Master's Thesis Economics & Informatics

Product Pricing using Adaptive Real-Time Regime-Based Acceptance Probability Estimates

Author:
Alexander Hogenboom, BSc
287485ah@student.eur.nl

Supervisors:
Prof. dr. ir. Uzay Kaymak
kaymak@ese.eur.nl

Wolfgang Ketter, Ph.D.
wketter@rsm.nl

Dr. Jan van Dalen
jdalen@rsm.nl

Computational Economics programme
Econometric Institute
Erasmus School of Economics
Erasmus University Rotterdam
February 14, 2009

Abstract

In a complex supply chain, in which several traders compete in a component procurement market as well as in a sales market where assembled products are sold through reverse auctions, product pricing is a vital, yet non-trivial task. In this thesis, a product pricing approach using adaptive real-time regime-based probability of acceptance estimations is proposed. Based on economic regime estimations, price distributions are approximated, which are adapted using relevant available information on prices and characteristics of customer requests for quotes. Artificial neural networks are trained to act as adapter and estimate parameters for the double-bounded log-logistic function assumed to be underlying the prices. This adaptation differs per market condition and is corrected using an error factor, which is updated on-line. Given the parametric approximation of the price distribution, the probability of acceptance is estimated using a closed-form mathematical expression. This expression can then be used to determine the price yielding a desired quota. The approach is implemented in the MinneTAC trading agent and tested against a price-following product pricing approach in the TAC SCM game. The new product pricing approach yields a significant performance improvement; more orders are obtained against higher prices. Profits are more than doubled.

Acknowledgements

The support of several people has greatly contributed to the realization of this thesis. In the first place, I would like to thank my supervisors for putting so much time and effort into supervising this project. While working on this project, I experienced a steep learning curve, but my supervisors have been really helpful by guiding me in my research; each in his own way.

I would like to thank Uzay Kaymak for always keeping the bigger picture in mind and hereby helping me sorting out my priorities. His feedback has always been really helpful as well. I would also like to thank Wolf Ketter for introducing me to the TAC SCM game and the ongoing research on the MinneTAC agent developed for this game, as working on this type of project has been an interesting experience. Wolf Ketter's seemingly endless flow of new ideas has been inspiring as well. Furthermore, I would like to thank Jan van Dalen for deepening my statistical knowledge and for really taking the time to explain statistical methods and techniques new to me. I also truly appreciate the valuable insights with respect to the project he shared during our meetings.

Also, I am very grateful to John Collins from the University of Minnesota for giving me the opportunity to use the MinneTAC agent as well as the systems at the University of Minnesota for my research. Without his support, I would not have been able to run all experiments.

Finally, I would like to thank my family and friends for supporting me and being patient with me, even though I tended to get completely consumed with my thesis from time to time. I really appreciate the support and the constructive feedback they gave me any time, day or night.

Contents

1 Introduction **1**
1.1 Background . . . 2
1.1.1 The TAC SCM Game . . . 2
1.1.2 The MinneTAC Trading Agent . . . 2
1.2 Goal . . . 4
1.3 Research Questions . . . 4
1.4 Methodology . . . 5
1.5 Thesis Structure . . . 6

2 Pricing Decisions in the TAC SCM Game **7**
2.1 Product Pricing . . . 7
2.1.1 Modeling the Probability of Acceptance . . . 7
2.1.2 Accounting for Other Aspects . . . 8
2.2 Details on the TAC SCM Game . . . 9
2.2.1 Game Constraints . . . 9
2.2.2 Customers . . . 10
2.2.3 Traders . . . 10
2.2.4 Suppliers . . . 12
2.3 Making Sales Decisions . . . 12
2.3.1 TAC SCM Sales Strategies . . . 13
2.3.2 The MinneTAC Approach . . . 14
2.4 Summary . . . 17

3 Realization of Product Prices in the TAC SCM Game **19**
3.1 From Requests to Orders . . . 19
3.2 Price Distributions in the TAC SCM Game . . . 21
3.3 Utilizing Price Distributions in Estimations of Customer Offer Acceptance Probabilities . . . 23
3.3.1 General Framework . . . 25
3.3.2 Incorporating a Log-Logistic Distribution . . . 27
3.4 Summary . . . 30

4 Towards an On-Line Product Pricing Approach **31**
4.1 Artificial Neural Networks 33
4.2 Incorporating Artificial Neural Networks into the Framework 34
4.3 Identifying Potentially Good Inputs 36
4.4 Summary 37

5 Introducing Adaptivity **39**
5.1 Coping With Market Disruptions 39
5.2 Feeding Back Market Responses 41
5.3 Summary 43

6 Adaptive Regime-Based TAC SCM Product Pricing **45**
6.1 Training the Artificial Neural Networks 45
6.2 Implementation in the MinneTAC Agent 47
6.3 Performance in the TAC SCM Game 48
6.3.1 Experimental Set-Up 49
6.3.2 Experimental Results 50
6.4 Summary 52

7 Discussion **55**

8 Conclusions **59**

A Detailed Experimental Results **67**

List of Figures

2.1 Simplified schematic overview of the current MinneTAC sales process configuration for an arbitrary product on an arbitrary game day. 16

3.1 Distributions of the number of RFQs, offers, minimum offers, and orders per product per day, aggregated over all games in the training set. 20

3.2 Normalized mean order prices for product 1, 3, and 7 (low-range, mid-range, and high-range market segment products, respectively) over time in a typical TAC SCM game (7308tac3) are depicted in (a), whereas (b) visualizes normalized mean order prices for product 1, 2, 9, 10, and 11 (low-range market segment products) over time in this game. 22

3.3 Distribution of normalized offer prices, minimum offer prices, valid offer prices, and order prices for product 14 on day 28 of a typical TAC SCM game (7308tac3). 24

3.4 Standard deviation of normalized offer prices, minimum offer prices, valid offer prices, and order prices for product 14 over time in a typical TAC SCM game (7308tac3). 24

4.1 Distributions of the found α and γ parameter values for daily distributions of valid offer prices (depicted in (a) and (b) respectively), the associated mean number of offers per RFQ $\overline{n}_{dg}$ for each day d and product g (depicted in (c)), and the associated p-value for the Kolmogorov-Smirnov test (depicted in (d)), aggregated over all games in the training set. 35

6.1 Simplified schematic overview of the proposed PPARRPAE sales process configuration for an arbitrary product on an arbitrary game day. 48

6.2 Final bank account balances for the two considered MinneTAC variants over all experiments. The separate values for each experiment are depicted in (a), whereas (b) shows final bank account balance distributions. 51

6.3 Increase in profit of the PPARRPAE approach compared to the benchmark, depicted in relation to the associated relative increase in the number of obtained orders per experiment. . . 51

List of Tables

1 Summary of notation. ix

1 Summary of notation, continued. x

2.1 Specification of all PC types considered in the TAC SCM game. 11

2.2 Specification of all components considered in the TAC SCM game. 11

3.1 Statistics on the probabilities of a distribution of order prices to be constant within a window (sizes ranging from 2 to 10) for product 1, 3, and 7 (low-range, mid-range, and high-range market segment products, respectively) for all days in all games in the training set. 23

4.1 Statistics on the found α and γ parameter values for daily distributions of valid offer prices, the associated mean number of offers per RFQ $\overline{n}_{dg}$ for day d for product g, and the associated p-value for the Kolmogorov-Smirnov test, aggregated over all games in the training set. 35

4.2 Correlations of found values for α and the natural logarithm of γ for daily distributions of valid offer prices and their associated mean number of offers per RFQ $\overline{n}_{dg}$ with on-line available variables for all products for all days in all games in the training set. 38

6.1 Optimized configuration of number of clusters and minimum standard deviation of clusters for radial basis function networks for estimating the α parameter, the natural logarithm of the γ parameter, and the $\overline{n}_{dg}$ parameter, given a dominant regime k. The table also quantifies the root mean squared deviation (RMSD) of parameter values predicted by the trained models from the parameter values in the test set. 46

6.2 Mean and standard deviation of final bank account balance per agent, calculated over all experiments. Values are expressed in millions. 50

6.3 Mean and standard deviation of number of obtained orders per agent, calculated over all experiments. Values are expressed in thousands. 50
6.4 Overall deviation of values for final bank account balance and number of obtained orders of the PPARRPAE approach, compared to the benchmark. Balance deviation is expressed in millions, whereas the deviation in the number of obtained orders is expressed in thousands. 50
6.5 Overall deviation of values for the root mean squared deviation of estimated probability of acceptance (RMSD PA) and the number of market polls of the PPARRPAE approach, compared to the benchmark. 52

A.1 Statistics on the number of RFQs, offers, minimum offers, and orders per product per day: mean and standard deviation (Stdev) per game, as well as over all games in the training set aggregated. 67
A.2 Statistics on the number of offers and the extent to which these offers are invalid. This table also quantifies the root mean squared deviation (RMSD) from the normalized reservation prices for both valid and invalid normalized offer prices. These statistics are calculated per game, as well as over all games in the training set aggregated. 68
A.3 Statistics on the number of minimum offers and the extent to which these minimum offers are invalid. This table also quantifies the root mean squared deviation (RMSD) from the normalized reservation prices for valid and invalid normalized minimum offer prices. These statistics are calculated per game, as well as over all games in the training set aggregated. 69
A.4 Final bank account balances per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. Balance values are expressed in millions. 70
A.5 Number of orders obtained per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. Values for the number of obtained orders are expressed in thousands. 71
A.6 Root mean squared deviation of estimated probability of acceptance per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. 72
A.7 Number of market polls per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. 73

List of Symbols

Table 1 provides an overview of mathematical notation used in this thesis. Symbols used in equations throughout this thesis are briefly explained.

