequalities together, we have

$$oq \leq oq + (odd - d - 2)C_{\min} - (Q_{RFQ} + \sum_{k=d+1}^{odd-1} S(d,k))$$

$$\leq (odd - d - 2)C_{\min} - (Q_{RFQ} - oq + \sum_{k=d+1}^{odd-1} S(d,k))$$

$$\leq (odd - d - 2)C_{\min} - \sum_{k=d+1}^{odd-1} S(d+1,k)$$

$$\leq \sum_{k=d+1}^{odd-1} C_{GPC}(d+1,k) - \sum_{k=d+1}^{odd-1} S(d+1,k)$$

Therefore, the supplier can commit to the production of the counter offer. Note that there is one day delay between the offer date and the day the counter offer being processed. Now we calculate the reserve price of the supplier for the counter offer:

$$p_{\text{reserve}} = p_{\text{base}}(1 - \delta(1 - \frac{oq + \sum_{k=d+1}^{odd-1} S(d+1,k)}{\sum_{k=d+1}^{odd-1} C_{GPC}(d+1,k)}))$$

$$\leq p_{\text{base}}(1 - \delta(1 - \frac{Q_{RFQ} + \sum_{k=d+1}^{odd-1} S(d,k)}{\sum_{k=d+1}^{odd-1} C_{GPC}(d+1,k)}))$$

$$\leq p_{\text{base}}(1 - \delta(1 - \frac{Q_{RFQ} + \sum_{k=d+1}^{odd-1} S(d,k)}{(odd - d - 2)C_{\min}}))$$
Therefore, the counter offer can be accepted. 

According to the theorem, it is equivalent for an agent to order a component by sending a direct order or replying with a counter offer with exactly the same offer conditions. This is not trivial because a counter offer can offer a price that is higher than the offer price. The theorem guarantees that such a counter offer will not disturb supplier’s scheduling.

4.6.2 Large Procurement

The following proposition shows that the new supplier model discourages large procurement.

**Proposition 4.6.2** Let \( r = (rq, rdd) \) be an RFQ that an agent sent to a supplier for a component on day \( d \) and \( o = (op, odd, oq) \) be the supplier’s offer with respect to the RFQ. If \( rdd_i < odd_i \), then \( op_i > p_{base}(1 - \frac{\delta_{odd-di}}{odd-d}) \).

**Proof:** Let \( ecd \) be the earliest completion date for all the RFQs that a supplier received for a component on day \( d \). If \( rdd_i < odd_i \), we have \( odd_i = ecd + 1 \). Let \( \epsilon = (ecd - d)C_{min} - Q_{RFQ} - \sum_{k=d+1}^{ecd} S(d,k) \). According to the definition of Earliest Completion Date, it follows that

\[
0 \leq \epsilon < C_{min}
\]

Then
\( o_i = p_{\text{base}} \left( 1 - \delta \left( 1 - \left( \frac{Q_{\text{RFQ}} + \sum_{k=1}^{c_{\text{ud}}} \text{S}(d,k)}{(c_{\text{ed}} - d)c_{\text{min}}} \right) \right) \right) \)

\[ = p_{\text{base}} \left( 1 - \delta \left( 1 - \left( \frac{(c_{\text{ed}} - d)c_{\text{min}} - \varepsilon}{(c_{\text{ed}} - d)c_{\text{min}}} \right) \right) \right) \]

\[ = p_{\text{base}} \left( 1 - \delta \frac{\varepsilon}{(c_{\text{ed}} - d)c_{\text{min}}} \right) \]

\[ > p_{\text{base}} \left( 1 - \frac{\delta}{c_{\text{ed}} - d} \right) \]

\[ = p_{\text{base}} \left( 1 - \frac{\delta}{c_{\text{odd}} - d - 1} \right) \]

According to the proposition, if an RFQ asks for more than 1000 units for an early delivery, the discount it can expect will be less than 0.125, regardless of other RFQs (\( \delta = 0.5 \)). If each of the six agents requests 1000 units for a component on the same day, the discount each agent can expect will be about 0.02 (i.e. a minimum discount).

### 4.6.3 Day 0 Procurement

As we have seen above, the new supplier model discourages large procurement. It can therefore significantly mitigate the problems of “Day-0-Procurement” and market-corning. Nevertheless, the new model still retains the feature of encouraging long term component procurement like the TAC-04 supplier model. This is also applicable to day 0.