Symbol	Definition
α	Parameter of the double-bounded log-logistic distribution, representing the median offer price
β	Smoothing factor for double exponential smoothing
γ	Parameter of the double-bounded log-logistic distribution, quantifying the distribution tightness
ϵ	Error term accounting for the ratio between received and predicted orders
$\widetilde{\epsilon}$	Smoothed error term accounting for the ratio between received and predicted orders
ϖ	Number of parts needed to make a product
φ	Shape parameter of the log-logistic distribution
χ	Shape parameter of the log-logistic distribution
c^{man}	Manufacturing cost of a product
c^{nom}	Nominal cost of a part of a product
$d\left(x_h, x\right)$	Distance between kernel center x_h and an instance x
$f\left(p; \theta\right)$	Distribution of normalized valid offer price p, with θ a vector of parameters
$F\left(p; \theta\right)$	Cumulative distribution of normalized valid offer price p, with θ a vector of parameters
$f_{\underline{p}}\left(\underline{p}; \theta\right)$	Distribution of normalized minimum valid offer price $\underline{p}$, with θ a vector of parameters
$F_{\underline{p}}\left(\underline{p}; \theta\right)$	Cumulative distribution of normalized minimum valid offer price $\underline{p}$, with θ a vector of parameters
$f\left(x\right)$	Function of x
$\hat{f}\left(x\right)$	Approximation of the function of x
$K_h\left(d\left(x_h, x\right)\right)$	Kernel function of hidden unit h used in an RBFN, depending on the distance between the kernel center x_h and an instance x

Table 1: Summary of notation.

Symbol	Definition
$L(\theta; \vec{p})$	Negated log-likelihood function of parameters in vector θ for a sample of prices $\vec{p}$
m	Number of RFQs issued
n	Number of normalized valid offer prices
$\overline{n}$	Mean number of offers per RFQ
p	Normalized valid offer price
p^*	Optimal offer price, expected to yield sales quota
$p^{*'}$	Corrected optimal offer price
$p^{\min}$	Normalized minimum order price realized over all RFQs of a product on a game day
$p^{\max}$	Normalized maximum order price realized over all RFQs of a product on a game day
p^{real}	Real price for a product
$\widetilde{p}$	Smoothed normalized order price realized over all RFQs of a product on a game day
$\widetilde{p}^{\min}$	Smoothed normalized minimum order price realized over all RFQs of a product on a game day
$\widetilde{p}^{\max}$	Smoothed normalized maximum order price realized over all RFQs of a product on a game day
$\underline{p}$	Normalized minimum valid offer price for an RFQ (order price)
$P(o\|p)$	Probability that a customer will place an order with an agent, given its offer price p
$P(o\|p)'$	Corrected probability that a customer will place an order with an agent, given its offer price p
$P\left(o\|p^{*'}\right)$	Expected acceptance probabilities associated with corrected optimal offer price $p^{*'}$
$P(R_k)$	Probability of regime R_k
q	Proportion of actually received orders
q^*	Sales quota (desired obtained proportion of orders)
u	Upperbound of the double-bounded log-logistic distribution
w	Weight of activation unit in an RBFN
x	Benchmark value
$\hat{x}$	Approximation of x

Table 1: Summary of notation, continued.

Chapter 1

Introduction

In today's global economy, supply chains are everywhere. A supply chain is a complex logistics system in which raw materials are converted into finished products and then distributed to the final users (consumers or companies). It includes suppliers, manufacturing centers, warehouses, distribution centers and retail outlets [9]. The main idea of supply chains is that every entity within a chain adds value to the final product and fulfills a function within the chain. It is possible to optimize individual elements within the chain, but it is also possible to optimize the supply chain as a whole, which may yield sub-optimal performance of individuals.

Effective Supply Chain Management (SCM), focussing on more flexible and dynamic relationships between entities in the supply chain, is vital to the competitiveness of traders within this chain. SCM can yield this effect, as it enables these traders to respond to changing market demands in a timely and cost effective manner [6]. Hence, research into performance optimization in a supply chain is important for the profit maximizing companies of today.

A very basic supply chain with suppliers, traders and customers is simulated in the TAC SCM game, an international competition for designing trading agents for an imaginary simulated personal computer (PC) supply chain. The main goal of this game is to optimize the profit an agent generates, and not optimizing the generated profit of the supply chain as a whole. A lot of research has been done on both the TAC SCM game and its participating agents, as analysis of market conditions and behavior exhibited by agents in a controlled environment could provide valuable insights in the consequences of behavior of entities in a supply chain. The associated software engineering challenges contribute to the importance and appeal of this field of research. The focus of this thesis is on the MinneTAC agent, a trading agent created by the University of Minnesota.

This section continues with a quick overview of the TAC SCM game and the MinneTAC agent considered in this research in Section 1.1. The goal of this research is defined in Section 1.2 and the methodology used for achieving this goal is described in Section 1.4. Finally, Section 1.5 presents an overview of the structure of this thesis.

1.1 Background

This section presents the TAC SCM game (see Section 1.1.1), as well as the the MinneTAC agent for this game (see Section 1.1.2) in a nutshell. This overview should provide a proper initial understanding of the environment considered in this thesis.

1.1.1 The TAC SCM Game

Since 2002, the Trading Agent Competition for Supply Chain Management (TAC SCM) game has been organized in order to promote and encourage high quality research into trading agents in supply chain environments. The TAC SCM game stimulates research with respect to more flexible and dynamic supply chain practices as opposed to the current common practices where supply chains are essentially static and rely on long-term relationships among key trading partners [6].

In the TAC SCM game, a supply chain for PCs is considered. This supply chain consists of customers, traders and suppliers. The 16 PC types available through this supply chain can be classified into a high-range, mid-range, and low-range market segment. Every game day, customers issue requests for quotes (RFQs) for finished computers, on which traders can bid. The requested products are subsequently assembled by the traders using components procured from suppliers. These trading agents are developed in the context of the TAC SCM game and they all try to maximize their profit over a game. The major challenge to this respect is the limited visibility of the market environment; little data is available on-line to an agent.

1.1.2 The MinneTAC Trading Agent

The University of Minnesota competes in the TAC SCM game with their MinneTAC trading agent [7]. For each individual market segment, the MinneTAC agent uses historical data as well as observable current data to characterize microeconomic conditions in order to guide tactical (e.g. product pricing) as well as strategic decision making processes (e.g. product mix and production planning) [15, 16, 17].

In this context, the distribution of prices in past games has been approximated using a Gaussian Mixture Model (GMM). Clustering the probabilities represented by these individual Gaussians (using the k-means algorithm [21]) yielded distinguishable statistical patterns, which are referred to as economic regimes (e.g. scarcity, a balanced situation, and oversupply).

Now, when on a particular day in a game the (expected) price of that day is used as input for the GMM, the individual Gaussians are all activated to a certain extent, which yields an approximation of the price distribution of that day. Subsequently calculating each cluster's normalized price density enables the regime probabilities for an arbitrary day, given the (expected) price, to be determined.

Typically, on an arbitrary day, the regime probabilities for that day are approximated using an exponentially smoothed prediction of the price of that day. The agent is not only able to identify the current regime and act accordingly, but the agent's behavior depends on expected future regimes as well. With respect to regime prediction, for short-term prediction for tactical decision making, the agent uses a Markov prediction process based on the last normalized smoothed mid-range price, hereby utilizing Markov transition matrices created off-line by a counting process over past games. For long-term prediction for strategic decision making, the MinneTAC agent uses a Markov correction-prediction process, which is similar to the Markov prediction process, except for that its predictions are based on all normalized smoothed mid-range prices up until the previous day. An alternative for the Markov prediction processes is regime prediction based on exponentially smoothed price predictions which take into account exponentially smoothed price trends.

The MinneTAC trading agent is founded on two pieces of software: the Apache Excalibur component framework [1], which enables components to be constructed for and used in configurations of working agents, and the AgentWare package distributed by the TAC SCM game organizers, which handles interaction with the game server. The usage of Excalibur yields a loosely coupled architecture of the agent, which facilitates easier development and maintenance. Furthermore, the architecture increases the flexibility of the MinneTAC agent in practice, as components can be used any time, hereby enabling the agent to be configurable in many ways.

All decision making processes in the agent serve one single goal: to maximize the overall profit generated by this agent in a game. With respect to tactical decisions related to sales, identified probabilities for current or future economic regimes are used in order to determine daily sales quotas. Using a configurable chain of evaluators [7], enabled by the Excalibur framework, the price likely to be accepted by the desired proportion of the customers (the sales target) is estimated. In order to compensate for the uncertainty in the generated predictions, a slight variability is added to the offered prices by generating prices in an interval around the determined price (interval randomization). The inputs for this product pricing process (a sales quota and a function representing the probability of customers to accept an offer) depend on the evaluators used. The software is designed to select the right configuration of evaluators in the right situations.

1.2 Goal

The goal of this thesis is to improve the product pricing process of the MinneTAC agent in order to maximize profits. In the current approach, the curve representing the probability of customers to accept an offer for a specified price on an arbitrary game day is approximated using an estimation of the median price and slope of the curve in the median of that day. This acceptance probability is subsequently used for determining the price to be offered in order for the sales agent to sell its desired quota for a specific product. This approach implies the loss of possibly crucial data on the range and distribution of prices in the observed time frame, whereas these characteristics vary per economic regime.

The regime model implemented in the MinneTAC agent can be used to predict prices, price trends, and price distributions for a given horizon. However, these price distribution estimations are not exploited yet in the product pricing process, as these distributions tend to be fairly static and general, because no factors other than a mean price estimate are accounted for. In order for the regime model's price distributions to be useful in the daily product pricing process, these price distributions need to be adapted to better approximate the real price distribution by incorporating additional information. An on-line update mechanism accounting for market responses to product pricing behavior resulting from the model is crucial as well.

Hence, the focus in this thesis is on the estimation of an arbitrary day's price distribution – given that day's regime probabilities – and subsequently determining the customer offer acceptance probabilities using this distribution, in order to improve MinneTAC's product pricing process. The main concern in this respect is how to incorporate additional information, such as information on leadtimes associated with relevant RFQs, and feedback through market responses in the process.

1.3 Research Questions

The main question to be answered in this thesis is:

How can estimating price distributions using economic regimes in the context of product pricing contribute to profit maximization?

In order to be able to properly answer this question, some sub questions need to be answered. The main question of this research can be divided into the following sub questions:

1. How can the relation between price distributions and the daily product pricing process be modeled?

2. In what way can daily product pricing using regime-based price distribution estimations be applied in real-time?

3. How can price distribution estimation, and hence product pricing, be adapted to dynamic market characteristics?

1.4 Methodology

In order to answer the questions posed in Section 1.3 and achieve the objective defined in Section 1.2, the methodology as discussed here is used.

First of all, the application of regimes in related work, as well as acceptance probability estimation and product pricing strategies should be looked into. Furthermore, the specifications of the TAC SCM game as well as the characteristics of the MinneTAC trading agent considered in this thesis are to be looked into. This should provide insight in constraints posed by the environment the agent considered in this thesis is designed for, as well as in the processes this research aims to improve. The literature survey should especially cover the economic regime identification process as well as the product pricing process used in the MinneTAC trading agent.

Then, historical game data should be analyzed in order to determine how exactly product pricing can be related to price distributions. Because the focus in this thesis is on adapting regime-based price distributions in order for them to be useful in the daily product pricing process, a subset of the historical game data, containing data available on-line to the agent, is to be considered in the subsequent translation into an on-line applicable real-time daily product pricing approach.

Furthermore, the proposed improvements to MinneTAC's process of product pricing are to be assessed. To this end, the quality of price distribution estimations can be evaluated using historical game data in the first place. However, the improvements should be tested in real game situations as well. Therefore, the proposed improvements are to be implemented in Java, in order to assess the in-game performance of these adaptations, compared to the approach currently used. In a controlled environment, the overall profit the agent generates in a game using the product pricing approach proposed in this thesis can then be compared to the profit generated when using the current approach. The extent to which sales targets are realized could also be assessed.

For each TAC SCM game run, a game server logs a vast amount of data, including data on sales, procurement, and so on. The dataset to be used in this research concerns the sales part of the agent, i.e. it only contains information about transactions between customers and agents and some additional (banking) information about these agents. The dataset is extracted from game data of the TAC SCM 2007 Semi-Finals and Finals games run on the tac5 [32] (games 9321 through 9328) and tac3 [31] (games

7306 through 7313) servers respectively. The dataset also contains game data extracted from the TAC SCM 2008 Semi-Finals and Finals run on the tac02 [41] (games 761 through 769) and tac01 [40] (games 792 through 800) servers respectively. The test set considered in this research consists of game data of the first two games and the last game of each set of games run on a server. All other data is part of a training set, which is used for analysis purposes.

1.5 Thesis Structure

This thesis is structured as follows. In Chapter 2, the application of regimes in general, acceptance probability estimation, and product pricing strategies are discussed, as well as the TAC SCM game specifications and the sales component of the MinneTAC agent. The connection of price distributions to the product pricing process in the TAC SCM game is modeled in Chapter 3. This model is translated into an on-line product pricing approach in Chapter 4, whereas adaptivity is introduced in Chapter 5. The performance of the proposed approach is evaluated in Chapter 6. Chapter 7 discusses the research results. Finally, conclusions are drawn in Chapter 8.

Chapter 2

Pricing Decisions in the TAC SCM Game

Before an approach can be proposed for adapting regime-based price distribution estimations in order for them to be applicable in a TAC SCM product pricing process, product pricing in general is to be looked into. This is done in Section 2.1. Because the newly proposed product pricing approach is to be developed for and tested in the TAC SCM game, Sections 2.2 and 2.3 provide insight in the characteristics of this game and TAC SCM sales strategies. Section 2.4 summarizes the findings.

2.1 Product Pricing

An approximation of the probability of acceptance of offers can be used in the product pricing process, as proposed in [8]. The proposed analysis of offer prices and their associated estimated probabilities of acceptance is rather intuitive in a product pricing process, because this can help a seller assessing how sales targets can be met. Therefore, Section 2.1.1 discusses how these acceptance probabilities can be approximated. However, more aspects can be taken into account when pricing products. This is discussed in Section 2.1.2.

2.1.1 Modeling the Probability of Acceptance

In [8], a competitive economy is considered, in which a product is differentiated on multiple attributes. The environment is constrained by limited visibility; sellers only have limited knowledge about market parameters. Products are priced using a dynamic pricing algorithm which considers an estimated distribution of the buyer reservation price for products of a seller. Inverting the cumulative form of this distribution yields a function expressing the proportion of buyers willing to pay the seller a specified price, which can also be interpreted as the probability that a customer accepts an offered price. This function can subsequently be used to determine the price expected to yield a specified sales target (expressed as the desired proportion of customers accepting the offered price).

The distribution underlying customer offer acceptance probabilities considered in [8] depends on market characteristics such as competing sellers' prices and buyers' purchase preferences. In the observed time frame, these characteristics are assumed to be constant. The distribution parameters are unknown and should be estimated. After each interval, the distribution parameters are updated, hereby taking into account the extent to which the sales target has been fulfilled.

Another way of modeling acceptance probabilities is by using linear regression on data points representing recent prices offered, along with the resulting acceptance rate [29]. The distributions of customer acceptance probabilities could also be trained off-line [4]. Another option is to try to model the decision function of the accepting entities, based on their decision histories, using for example Chebychev polynomials [30].

2.1.2 Accounting for Other Aspects

Related work on product pricing suggests some other aspects beside straightforward acceptance probability estimations as discussed in Section 2.1.1 to be taken into account as well when pricing products. For instance, current and/or future offers of other suppliers (outside options) could be considered [20, 19]. In [35], outside options are considered as well. Here, a dynamic product pricing model is proposed in which the price change of the product itself as well as the relative price of competing products is quantified in a price elasticity. Using scenario analysis for distinguishing between various situations of price elasticity, the pricing policy expected to maximize revenue over the sales horizon is selected using a genetic algorithm. These results indicate that relating product pricing strategies to market conditions can be useful for profit maximization.

When determining an optimal product pricing strategy, the effect of the product prices over time inherent to this strategy could also be taken into account. Through expensiveness, as perceived by the customers, price history tends to influence current demand. Higher past prices yield lower perceived current expensiveness, which results in higher demand [33].

Another product pricing approach taking into consideration customer valuations is proposed in [42]. In the context of English auctions, the goal in this approach is to determine the optimal reservation price a seller should set, given an unknown distribution of private values of the bidding parties. The proposed framework is based on the idea that bids occurred in an auction are not equal to, yet related to each bidder's private value for the auctioned good.