**Lemma 4.6.3** Let \( r_i = (r_{qi}, r_{ddi}) \) and \( r_j = (r_{qj}, r_{ddj}) \) be two RFQs for the same component and sent to the same supplier on day 0. \( o_i = (op_i, odd_i) \) and \( o_j = (op_j, odd_j) \) are the offers for each of the RFQs, respectively. If \( r_{ddi} \leq r_{ddj} \), then \( op_i \geq op_j \) and \( odd_i \leq odd_j \).
Proof: Let $ecd$ be the earliest completion date for all the RFQs the supplier received for the component on day 0. The offer due dates for $r_i$ and $r_j$ are then respectively:

$$\text{odd}_k = \begin{cases} rdd_k & \text{if } rdd_k > = ecd + 1; \\ ecd + 1 & \text{otherwise}. \end{cases} \quad k = i, j$$

Therefore, if $rdd_i \leq rdd_j$, then $\text{odd}_i \leq \text{odd}_j$. For the offer prices, since we have $S(0, d) = 0$ for all $d$, we have

$$op_k = p_{\text{base}}(1 - \delta(1 - \frac{Q_{\text{RFQ}}}{(rdd_k - 1)c_{\text{min}}})) \quad k = i, j$$

Thus, if $\text{odd}_i \leq \text{odd}_j$, $op_i \geq op_j$. 

The lemma shows that if an agent makes a long-term procurement for a component on day 0, the agent can gain a good discount. This does not mean, however, that the “Day 0 Procurement” problem will return. Before we explain why, we remark that the order quantity decision on day 0 for each agent under the new supplier model is no longer an optimization problem, but a game-theoretical problem because the discount an agent can get depends on the total quantity of RFQs, rather than individual RFQs. Agents have to compete with each other for day 0 procurement. The following theorem shows that agents have no incentive to join the competition of day 0 procurement under the new supplier model.

**Theorem 4.6.4** Let $n$ be the number of agents playing in a game. For each component from a supplier, if all agents request the component on day 0, and all with their equilibrium quantities, and ask for the same due date, the discount each agent can get is $\left(\frac{\delta}{n+1}\right)$ and the associated equilibrium quantity for each agent is $\frac{(dd-1)c_{\text{min}}}{n+1}$, where $dd$ is the common due date.
**Proof:** Let \( q_i \) be the total requested quantity from agent \( i \). According to the proof of Lemma 4.6.3, the discount each agent can receive is then:

\[
\text{discount} = \delta (1 - \frac{\sum_{k=1}^{n} q_k}{(dd - 1)C_{\text{min}}})
\]

Thus, the benefit for ordering the component on day 0 for agent \( i \) will be:

\[
B_i = p_{\text{base}} \cdot q_i \cdot \text{discount}
\]

To maximize \( B_i \), the first order condition for agent \( i \) is:

\[
(dd - 1)C_{\text{min}} - \sum_{k=1}^{n} q_k - q_i = 0, \quad i = 1, \cdots, n
\]

Solving the set of equations gives the Nash equilibrium order quantity for each agent:

\[
q_i^* = \frac{(dd - 1)C_{\text{min}}}{n + 1}
\]

The discount each agent can gain is then:

\[
\text{discount}^* = \frac{\delta}{n + 1}
\]

Let \( n = 6 \) and \( \delta = 0.5 \). According to the theorem, if each agent order components on day 0 with the equilibrium quantity, the discount each agent can expect will be 0.07, which is much less than the average discount a supplier is giving (supplier’s expected discount is 0.25). Therefore, no agent will have the incentive to involve in the severe competition of day 0 procurement. Since only half of production capacity is scheduled ahead, the other half of production capacity is left for short-term ordering.
An agent can expect to acquire their component in a reasonable price during normal days. However, a small amount of day 0 procurement is still profitable for agents, which also benefits suppliers because a reasonable amount of long-term contacts are always welcome to any enterprise.

### 4.6.4 Bargaining Range

Finally, we consider the bargaining range of component purchasing from the buyers perspectives. Since the new negotiation model for the component market is basically a special type of auction, the actual component price depends on market competition. In addition, an agent has no means to probe the exact value of a supplier’s reserve price because it changes dynamically. Nevertheless, there are still several ways for agents to guess the supplier’s reserve price in order to make a good bargain.

Figure 4.6.4 shows the result of an experiment in which a dummy-agent-based agent competes with five other built-in dummy agents. The curves illustrate the suppliers’ offer prices and reserve prices for 2.0GHz CPUs on each day. It shows that the prices for day 0 are relatively less than its following days but higher than the average price. The offer prices are evenly distributed all the way in the game while the reserve prices start high and evenly distribute later on, reflecting the effects of suppliers’ self adjustment of production capacity. It can also be observed that the lower the offer price is, the less an agent can bargain. This is more visible for day 0 procurement.

Figure 4.6.4 illustrates the bargain range for due dates. It can be seen that the average bargaining range of due dates is about 4 to 5 days. The actual bargaining range, however, varies with the agents’ bargaining strategies. Normally, higher counter offer
prices give a wider bargaining range for due dates.

Figure 4.2: Agent’s bargaining range for component price (Pintel CPU 2.0GHz)

Figure 4.3: Agent’s bargaining range for component due date (Pintel CPU 2.0GHz)
4.7 Summary

In this chapter, we presented a new negotiation mechanism for the TAC SCM component market and a new supplier model to demonstrate how negotiation strategies can be applied in automated negotiation. With the proposed negotiation mechanism, both sellers and buyers are able to operate the component market autonomously. The new supplier model simulates the operation of modern enterprises in more aspects than the original proposal for the following reasons:

- A supplier always tries to maintain its credibility by the fulfillment of component contracts.
- A supplier always tries to utilize market competition to maximize its profits.
- A supplier always tries to balance long-term business relationships and short-term market supply.
- A supplier always tries to maintain its negotiation power.

With the new component market model, we have shown that most of the existing problems with the TAC SCM game, such as day 0 procurement, lottery effects of component supply and coverage of a particular component market, have been effectively solved.
Chapter 5

Enhancements of the TAC SCM Game

5.1 Introduction

The TAC SCM game presents participants with challenging decision making problems in a context of great strategic uncertainty[28]. Nevertheless, far from being like the real supply-chain environments it is intended to model, the game comes with some critical limitations which greatly prevent it from being widely used as a flexible research tool for our supply chain designers and trading agent developers. In the following sections, we will address those problems and limitations existing in the TAC SCM game and propose enhancements.

5.2 A Customer Behavior Model with Linear Reserve Price Function

Under the current TAC SCM specification, market demand is expressed as customer’s requests for quotes (RFQs). Each RFQ is for a quantity chosen uniformly in [Qmin,
Qmax]. The level of demands (i.e. total requested PCs) for each day is drawn stochastically from a Poisson distribution with a mean value, which also evolves according to a given stochastic process.

This stochastic feature of market demand contributes partly to market uncertainty in the game and tends to bring some real supply chain challenges in front of competitor agents, since it is agreed in vast supply chain literature that customer demand level is the primary uncontrollable factor in the market[1]. In the current TAC SCM game, however, the customer’s reserve price (i.e. the maximum price that a customer is willing to pay) is randomly chosen from an interval given in the game specification. This random nature of the reserve price disappointingly turns otherwise interesting market uncertainty into the complete intractability of customer behavior. Due to this intractability, there is no major determinant of potential profits available to the competing agents and discussions of price prediction, Cournot (quantity) competition, stock/production control strategies and so on, seem to lose their grounds.

Recall that in section 2.2.1, we specified the quantity competition in the product market with a variation of the Cournot model, assuming that the market-clearing price is a constant value if the aggregate quantity in the market is no more than the market demand. This is based on the analysis of real TAC SCM game data and the fact that if the market demand is more than the supply, agents can normally take customer orders at reserve price or a little bit less. This modelling approach provided enormous help in our agent design and empirical solutions to the above problems. Nevertheless, as far as the enhancement and refinement of the TAC SCM e-Market scenario is concerned, an explicit customer behavior model needs to be built.
In fact, as often discussed in economics literature[1], the reserve price, although varying from time to time, depends inevitably on how many products there are supplied in the market and how many products there are demanded at the same time.

Here we express the reserve price as a linear function of the aggregate quantity supplied in the market and the market demand:

\[
P_r(Q, M_d) = \begin{cases} 
P_0 - \alpha(Q - M_d), & \text{if } P_0 > \alpha(Q - M_d); \\
0, & \text{otherwise.} 
\end{cases} \quad (5.2.1)
\]

where

- \( P_r \): Customer reserve price, \( P_r \geq 0 \)
- \( Q \): Aggregate quantity of products supplied, \( Q \geq 0 \)
- \( M_d \): Market demand, \( M_d \geq 0 \)
- \( \alpha \): Price coefficient, \( \alpha > 0 \)
- \( P_0 \): A constant, \( P_0 > 0 \)

Note that the above price function differs from the price function we introduced in section 2.2.1 in two aspects: (1) In the new model, a customer’s reserve price is a function of two market variables - total supplies and total demands while in 2.2.1 we assumed that the market demand is a constant (i.e. average value); (2) In the new model, we can see that the the reserve price is allowed to have any value larger than 0 and it may be unlimitedly large when there is a huge market shortage while in 2.2.1, we assumed that the reserve price is always a constant in case the market demand exceeds market supply.
Also note that both manufacturers and customers require environmental information (e.g. demand, supply, price coefficient etc.) to support their decision making processes. An Information Sharing Mechanism (ISM) (discussed in Section 5.4) can be used to facilitate the necessary information exchange between manufactures and customers in an e-supply chain (or e-Market) scenario.