Product pricing strategies can influence overall profits in another way: the impact on market shares differs per type of price change [36]. This indicates that when pricing products, the (indirect) impact on overall profits should be accounted for; temporary price changes cause temporary market share changes, whereas structural changes in prices as well as evolving prices have permanent effects on market shares and hence profits. Firms should also incorporate anticipations of their competitors' reactions to their pricing policy in the process of determining the optimal product pricing policy.

Finally, in case of known demand and uncertain supply, a responsive pricing policy, in which the retail price is determined after observing the realized supply yield, is shown to result in a higher expected profit than a pricing policy in which the realized supply yield is not taken into account [38]. This indicates, that modeling (expected or observed) supply-side behavior in the product pricing process could contribute to profit maximization.

2.2 Details on the TAC SCM Game

As stated in Section 1.2, this thesis specifically focuses on the product pricing process in the context of the TAC SCM game. This game considers a supply chain for PCs, which consists of customers, traders and suppliers. The 16 PC types available through this supply chain can be classified into a high-range, mid-range, and low-range market segment and vary in components used. Each product consists of four components, each of which has multiple variants and suppliers. In the TAC SCM game, six traders compete with each other in a sales market for customers and in a procurement market (consisting of eight suppliers) for computer components, the latter of which are used in assembling the PCs requested by the customers. The agent with the highest bank account balance at the end of the game wins the game.

2.2.1 Game Constraints

A typical TAC SCM game has a runtime of 220 days, where each day has a duration of 15 seconds. Suppliers and customers are simulated by the game, but the traders are software agents to be developed by the competing teams. During a TAC SCM game, human intervention is not allowed. Banking, production and warehousing services are provided.

Some costs in the game can be associated with the banking, production, and warehousing services. Inventory storage costs are randomly chosen at the beginning of the game, are equal for all agents, and remain fixed throughout the game. As each trader starts with no inventory and an empty bank account, traders are likely to be needing to loan money for creating some initial inventory. Therefore, the trader's bank account balance may be negative, in exchange for interest, charged on a daily basis. Positive bank account balances however are rewarded with daily savings interest being paid.

For each TAC SCM game run, a game server logs a vast amount of data, including data on sales, procurement, and so on. However, not all data is publicly available to the trading agents during a TAC SCM game, as this would be rather unrealistic. In reality, a detailed and complete overview of the market is nearly impossible. On-line available information is the only information visible to a trading agent during a game and thus the only information which can be directly incorporated into all kinds of decision making processes. Other data could of course be approximated.

The data always available to the agent during a game consists of the received RFQs and orders of the agent itself (up until the current game day), the preceding day's minimum and maximum price of each PC type, and market reports issued every 20 game days [6]. These market reports include four component type supply reports containing information about the aggregate quantities shipped by all suppliers, the aggregate quantities ordered from all suppliers, and the mean price per stock keeping unit (SKU) for all component types (CPU, memory, hard disk, and motherboard) ordered. Market reports also contain information on supplier production capacity and customer demand data, including request volume, order volume, and average price per PC type.

2.2.2 Customers

Every game day, customers issue RFQs for finished computers, on which traders can bid. In these RFQs, a randomly chosen type of computer, quantity, due date, reservation price, and a randomly chosen penalty for late delivery are specified. Subsequently, customer demand is generated by customers selecting from quotes thus submitted by traders. The customer demand per market segment is in fact drawn from a Poisson distribution, which has a reverting random walk as its input.

2.2.3 Traders

The requested products are subsequently assembled by the traders using components procured from suppliers. Hereby, the four components of each assembled PC are procured from eight suppliers. Table 2.1 [6] presents an overview of all considered PC types and the components each type consists of. These components are further specified in Table 2.2 [6]. A trading agent may send up to five RFQs per day per supplier for each of the two products offered by that supplier. In case of an order, a trader is immediately billed for a portion of its order's costs.

Traders are all endowed with an identical factory with a limited-capacity assembly cell, in which any type of PC can be assembled. Furthermore, each trader is equipped with an identical warehouse, in which components as well as finished PCs can be stored. In order to properly operate this factory, a trader has to create daily production schedules for determining how to allocate available component inventory and factory capacity to the production of PCs. On a daily basis, a trading agent must also decide which RFQs to bid on, which components to (attempt to) procure, and which supplier offers to accept. Finally, trading agents must create daily delivery schedules. These schedules define which assembled PCs are to be shipped to which customers.

Orders are due to be delivered on the date specified in the negotiated contracts, hereby taking into account that deliveries by definition take up one day. Late deliveries lead to penalties and even order cancellations in case deliveries are more than five days late. All pending orders after the last game day are charged the remaining penalty as well. All these tasks should be performed while serving one single goal: to maximize final bank account balance over a game. The major challenge in this respect is the limited visibility of the market environment; little data is available on-line to a trading agent.

Product id	Market segment	Components
1	Low-range	100, 200, 300, 400
2	Low-range	100, 200, 300, 401
3	Mid-range	100, 200, 301, 400
4	Mid-range	100, 200, 301, 401
5	Mid-range	101, 200, 300, 400
6	High-range	101, 200, 300, 401
7	High-range	101, 200, 301, 400
8	High-range	101, 200, 301, 401
9	Low-range	110, 210, 300, 400
10	Low-range	110, 210, 300, 401
11	Low-range	110, 210, 301, 400
12	Mid-range	110, 210, 301, 401
13	Mid-range	111, 210, 300, 400
14	Mid-range	111, 210, 300, 401
15	High-range	111, 210, 301, 400
16	High-range	111, 210, 301, 401

Table 2.1: Specification of all PC types considered in the TAC SCM game.

Component id	Supplier	Description
100	Pintel	Pintel CPU, 2.0 GHz
101	Pintel	Pintel CPU, 5.0 GHz
110	IMD	IMD CPU, 2.0 GHz
111	IMD	IMD CPU, 5.0 GHz
200	Basus, Macrostar	Pintel motherboard
210	Basus, Macrostar	IMD motherboard
300	MEC, Queenmax	Memory, 1 GB
301	MEC, Queenmax	Memory, 2 GB
400	Watergate, Mintor	Hard disk, 300 GB
401	Watergate, Mintor	Hard disk, 500 GB

Table 2.2: Specification of all components considered in the TAC SCM game.

2.2.4 Suppliers

All suppliers provide two components and every component type is provided by two suppliers. Suppliers are revenue maximizing entities preferring to serve traders with a good reputation on a make-to-order basis. A supplier's production capacity is determined on a daily basis using a random walk. In case the production capacity does not enable due dates to be met, late orders are given priority, whereas excess production capacity can be used to produce already outstanding future orders. However, orders are not shipped before their due dates. In case a supplier is not be able or willing to supply the entire quantity requested by the due date, this supplier issues two amended offers, each of which relaxes either quantity or due date; a partial offer (embodying the delivery of only part of the quantity on the due date) and an earliest complete offer (proposing the delivery of the entire quantity after the due date).

2.3 Making Sales Decisions

In the TAC SCM game, trading agents must make a number of decisions on a daily basis, as discussed in Section 2.2.3. With respect to sales, these decisions can have a tactical nature (e.g., determining product pricing strategies with respect to bidding on customer RFQs) as well as a strategic nature (e.g., product mix decisions – which translate into RFQ selection behavior – and production planning).

Due to the specific characteristics of the TAC SCM game environment detailed in Section 2.2, some research results in related work on product pricing presented in Section 2.1 may not be directly applicable in the context of the TAC SCM game. For instance, in [33], it is argued that sellers must be aware of the fact that customer demand depends on price history, quantified in perceived expensiveness. However, this extent of rationality is not modeled in the TAC SCM game, where customer demand is randomized (see Section 2.2.2). Also, in [8], the modeled price distribution is considered to be constant in the observed time frame, whereas price distributions vary with the market characteristics within the TAC SCM game environment [15, 16, 17]. The customer offer acceptance probability approximation (and hence the product pricing process) should therefore be related to market conditions, as also suggested in [35]. However, market conditions are hard to detect in the TAC SCM game due to limited visibility.

Hence, this section discusses how to make sales decisions in the TAC SCM game. An overview of mechanisms underlying the sales related decision process in the context of the TAC SCM game as proposed in recent literature is presented in Section 2.3.1. The strategy used by the MinneTAC agent is elaborated in Section 2.3.2.

2.3.1 TAC SCM Sales Strategies

With respect to the sales processes, several strategies are used by agents developed for the TAC SCM game. The so-called Dummy agent is a fairly non-complex agent that comes with the TAC SCM game and is used as default competitor. With respect to sales, this agent only bids on those RFQs, for which there is enough time to procure, manufacture, and deliver the requested product. Furthermore, the agent does not want the reservation prices to be too low compared to the production costs. For the selected RFQs, prices are offered a random amount under the reservation prices, depending on factory utilization level (the higher the utilization, the higher the offered prices).