5.3 Extendible Production Capacity of Manufacturing Agents

Competitor agents are supposed to compete with each other in terms of price (i.e. Bertrand Competition) or production scheduling (i.e. Cournot Competition), or in most circumstances, the combination of both. However, Bertrand competition does not make sense if there is no constant cost associated with each agent’s production activities, especially the majority of competitor agents implements the prevailing strategy of ”Aggressive Day-0 Procurement”\footnote{A close examination of the game results in a key observation that prices for components on day 0 are half of the base prices and will never be lower after that day. Therefore, it would appear advantageous to agents to procure a large number of components on day 0.} [28], even the costs of each PC are similar for all agents; and Cournot competition does not make sense either if agents have no way to extend their production capacities when they are driven by large market demand to do so.

Therefore, the fact that PC assembly agents’ production capacities are fixed and non-extendable results in similar marginal costs for all agents. This is another limitation that weakens the flexibility of the entire simulated supply chain. To solve this
problem with minimal changes to the original game, we add another supplier called *Manufacturing Equipments Supplier* (MES) into the game scenario, as shown in Figure 5.1.

In the new scenario, competitor agents are allowed to extend their production capacities via buying additional *assembly cells* from MES. MES trades with assembly agents in the same way as those component suppliers do - through the RFQ mechanism. The values and ranges of a number of parameters related to MES’s production (e.g. nominal capacity of MES, base price per assembly cell, capacity of each assembly cell etc.) rely on the insights gained from trial games and simulations.

Nonetheless, it is easy to foresee that adopting the above change would affect the competition in two ways: (1) Since each agent may own a different number of assembly cells, the marginal costs for them may be different. This cost uncertainty would inevitably affect pricing/planning strategies; (2) In the new scenario, agents can choose
both quantity and price. When capacity is costly and cannot be altered quickly, competitor agents would need to make crucial decisions on capacity investments at an early stage of the game\(^2\).

### 5.4 Agent Cooperation via an Information Sharing Mechanism

The performance of a supply chain depends critically on how its members coordinate their decisions. And it is hard to imagine coordination without some form of information sharing[19].

Information sharing refers to those situations where more than one player of a game share the same piece of information. It can occur either cooperatively or non-cooperatively; either horizontally (among competing agents) or vertically (between upstream and downstream members of a supply chain). Information that may be shared includes customer demand, production cost, point-of-sale data, inventory status, etc.

Although in the current TAC SCM game, the periodic price report and market report provide some degree of information sharing[5], a formal information sharing mechanism is lacking in the game to assist the study of the following interesting issues: How do prices/quantities and expected profits with and without information

\(^2\)In fact, the resulting equilibrium may be similar to the Cournot equilibrium because while prices are easy to change quickly it is likely to be difficult to change the production capacity in the short run. In the later stages of the game, an agent has no incentive to conquer a higher market share than it is able to supply.
sharing depend on (1) the type of competition (Cournot or Bertrand), (2) the type of uncertainty (demand or cost uncertainty), (3) the correlation of the stochastic demands/costs among firms, and (4) the type of shared information[52].

### 5.4.1 An Information Sharing Mechanism

We present the following information sharing mechanism (Figure 5.2) as a further enhancement of TAC SCM.

In the center of the framework is a key role (it may be played by the server), called *Information Sharing Coordinator* (ISC), which organizes and coordinates all possible information sharing activities (including periodic price/market report) that may occur in the supply chain with all relevant information. ISC accomplishes its task in *domains*. Each domain has its own registered participants and a unique *domain interest* which is associated with one type of information sharing. Not all the roles (i.e. manufacturers, supplies, customers, etc.) in the supply chain are allowed to join each domain. In fact, ISC needs the input of *domain policies* to establish or initialize domains, decide which roles may or may not join each domain, which roles are the *beneficiaries of shared information* (BSI) and which are the *contributors of shared information* (CSI). A domain member may act as BSI and CSI at the same time.

The domain policies are broadcast to all supply chain members at the beginning of the game. During the course of a game, each supply chain member makes its independent decision to join an information sharing domain or not. If a supply chain member decides to join a domain, it sends an *application for joining* (AFJ) to ISC, indicating which domain it intends to join. After verifying that the sender of an AFJ
Figure 5.2: An information sharing mechanism
is eligible to join the specific domain, ISC registers the applicant in the domain. If a supply chain member decides to quit an information sharing domain, it sends a notice of quitting (NOQ) to ISC. ISC then de-registers the sender from the domain.

Information sharing in a domain is realized in this way: at some time interval (daily, weekly etc), ISC will get specific pieces of private information that is of the domain’s interest from each of the contributors in the domain, it then processes all the information it obtains and sends the (processed) shared information (SI) to all the beneficiaries. Again, ISC depends on the domain policies to decide the frequency of information sharing/updating, how to process the raw private information it obtains and in what form (e.g. market report, inventory report, etc.) the shared information is sent to the beneficiaries.

### 5.4.2 Impact of Sharing Private Inventory Information

A significant part of supply chain management research is devoted to understanding the role of information sharing in achieving supply chain coordination.

Research on information sharing in the supply chain was initialized by Forrester[19] who demonstrated that the replenishment orders generated by a stage in a supply chain exhibit more volatility than the demand the stage faces. Lee, Padmanabhan and Whang[19] named this phenomenon the “Bullwhip” effect[15]. The famous Beer Game[41], which simulates the “Bullwhip” phenomenon has recently attracted much attention from both supply chain management researchers as well as multi-agent system (MAS) designers. Sharing of customer/market demand information is often suggested to combat the undesirable “Bullwhip” effect. Vives[19] found that allowing
for price competition and differentiated products, sharing information about common demand intercept can increase firms’ profits. Esther Gal-Or[19] found that firms will be strictly more profitable when they share demand information. Esther Gal-Or and Carl Shapiro[19] showed that there will be an equilibrium for firms to share their private cost information if firms are Cournot competitors producing complementary products, and the only uncertainty is each firm’s marginal cost[47].

While most models of information sharing among oligopolists deal with cases of exchanging private information about either constant marginal cost or market demand, few have considered the situation of sharing private inventory information.

In this section we deal with information sharing among horizontal competitors. Our intention is to see if the sharing of their private inventory information has any significant impact on the supply chain. We build an elaborate simulation based on the enhanced TAC SCM scenario that we introduced in previous sections. We then test the impact of sharing private inventory information against out performance measures, under a number of supply chain configurations.

5.4.2.1 The Model

We consider a two-tier duopoly market with two competing PC assembly firms (agents) offering identical products to the customer. The two firms are engaged in an iterative Cournot competition game for a period of $n$ days, i.e. making quantity (production) decisions independently on each day.

The market demand $M_d$ for each day is a random and non-negative variable with a mean value of constant $M$. The daily market price $P_m$, as shown below, is a linear
function of market demand $M_d$ and the aggregate quantity of products, $Q$, that the duopolists are holding. Let $l_1$ and $l_2$ $(l_1, l_2 \geq 0)$ denote the stock level of firm 1 and firm 2 respectively.

\[
P_m(Q, M_d) = \begin{cases} 
    P_0 - \alpha(Q - M_d), & \text{if } P_0 > \alpha(Q - M_d); \\
    0, & \text{otherwise.}
\end{cases} \tag{5.4.2}
\]

\[
Q = l_1 + l_2. \tag{5.4.3}
\]

where

- $\alpha$: price coefficient, $\alpha > 0$

- $P_0$: a constant, $P_0 > 0$

Two types of information sharing may occur in the game (both on a daily basis). One is in the form of production report describing how many PCs each firm produced in the previous day. This type of information sharing is compulsory for both firms, so each of them is guaranteed to know its opponent’s latest strategy. Another type of information sharing is in the form of an inventory report containing the current stock level for each firm. This type of information sharing occurs if and only if both firms are willing to disclose their private inventory information.

For simplicity, there is no restriction on production capacity and the marginal production cost is a constant $C$ for both firms.
5.4.2.2 Best Response Function

**Theorem 5.4.1** In an iterative Cournot game described in the above supply chain model, when the time comes for both agents to make production decisions independently, let $q_1$ and $q_2$ denote the latest production strategy of agent 1 and agent 2 respectively, let $l_1$ and $l_2$ denote the current inventory level of agent 1 and agent 2 respectively, if both agents are aware of each other’s inventory status through information sharing and assume their opponents will keep their production strategies unchanged, the best response function $R^*_i$ of firm $i (i \in [1, 2])$ can be written as:

$$R^*_i = \begin{cases} 
A, & \text{if } B > A \geq 0; \\
B, & \text{if } A \geq B \geq 0; \\
0, & \text{otherwise.}
\end{cases}$$  \hspace{1cm} (5.4.4)

$$A = \frac{P_0}{\alpha} - l_i - l_j - q_j + M, (j \in [1, 2], j \neq i).$$  \hspace{1cm} (5.4.5)

$$B = \frac{P_0 - \alpha(2l_i + l_j + q_j - M) - C}{2\alpha}, (j \in [1, 2], j \neq i).$$  \hspace{1cm} (5.4.6)

where

- $\alpha$: price coefficient, $\alpha > 0$
- $P_0$: a constant, $P_0 > 0$
- $M$: mean value of random market demand, $M > 0$
- $C$: constant marginal cost, $C > 0$
Proof: Assume agent 1 increases its inventory level by producing $R_1$ products, let $\Pi_1$ denotes firm 1’s expected profits (payoff). Then firm 1’s payoff function can be written as follows:

$$\Pi_1 = \begin{cases} (R_1 + l_1)(K - C), & \text{if } K \geq 0; \\ -C(R_1 + l_1), & \text{otherwise.} \end{cases} \quad (5.4.7)$$

$$K = P_0 - \alpha(l_1 + l_2 + q_2 + R_1 - M). \quad (5.4.8)$$

Therefore, we have

$$\frac{\partial \Pi_1}{\partial R_1} =$$

$$\begin{cases} P_0 - C - \alpha(2R_1 + 2l_1 + l_2 + q_2 - M), & \text{if } R_1 \leq \frac{P_0}{\alpha} - l_1 - l_2 - q_2 + M; \\ -C, & \text{otherwise.} \end{cases} \quad (5.4.9)$$

Let $A = \frac{P_0}{\alpha} - l_1 - l_2 - q_2 + M$, then

$$\begin{cases} \frac{\partial \Pi_1}{\partial R_1} = 0 & \implies \begin{cases} R_1 = B \\ R_1 \leq A \\ \frac{B - \alpha(2l_1 + l_2 + q_2 - M) - C}{2\alpha} \end{cases} \quad (5.4.10) \end{cases}$$

First, we consider the case where $A < 0$. Since $R_1 \geq 0$, we obtain $R_1 > A$. According to (5.4.9), $\Pi_1$ is a linear decreasing function of $R_1$. So the best solution is $R_1 = 0$.

Next, we consider the case where $B < 0$. Since $R_1 \geq 0$, we obtain $R_1 > B$. According
to (5.4.7)(5.4.8)(5.4.9), $\Pi_1$ is a second-order decreasing function of $R_1$ when $R_1 \leq A$ or a linear decreasing function of $R_1$ when $R_1 > A$. In either case, the best solution is $R_1 = 0$.

We have just proven that the optimal solution is $R_1 = 0$ when $A < 0$ or $B < 0$.

Now, we deal with the case where $B > A \geq 0$. According to (5.4.7)(5.4.8)(5.4.9), $\Pi_1$ is a second-order increasing function of $R_1$ when $R_1 \leq A$ or a linear decreasing function of $R_1$ when $R_1 > A$. Therefore, the optimal solution is $R_1 = A$.

Finally, it is not difficult to prove that when $A \geq B \geq 0$, $\Pi_1$ is a linear decreasing function of $R_1$ (when $R_1 > A$) or a second-order function of $R_1$ with a maximum value at $R_1 = B$ (when $R_1 \leq A$). Therefore, the optimal solution in this case is $R_1 = B$.

We summarize by solving the following optimization problem:

$$\max_{0 \leq R_1 < \infty} (\Pi_1)$$

yields the best response $R_1^*$ of firm 1:

$$R_1^* = \begin{cases} A, & \text{if } B > A \geq 0; \\ B, & \text{if } A \geq B \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

$$A = \frac{P_0}{\alpha} - l_1 - l_2 - q_2 + M.$$  \hfill (5.4.13)

$$B = \frac{P_0 - \alpha(2l_1 + l_2 + q_2 - M) - C}{2\alpha}.$$  \hfill (5.4.14)

The best response $R_2^*$ of firm 2 can be obtained similarly.
5.4.2.3 Supply Chain Configurations

Simulating a complex chain is obviously a difficult endeavor. The enhanced TAC SCM game discussed in this chapter provides us with an ideal simulated supply chain with flexibility for our experimental study. To make the supply chain environment fully comply with the model described above, several configurations are made to the game:

1. The market price of additional assembly cells supplied by MES (Manufacturing Equipment Supplier, discussed in 5.3) is set to zero throughout the game so as to remove the restriction on agents’ production capacities.

2. The market price of any PC component (supplied by component suppliers) is set to half of its base price (specified in the TAC SCM configuration file) throughout the game so the average cost for producing a PC (i.e. the marginal cost) is approximately $1000.

3. The production capacities for all suppliers are set to be unlimited so as to minimize delivery delay.

4. The domain policies used by ISC (Information Sharing Coordinator, discussed in 5.4) in the simulation experiment are described in Table 5.1.