More complex approaches have been proposed in literature. For instance, TacTex [29] predicts demand (using a Bayesian approach introduced by the DeepMaize team [18]) and offer acceptance (using linear regression) and adapts these offer acceptance estimations to its opponents' behavior. Another approach is to directly model the behavior of the competing agents and thus predict their offers [19]. Here, a genetic programming technique reveals a priori the attributes most indicative of the price offered by an arbitrary competitor. This information is subsequently used in order to construct an artificial neural network aiming to determine this competitor's offered price. The SouthhamptonSCM [12] agent uses fuzzy reasoning for daily price adaptation. For predicting whether a particular price for a particular product will be accepted by a customer, the CMieux agent [4] uses probability distributions trained off-line. These distributions are used in solving a continuous knapsack problem for selecting offers that maximize expected revenue. Here, a knapsack is to be filled with items (RFQs), the value of each of which considers the probability distributions for orders to be accepted for the specified price.

On the other hand, some TAC SCM trading agents use less complex techniques for their sales decisions. One of those agents is the Botticelli agent [3]. This trading agent uses a hill climbing heuristic for determining on what customer RFQs to bid. PhantAgent [37] does not use complex optimization techniques either. Simple heuristics are used for determining what to sell for what price.

In the context of using simple heuristics instead of more complex approaches when pricing products, it is argued in [37] that exact acceptance probabilities for given prices are not very relevant and an average bid acceptance probability is used, hereby assuming all six competing agents to have an equal market share on average. These assumptions are in contrast with the MinneTAC approach, in which exact acceptance probabilities are considered to be very valuable in the theoretical foundations of the economic regime model used in MinneTAC, as discussed in Section 2.3.2.

2.3.2 The MinneTAC Approach

Predicting sales prices is an important part of the decision process of trading agents [15, 16, 17]. This also holds for MinneTAC decision making processes. The estimated sales prices used in the MinneTAC agent originate from microeconomic conditions, which are characterized for each individual market segment – economic regimes are identified and predicted. The use of regimes is motivated by recent research showing that the ability of decision makers to correctly identify the current regime and predict the onset of a new regime is crucial in order to prevent over- or underreaction to market conditions [23].

A regime can be considered to be a set of conditions, characterizing the state of a system or process. Regimes provide an intuitive way of conditioning behavior in different scenarios. In literature, several approaches to regime identification and prediction have been proposed in different contexts. For instance, in the context of using real-time signals to determine the state of the plasma used in a nuclear fusion reactor, a Mamdani type of fuzzy logic system [22] and support vector machines have been considered [27]. The power of fuzzy techniques in the context of regimes has also been demonstrated in [11]. Furthermore, a Markov switching approach to modeling regime switches as proposed by Hamilton [10] is used in many approaches to predicting regime switches in electricity markets. Whereas the original Hamilton model considers two regimes, three regimes are considered in [13]. In other approaches, the Hamilton model is modified in order to support transition probabilities varying over time [2, 26].

In the MinneTAC agent, the regime of an arbitrary game day d for good g can be identified using regime probabilities; the regime having the highest probability, given the estimated normalized mean price of that day is assumed to be the current dominant regime. This price estimate is a smoothed normalized mid-range price, which is the average of the exponentially smoothed normalized minimum and maximum price, both of which are smoothed using double exponential smoothing (Brown linear exponential smoothing). To this end, prices are first of all normalized using

$$p_{dg} = \frac{p_{dg}^{\text{real}}}{c_{dg}^{\text{man}} + \sum_{j=1}^{\varpi_g} c_{dgj}^{\text{nom}}}, \tag{2.1}$$

where on game day d for good g, p_{dg} is the normalized price, p_{dg}^{real} is the real price, c_{dg}^{man} is the manufacturing cost, ϖ_g is the number of parts needed to make good g, and c_{dgj}^{nom} is the nominal cost of part j of good g on day d.

Equations (2.2) through (2.4) show how to smooth a day's normalized minimum price for a good p_{dg}^{min} using double exponential smoothing with smoothing factor $\beta = 0.5$, the result of which is referred to as $\widetilde{p}_{dg}^{\text{min}}$. Here, smoothing is done using two components, the first of which consists of a linear combination of a new observation and the previous first component. The second component is a linear combination of the first component and the previous second component. Then, a linear combination of the two components yields the smoothed price. The same procedure is applied to the normalized maximum price $p_{dg}^{\max}$ as well, which results in $\widetilde{p}_{dg}^{\max}$. Finally, the exponentially smoothed normalized price of game day d, $\widetilde{p}_{dg}$ is defined as shown in (2.5).

$$\widetilde{p}_{dg}^{\min\prime} = \beta p_{dg}^{\min} + (1-\beta)\widetilde{p}_{(d-1)g}^{\min\prime}, \tag{2.2}$$

$$\widetilde{p}_{dg}^{\min\prime\prime} = \beta \widetilde{p}_{dg}^{\min\prime} + (1-\beta)\widetilde{p}_{(d-1)g}^{\min\prime\prime}, \tag{2.3}$$

$$\widetilde{p}_{dg}^{\min} = 2\widetilde{p}_{dg}^{\min\prime} - \widetilde{p}_{dg}^{\min\prime\prime}, \tag{2.4}$$

$$\widetilde{p}_{dg} = \frac{\widetilde{p}_{dg}^{\min} + \widetilde{p}_{dg}^{\max}}{2}. \tag{2.5}$$

In the TAC SCM game, price information is only available up until the preceding game day. Hence, on game day d, the most up-to-date mean price approximation for good g is $\widetilde{p}_{(d-1)g}$. This can be used as input for an off-line trained model which predicts the mean price of day d using exponential smoothing prediction of $\widetilde{p}_{dg}$ and subsequently returns the regime probabilities for day d. As an economic regime is regarded as a distribution of prices over sales volume, acceptance probability densities associated with given product prices are implicitly incorporated in this model. These probabilities are in fact inverse cumulative price density functions.

In the training phase, a product-level price density function has been modeled on historical normalized order price data using a Gaussian Mixture Model with a sufficient number of Gaussians, reflecting a balance between prediction accuracy and computational overhead. Clustering these price distributions over time periods (using the k-means algorithm [21]) yielded distinguishable statistical patterns, referred to as economic regimes. When on game day d the exponentially smoothed prediction of $\widetilde{p}_{dg}$ for good g is supplied to the model, the individual Gaussians in the model are all activated to a certain extent, thus generating an expected price distribution. Subsequently calculating all clusters' normalized price density enables the regime probabilities, given prices, to be determined.

The behavior of the MinneTAC agent depends on expected future regimes as well. With respect to regime prediction, for short-term prediction for tactical decision making, the agent uses a Markov prediction process based on the last normalized smoothed mid-range price, hereby utilizing Markov transition matrices created off-line by a counting process over past games. For long-term prediction for strategic decision making, the MinneTAC agent uses a Markov correction-prediction process, where predictions are based on all normalized smoothed mid-range prices up until the previous day instead of the last normalized smoothed mid-range price only.

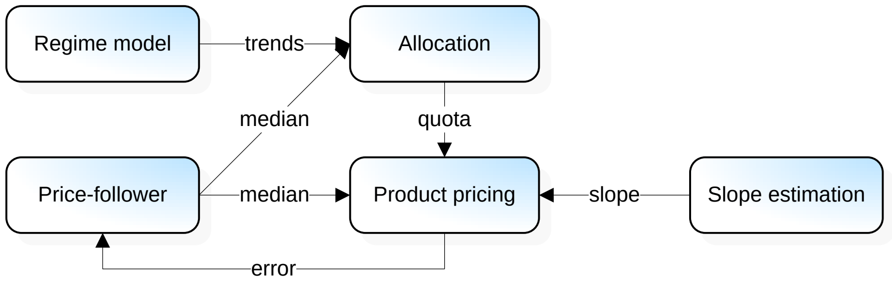


Figure 2.1: Simplified schematic overview of the current MinneTAC sales process configuration for an arbitrary product on an arbitrary game day.

In the Markov models, three basic states and two regimes characterized by extreme conditions are considered. These economic regimes can be extreme scarcity, scarcity, a balanced situation, oversupply, or extreme oversupply. Regimes can also be predicted based on exponentially smoothed price predictions taking into account exponentially smoothed price trends.

When current or future regime probabilities have been determined, products are priced based on the likelihood of customer acceptance of these prices, such that a sales quota is fulfilled. These processes of regime identification, regime prediction, and product pricing, as well as other processes in the MinneTAC trading agent, are supported by a configurable chain of evaluators [7]. The software selects the optimal configuration of evaluators.

In the 2008 sales process configuration, depicted in Figure 2.1, price trends are estimated by the regime model. The median price of a product is estimated using a price-following approach implementing a Brown linear exponential smoother. These trends, combined with the estimated median, are used in the allocation process, where sales quotas are generated based on – among other things – these price predictions. The curve representing the acceptance probability is approximated using the estimated median price and the curve's slope in that median, estimated using exponentially smoothed prices. This acceptance probability is used for determining the price to be offered in order for the sales agent to sell its desired quota.

In order to compensate for the uncertainty in generated predictions, interval randomization is applied to offer prices, which adds a slight variability to these prices. The estimated median is corrected using feedback derived from the desired acceptance probability and the associated true acceptance probability observed the next day. The error is computed as the difference between the optimized offer price and the actual price. The latter price is derived by solving the acceptance probability estimate to the observed probability. A major drawback here is the assumption that customer feedback is in response to the optimized offer price, whereas this feedback in fact is in response to a price randomized in an interval around this price.

2.4 Summary

According to literature discussed in this chapter, the probability of acceptance of prices offered by sellers can be modeled and trained off-line. Another option is to model acceptance probabilities on-line using for example linear regression or by explicitly modeling and learning the decision function and/or behavior of buyers. Another approach is to base acceptance probability estimations on the (observed or estimated) inverse of the cumulative distribution of prices customers are willing to pay.

In order for the probability of acceptance thus estimated to be useful in product pricing, some other factors need to be taken into consideration. First of all, product pricing strategies could be related to market conditions, outside options, or maybe even other additional information. Furthermore, modeling (expected or observed) supply-side behavior in the product pricing process could contribute to profit maximization. When modeling behavior of bidding parties, bids should not be considered equal to, yet related to each bidder's private value for the considered good.

Within the TAC SCM game, there are no daily direct indications of market conditions, other than observed daily minimum and maximum order prices for a product. Therefore, most product pricing approaches proposed for the TAC SCM game just model the behavior of customers and competitors in order to determine the optimal price for a product. MinneTAC on the other hand does consider market conditions in the form of economic regimes. However, this information is not directly linked to the product pricing process yet; a straightforward price-following approach, only accounting for the available order price information, is used for estimating acceptance probabilities instead. This implies that, based on the findings in literature, MinneTAC's product pricing process could be improved when regime information (as well as other additional information) is linked to the estimation of acceptance probabilities.

Chapter 3

Realization of Product Prices in the TAC SCM Game

As MinneTAC's regime model internally considers price distribution estimations, these estimations could be connected to the product pricing process in order for regime information to be utilized in this process, as suggested in Chapter 2. Therefore, the connection of price distributions to the product pricing process in the context of the TAC SCM game is modeled in this chapter. To this end, Section 3.1 discusses how the general product pricing process works in the TAC SCM game. Section 3.2 expands on the characteristics of the associated price distributions. In Section 3.3, price distributions are connected to the product pricing process by modeling the probabilities that customers accept prices offered for their RFQs. These acceptance probability estimations can be used in the product pricing process (see Chapter 2). Finally, the findings are summarized in Section 3.4.

3.1 From Requests to Orders

In the TAC SCM game, on each of 220 game days, customers issue RFQs, in which they request specific quantities for any of the 16 available PC types. Customers hereby express the maximum price they are willing to pay for this order through a reservation price. Several customers may request some quantity of a product, so each game day, multiple negotiation processes can take place for one PC type. In such a negotiation process, traders may respond to a customer's RFQ by submitting their offers. The customer then places an order with the trader offering at the lowest price, if this minimum offer price does not exceed the customer's reservation price.

Hence, in a typical TAC SCM game, the number of offers placed by traders (and thus the number of offer prices) for a specific product may amount to six times the number of RFQs issued by customers for that product, in case all traders respond to all RFQs issued for the considered product.

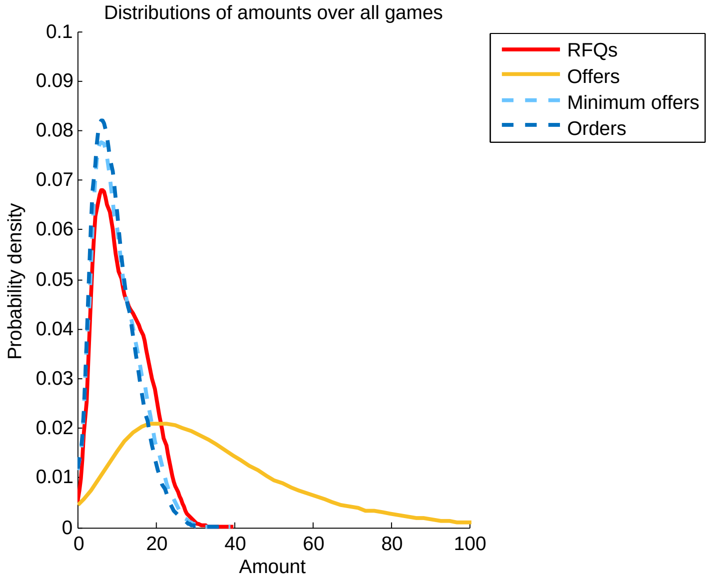


Figure 3.1: Distributions of the number of RFQs, offers, minimum offers, and orders per product per day, aggregated over all games in the training set.

The minimum offer prices per product form a subset of all offer prices. The size of this subset equals at most the number of RFQs per product, as each negotiation process – induced by a customer's RFQ – is associated with its own minimum price. In turn, the prices of the orders resulting from these minimum offer prices form a subset of the latter prices, as best offers do not necessarily result in orders due to the reservation price constraint.

Figure 3.1 visualizes the distributions of the number of RFQs, offers, minimum offers, and orders per product per day, aggregated over all games in the training set. A more detailed statistical description of these distributions can be found in Table A.1 in Appendix A. This table shows the mean and standard deviation of the distribution of the number of RFQs, offers, minimum offers, and orders per product per day on game level. On average, the number of RFQs per product per day equals about 11. The number of offers generated in reaction to these RFQs apparently roughly equals 34 per product per day, whereas the number of minimum offers only equals about 10. Surprisingly, the average number of orders is even lower: 9.

The latter observation implies that in practice, for some RFQs holds that none of the traders bidding on such an RFQ offers a valid price (a price not exceeding the customer's reservation price associated with that RFQ). Tables A.2 and A.3 in Appendix A illustrate the magnitude of this problem. The irrational pricing behavior occurred in the TAC SCM 2007 Semi-Finals and Finals games and was mainly caused by one trading agent. Per game in the training set, up to about 10% (3% on average) of all offer prices is invalid and would never be accepted by a customer. This irrational pricing behavior can cause over 20% (5% on average) of all active negotiation processes not to result in an order. Invalid prices exhibit less distortion with respect to their associated reservation prices than valid prices do. This distortion can be quantified by the standard deviation of observations with respect to their benchmark values; the root mean squared deviation (RMSD):

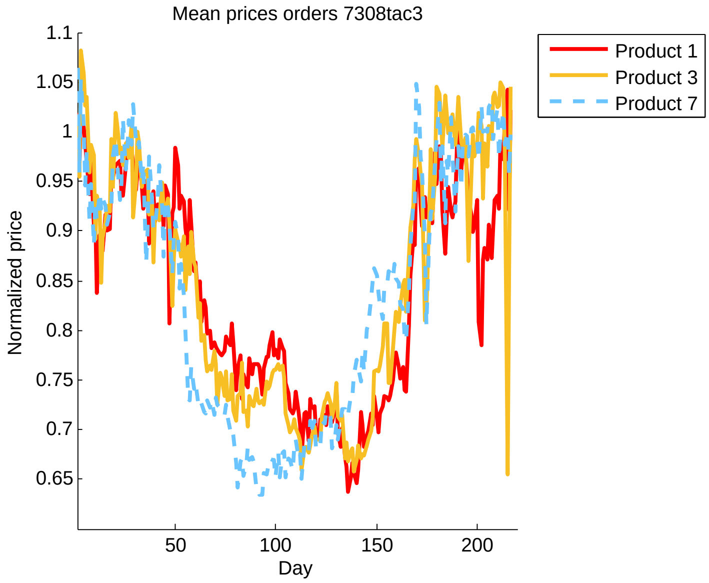
$$\text{RMSD} = \sqrt{\frac{\sum_{\omega=1}^{\Omega} \left(\hat{x}_{\omega} - x_{\omega}\right)^{2}}{\Omega}}. \tag{3.1}$$

In (3.1), $\hat{x}_{\omega}$ is an observation (in a set of Ω observations), the associated benchmark value of which is x_{ω}. The RMSD values of normalized prices from their associated normalized reservation prices in Tables A.2 and A.3 indicate that invalid pricing behavior is overall not relatively more extreme than valid pricing behavior. This implies that the pricing behavior of agents placing invalid offers is more or less normal, except for that these agents sometimes place offers which would never be accepted by a customer.