5. The customer reserve price function is:

\[ P_r(Q,M_d) = \]
Enhancements of the TAC SCM Game

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Domain 1</th>
<th>Domain 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Interest</td>
<td>Production</td>
<td>Inventory</td>
</tr>
<tr>
<td>Role(s) Allowed to Join</td>
<td>Level</td>
<td>Level</td>
</tr>
<tr>
<td>Compulsory to Join?</td>
<td>Agents</td>
<td>Agents</td>
</tr>
<tr>
<td>CSI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BSI</td>
<td>Daily</td>
<td>Daily</td>
</tr>
<tr>
<td>Sharing Frequency</td>
<td>Production</td>
<td>Inventory</td>
</tr>
<tr>
<td>Form of Sharing</td>
<td>Report</td>
<td>Report</td>
</tr>
</tbody>
</table>

Table 5.1: Domain policies used in the simulation experiment
(Agents: PC Assembly Agents)

\[
\begin{align*}
3000 - 0.2(Q - M_d), & \quad \text{if } Q - M_d < 15000; \\
0, & \quad \text{otherwise.}
\end{align*}
\]

where

- \( Pr \): Customer reserve price, \( Pr \geq 0 \)
- \( Q \): Aggregate quantity of PCs in both agents’ inventories, \( Q \geq 0 \)
- \( M_d \): Market demand, \( M_d \geq 0 \)

Since agents are only involved in Cournot (quantity) competition rather than Bertrand (price) competition, we can easily replace the customer reserve price \( Pr(Q, M_d) \) in the above equation with market price \( P_m(Q, M_d) \).

6. The average daily market demand is set to 300 for each PC type.

5.4.2.4 The Simulation Experiment and Results

For the purposes of brevity and to enhance clarity, we keep our explanation of the details of the simulation to a minimum.
Two groups of assembly agents are used in our simulation experiment. Each group consists of two competing agents. Agents in each group compete with their opponents in the same group in the above configured supply chain environment for 20 Cournot duopoly games with 365 simulated days for each game. In group 1, agent 1 and agent 2 keep sharing with each other their private inventory information throughout the course of the experiment. Both agent 1 and agent 2 implement the best response function (5.4.4). In group 2, on the contrary, agent 3 and agent 4 keep their own inventory information unveiled all the time. Both agent 3 and agent 4 adopt the best response function (5.4.4) as well, each assuming that its opponent stacks the same level of products in the inventory as it does.

Our intention is to see if the sharing of private inventory information has any significant impact to the supply chain. We use the following two performance measures:

- Total profits of both assembly agents
- Average market price per PC

After 30 runs of simulation (100 days each), under both circumstances of sharing and concealing private inventory information, we calculate the average total profits of the two competing agents and the average market price per PC. Table 5.2 summarizes the results.

The above simulation results indicate that sharing private inventory information

<table>
<thead>
<tr>
<th>Performance Measures</th>
<th>Share Inventory</th>
<th>Conceal Inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Profits for Agents</td>
<td>$190.44M</td>
<td>$48.45M</td>
</tr>
<tr>
<td>Average Price Per PC</td>
<td>$1145</td>
<td>$1072</td>
</tr>
</tbody>
</table>

Table 5.2: Sharing vs. concealing private inventory information
among competing agents benefits the agents in terms of total profits, but it also results in higher product price for the customer.

5.5 Discussion and Related Work

We believe that our enhancements will not only make the TAC SCM games more interesting, but more importantly, will make the emerging techniques for trading agent design and e-Marketplace development more applicable to real-world e-commerce. However, more experiments need to be done to investigate the impact of information sharing and agents’ cooperation, as well as to enrich the proposed information sharing mechanism which has limited scope at this moment. This will be a direction of the author’s future research work.

Our enhancements are inspired by the intensive discussions that have occurred in the TAC SCM forum over the last two years, mainly due to the serious problem of lottery effects in the component market. Many suggestions and great ideas dealing with this issue have been integrated in the draft proposal of the TAC SCM-05 specification, which was recently released on the TAC Website for discussion[23]. Similar to our component market model, the proposed specification introduced an auction mechanism into the component market by allowing agents to set a reserve price. The advantage of this approach is that the old three-round interaction model can be kept without any change. Nevertheless, much uncertainty still exists for agents in regard to component price and availability. Another similarity between our proposal of the component market model and the new specification is that both models reserve some
capacity for long-term commitments. A linear reduction function is introduced to limit committed contracts. No formal analysis, however, has been done for either the original specifications or the new specification for the component market.

5.6 Summary

This chapter first discussed several enhancements of the recent supply chain management trading agent competition (TAC SCM) game. We presented a new multi-agent based supply chain model that has the following characteristics:

- A customer model with linear reserve price function
- Extendable production capacities of assembly agents
- An information sharing mechanism

We then explored the impact of sharing private inventory information among supply chain competitors, using a simulation approach. We based our simulation study on the enhanced supply chain model. Our experimental results show that sharing private inventory information results in increased total profits of the upstream manufacturers, and a higher product price for downstream customers, in a two-tier Cournot duopoly market.