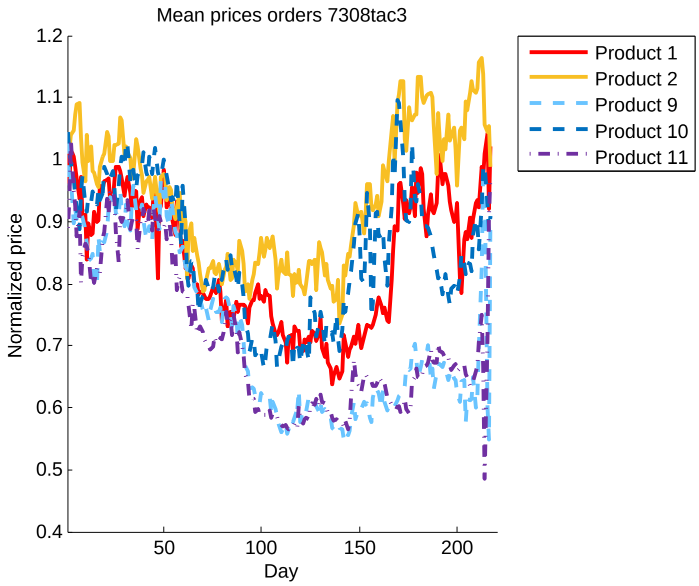
3.2 Price Distributions in the TAC SCM Game

The observations exhibited in Section 3.1 imply that for an arbitrary sample of offered prices, the distribution of prices associated with placed orders is related to the distribution of all offer prices through the distribution of all *valid* offer prices. The distribution of the minimum valid offer prices is equal to the distribution of the order prices. In order to facilitate a better understanding of the characteristics of price distributions in the TAC SCM game, this section discusses the most crucial features of the way prices are typically distributed in the TAC SCM game.

Figure 3.2(a) depicts the mean price associated with orders for a low-range, mid-range, and high-range market segment product in a typical TAC SCM game. Clearly, product price distributions can be very distinctive between market segments. However, within a market segment, price distributions may also differ per product, as shown in Figure 3.2(b). Hence, price distributions should be considered on product level. Figure 3.2 also suggests a high volatility in distribution characteristics; distributions apparently are not very stable over time. This observation is supported by the results of an analysis of variance (ANOVA) in Table 3.1, which assesses for a typical low-range, mid-range, and high-range market segment product the probabilities of multiple subsequent order price samples to be drawn from a population with the same mean. The initial probability of similar order price distributions within a window of 2 periods, which on average approximately equals a mere 20%, turns out to decrease fast when window sizes increase.



(a)



(b)

Figure 3.2: Normalized mean order prices for product 1, 3, and 7 (low-range, mid-range, and high-range market segment products, respectively) over time in a typical TAC SCM game (7308tac3) are depicted in (a), whereas (b) visualizes normalized mean order prices for product 1, 2, 9, 10, and 11 (low-range market segment products) over time in this game.

	Product 1		Product 3		Product 7	
Window	Mean	Stdev	Mean	Stdev	Mean	Stdev
2	0.2009	0.2896	0.2014	0.2835	0.1922	0.2829
3	0.1162	0.2319	0.1209	0.2310	0.1070	0.2200
4	0.0725	0.1828	0.0818	0.1947	0.0689	0.1781
5	0.0467	0.1454	0.0581	0.1663	0.0471	0.1473
6	0.0320	0.1206	0.0424	0.1427	0.0334	0.1243
7	0.0232	0.1024	0.0302	0.1170	0.0254	0.1087
8	0.0173	0.0911	0.0224	0.0983	0.0186	0.0909
9	0.0120	0.0756	0.0167	0.0831	0.0136	0.0750
10	0.0089	0.0656	0.0126	0.0712	0.0103	0.0652

Table 3.1: Statistics on the probabilities of a distribution of order prices to be constant within a window (sizes ranging from 2 to 10) for product 1, 3, and 7 (low-range, mid-range, and high-range market segment products, respectively) for all days in all games in the training set.

Figure 3.3 depicts a typical distribution of several types of prices related to a product on a TAC SCM game day: offer prices, minimum offer prices, valid offer prices, and order prices. Offer prices and valid offer prices appear to be nicely distributed, but due to data sparsity, distributions of minimum offer prices and especially order prices are hard to estimate. Furthermore, the dispersion of prices in the latter two distributions tends to be rather small (often close to or equal to zero), as depicted in Figure 3.4.

3.3 Utilizing Price Distributions in Estimations of Customer Offer Acceptance Probabilities

Taking into consideration the observations in Section 3.2, a mathematical framework for determining the customer offer acceptance probabilities, based on estimations of order price distributions, can be constructed. An order price distribution should be equal to the distribution of the minimum valid offer prices and hence is directly linked to the distribution of all valid offer prices. However, the combination of data sparsity and low dispersion of prices implies that an order price distribution is likely to have a weaker linkage to the distribution of all valid offer prices, when fit directly on available order price data. Hence, distributions of order prices cannot be accurately determined by directly fitting on the available data on order prices.

As data sparsity prevails due to the fact that window sizes cannot be enlarged in order to increase the sample size (price distributions do not tend to be stable over time), order price distributions are derived as the distributions of minimum valid offer prices. The robustness thus introduced may result in this approach to be preferable over other approaches involving direct estimation of realized prices or estimation of individual competitors' offers, such as proposed in [19]. Even more, incorporating offer price distributions (and the distributions of their minima) rather than individual

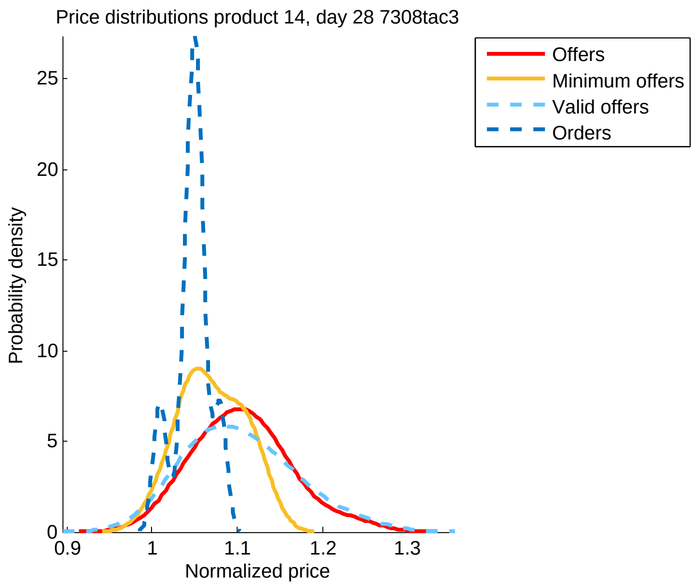


Figure 3.3: Distribution of normalized offer prices, minimum offer prices, valid offer prices, and order prices for product 14 on day 28 of a typical TAC SCM game (7308tac3).

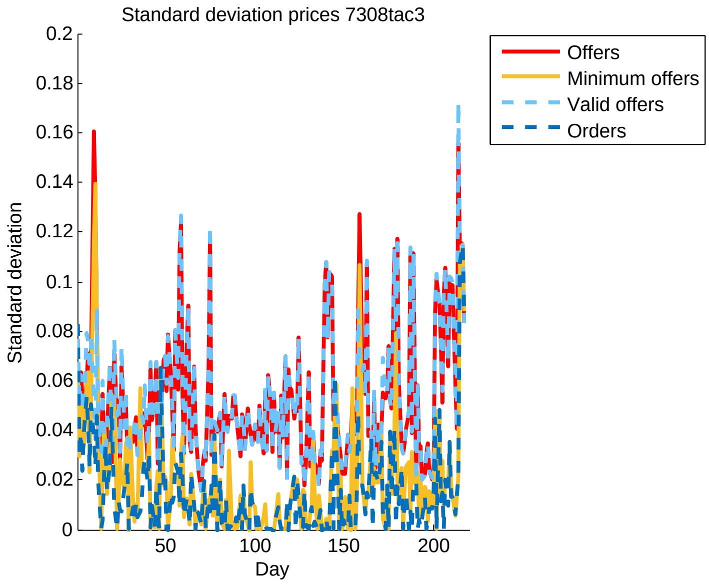


Figure 3.4: Standard deviation of normalized offer prices, minimum offer prices, valid offer prices, and order prices for product 14 over time in a typical TAC SCM game (7308tac3).

offers into the framework can compensate for a drawback associated with the latter approach, in which offer prices of competitors are predicted even though these competitors may not actually bid. This structurally causes the agent's offers to be relatively low (i.e. lower than necessary in order to win the bidding process). When reasoning in terms of offer price distributions rather than individual offers, the phenomenon of taking into account non-existing offers is better accounted for, as the offer price distributions are formed by all offers actually done rather than offers all competitors would make, should they actually bid.

This section continues with the formalization of the proposed approach in Section 3.3.1. Subsequently, a distribution appropriate for use in the context of the TAC SCM game is incorporated into this framework in Section 3.3.2.