The primary contribution of this chapter, however, is not the conclusions we draw from our experiment addressing a specific information sharing instance under our particular supply chain configurations. Rather, we view the enhanced supply chain model we built, based on the original TAC SCM scenario, as a useful and flexible
research tool and the simulation experiment as an illustration for trading agent developers and supply chain designers to reason with real and challenging issues in a complex supply chain environment.
Supply Chain Management (SCM) deals with the planning and coordination of product marketing, production/delivery scheduling and resource procuring activities across the multiple organizations involved in the delivery of goods to end-users. With the emergence of e-Marketplaces, it is only natural to seek automated solutions that are capable of responsively coping with dynamic market conditions and rapidly evaluating and re-evaluating a large number of bidding and procurement options in real time, while taking into account strategies employed by competitors. This makes it imperative to use automated programs (or software trading agents) to perform and support SCM.

This thesis considered the problems posed by e-Marketplace development and the design of intelligent trading agents for supply chain management in e-Marketplaces. Those problems are interesting because good solutions are thought to require a synthesis of several fields: industrial operations research, economics, game theory, artificial intelligence, machine learning, multi-agent systems, computational complexity theory, statistics, and probability.

The primary contribution of this thesis is its detailed analysis of the characteristics and properties of a typical e-Marketplace environment, together with in-depth
Conclusion and Future Work

exploration of optimal trading agent strategies and extensive discussion of possible enhancements of state-of-the-art e-Marketplace architecture.

The recent Trading Agent Competition for Supply Chain Management (TAC SCM) game scenario was chosen as a representation of a typical e-Marketplace and the basis of this research. To the best of our knowledge, TAC SCM is the only competition-based e-Marketplace simulation that captures the challenges present in an integrated procurement, production and customer-bidding environment. An instance of this game involves six software trading agents attempting to maximize profits by manufacturing computers in a simulated three-echelon market economy. To be successful, an agent needs to perform the following highly inter-related tasks on each simulated day, taking into account its limited production capacity: (1) choose RFQs (Request for Quotes) and bid for customer orders; (2) negotiate supply contracts for components; (3) schedule daily assembly activities and deliveries; and (4) manage storage.

Following a concise description of the TAC SCM game in chapter 1, we presented in chapter 2 the theoretical models of TAC SCM, based on existing economic models. Since the downstream market in TAC SCM (i.e. the product market) is characterized by the combination of quantity competition and price competition between manufacturers (agents), we specified the quantity competition by a variation of the Cournot model and viewed the price competition as an extension of the Bertrand game. Because the market supply is fixed with only a small reverting random walk, we specified the upstream market with a constant-supply model. In chapter 3, we presented the architecture and strategies of our “Jackaroo” agent, one of the most influential agents in the TAC SCM tournaments. Most of the strategies employed in the “Jacka-
roo” agent are rooted in theoretical models that we discussed in chapter 2 and statistical analysis of the TAC SCM games. We outlined the basic theory and algorithms dealing with essential decision-making problems including component procuring, product pricing, production scheduling and price forecasting. We presented experimental results that indicate the correctness and effectiveness of our modelling approaches. In chapter 4, we presented a new negotiation mechanism for the TAC SCM component market and a new supplier model to demonstrate how negotiation strategies can be applied in automated negotiation. With the proposed negotiation mechanism, both sellers and buyers are able to operate the component market autonomously. The new supplier model simulates the operation of modern enterprises in more aspects than in the original proposal. In chapter 5, we explored some critical limitations which greatly impedes the TAC SCM from being widely used as a flexible research tool for supply chain designers and trading agent developers. We proposed three enhancements for the TAC SCM e-Marketplace mechanism: (1) a customer behavior model with linear reserve price function; (2) extendible production capacity of manufacturing Agents and (3) an information sharing mechanism.

One potential area for long-term future work is the development of a generic e-Marketplace framework with a robust market mechanism and efficient negotiation protocols. Another related future work is the development of a generic trading agent architecture that will fit in the generic e-Marketplace. We hope an e-Commerce development package consisting of the above two components can be used to create highly-sophisticated e-Marketplaces for any specific market types as well as facilitate e-Trading.
In one and a half years of study, I have written a total of five academic papers, each published by international journals and conferences. Two of these were written solely by me while the rest were co-authored by others. Among them, the paper “A Multi-Agent Mediated Simulation Study of Information Sharing Based on An Enhanced TAC SCM Game” received the Best Paper Award from the ACS-CISIM 2004 International Conference.


4. Kanghua Zhao, A Multi-Agent Based Supply Chain Simulation Environment.

Bibliography


[16] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Wolley, KAoS: Toward an Industrial-


http://www.sics.se/tac/.


[72] M. P. Wellman, P. R. Wurman, K. O’Malley, R. Bangera, S. Lin, D. Reeves and W.