3.3.1 General Framework

Let n randomly sampled valid normalized prices p offered for an arbitrary RFQ be identically and independently distributed in accordance with a distribution $f(p;\theta)$ and a cumulative distribution $F(p;\theta)$, with $0 < p < 1.25$ and θ a vector of unknown parameters. For such a distribution, the minimum valid offer price (and thus the order price) $\underline{p}$ is distributed as follows [14]:

$$f_{\underline{p}}\left(\underline{p};\theta\right) = n\left(1 - F\left(\underline{p};\theta\right)\right)^{n-1} f\left(\underline{p};\theta\right), \quad 0 < \underline{p} < 1.25, \quad n > 0, \tag{3.2}$$

$$F_{\underline{p}}\left(\underline{p};\theta\right) = 1 - \left(1 - F\left(\underline{p};\theta\right)\right)^{n}, \quad 0 < \underline{p} < 1.25, \quad n > 0. \tag{3.3}$$

Let n_{dgr} randomly sampled valid normalized prices offered on game day d for product g for RFQ r $(p_{dgr1}, ..., p_{dgrn_{dgr}})$ be identically and independently distributed and have an associated order price $\underline{p}_{dgr}$. Furthermore, let t be the size of the observed time frame (i.e. the number of repeated daily observations of valid offered prices), s the size of the set of considered products, and b_{dg} the number of bidding processes started for product g on game day d. Then, the joint distribution of order prices in an s-sized set in the t-period sample can be formulated as

$$f_{\underline{p}}\left(\underline{p}_{111}, ..., \underline{p}_{tsb_{ts}};\theta\right) = \prod_{d=1}^{t}\prod_{g=1}^{s}\prod_{r=1}^{b_{dg}} n_{dgr}\left(1 - F\left(\underline{p}_{dgr};\theta\right)\right)^{n_{dgr}-1} f\left(\underline{p}_{dgr};\theta\right),$$
$$0 < \underline{p}_{dgr} < 1.25, \quad n_{dgr} > 0. \tag{3.4}$$

The unknown parameters θ of the distribution of offer prices incorporated in this framework can be estimated by minimizing $L(\theta;\vec{p})$, the negated log-likelihood function of these parameters for a sample of prices $\vec{p}$ – in this case the observed order prices $(\underline{p}_{111}, ..., \underline{p}_{tsb_{ts}})$ – as defined in (3.5).

$$
\begin{aligned}
&L\left(\theta;\underline{p}_{111},...,\underline{p}_{tsb_{ts}}\right)\\
&=\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}-\ln\left(n_{dgr}\left(1-F\left(\underline{p}_{dgr};\theta\right)\right)^{n_{dgr}-1}f\left(\underline{p}_{dgr};\theta\right)\right),\\
&0<\underline{p}_{dgr}<1.25,\quad n_{dgr}>0. \qquad (3.5)
\end{aligned}
$$

However, due to the sparseness of data on order prices, the unknown parameters θ could best be estimated by fitting the distribution underlying all prices offered for s products in t periods on the associated sample of observed valid offer prices $(p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}})$, which is typically larger than the associated sample of order prices (see Section 3.1). Assuming all prices in the sample to be identically and independently distributed in accordance with the distribution of offer prices $f(p;\theta)$, the joint distribution of all valid offer prices in an s-sized set of product types in a t-period sample can be formulated as

$$
f\left(p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}};\theta\right)=\prod_{d=1}^{t}\prod_{g=1}^{s}\prod_{r=1}^{b_{dg}}\prod_{i=1}^{n_{dgr}}f\left(p_{dgri};\theta\right),\quad 0<p_{dgri}<1.25. \qquad (3.6)
$$

Given (3.6), the parameters θ of the distribution from which all offer prices are assumed to be originating can be estimated by minimizing the negated log-likelihood function of these parameters for a sample of observed offer prices $(p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}})$. This can be done by applying

$$
L\left(\theta;p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}}\right)=\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}-\ln\left(f\left(p_{dgri};\theta\right)\right),\quad 0<p_{dgri}<1.25. \qquad (3.7)
$$

For an RFQ r for product g at game day d, the probability that a customer will place an order with an agent, given its offer price p, $P(o|p)$, can be determined using the cumulative density function of the order price as defined in (3.3). As $F_{\underline{p}}(p;\theta)$ yields the fraction of order prices $\underline{p}$ realized at or below p, the output of this function may be regarded as the probability that an order o is placed with another agent offering a similar or better deal. Hence, the probability that a customer accepts an offered price p can be approximated using

$$
\begin{aligned}
P(o|p)&=1-F_{\underline{p}}(p;\theta),\quad 0<p<1.25\\
&=(1-F(p;\theta))^{n_{dgr}},\quad 0<p<1.25,\quad n_{dgr}>0. \qquad (3.8)
\end{aligned}
$$

3.3.2 Incorporating a Log-Logistic Distribution

Equations (3.2) through (3.8) assume the prices to be distributed in accordance with a distribution the type and parameters of which have not been defined yet. For this purpose, a log-logistic distribution is proposed, as this distribution is defined on the domain $0 < p < \infty$ and covers a variety of shapes depending on the parameters φ and χ. Another attractive feature is that an analytical closed form expression exists for the cumulative density function. The log-logistic distribution $f(p;\varphi,\chi)$ and its cumulative form $F(p;\varphi,\chi)$ [25] can be described as follows:

$$f\left(p;\varphi,\chi\right) = \frac{\varphi\chi p^{\chi-1}}{\left(1+\varphi p^{\chi}\right)^2}, \quad 0<p<\infty, \quad \varphi,\chi>0, \tag{3.9}$$

$$F\left(p;\varphi,\chi\right) = 1 - \frac{1}{1+\varphi p^{\chi}}, \quad 0<p<\infty, \quad \varphi,\chi>0. \tag{3.10}$$

The domain of the log-logistic distribution may in theory equal the range of the prices possibly offered during a TAC SCM game, but in practice, all valid normalized offer prices can only be in the interval $[0, 1.25]$, because all normalized reservation prices are in the interval $[0.75, 1.25]$. Hence, the log-logistic distribution should in this case be double-bounded, such that the upperbound u yields a distribution defined on the domain $0 < p < 1.25$. This can be accomplished by expressing the log-logistic distribution in relation to the cumulative density at $p = u$, which yields

$$\begin{aligned} F\left(p;\varphi,\chi\right) &= \frac{1-\frac{1}{1+\varphi p^{\chi}}}{1-\frac{1}{1+\varphi u^{\chi}}}, \quad 0<p<u, \quad \varphi,\chi>0 \\ &= \frac{\varphi+u^{-\chi}}{\varphi+p^{-\chi}}, \quad 0<p<u, \quad \varphi,\chi>0, \end{aligned} \tag{3.11}$$

with $u = 1.25$ in order for the cumulative distribution to be ranging from 0 to 1 for valid offer prices in the interval $[0, 1.25]$. Inherently, the double-bounded log-logistic probability density function can be defined as

$$f\left(p;\varphi,\chi\right) = \frac{\left(\varphi+u^{-\chi}\right)\chi p^{-\chi-1}}{\left(\varphi+p^{-\chi}\right)^2}, \quad 0<p<u, \quad \varphi,\chi>0. \tag{3.12}$$

Using (3.11), the median valid offer price (where the cumulative density equals 0.5) can be derived as $\left(\varphi+2u^{-\chi}\right)^{-\frac{1}{\chi}}$. Reparameterizing the distribution, such that α represents the median and γ equals χ, yields

$$f\left(p;\alpha,\gamma\right) = \frac{\left(\alpha^{-\gamma}-u^{-\gamma}\right)\gamma p^{-\gamma-1}}{\left(\alpha^{-\gamma}-2u^{-\gamma}+p^{-\gamma}\right)^2}, \quad 0<p<u, \quad \alpha,\gamma>0, \tag{3.13}$$

$$F\left(p;\alpha,\gamma\right) = \frac{\alpha^{-\gamma}-u^{-\gamma}}{\alpha^{-\gamma}-2u^{-\gamma}+p^{-\gamma}}, \quad 0<p<u, \quad \alpha,\gamma>0. \tag{3.14}$$

The parameters of the distribution underlying all sampled valid normalized offer prices of s product types in a period of t game days can be estimated by minimizing the log-likelihood function specified in (3.7), hereby assuming an underlying log-logistic distribution as described in (3.13). Then, the unknown parameters α (representing the median) and γ (quantifying the distribution tightness) can be estimated on a routinely basis. The optimal values for α and γ are to be found, such that for the log-logistic log-likelihood function defined in (3.15), the first order partial derivatives to α and γ (defined in (3.16) and (3.17), respectively) equal 0:

$$
\begin{aligned}
&L\left(\alpha,\gamma;p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}}\right)\\
&=\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}-\ln\left(\frac{\left(\alpha^{-\gamma}-u^{-\gamma}\right)\gamma p_{dgri}^{-\gamma-1}}{\left(\alpha^{-\gamma}-2u^{-\gamma}+p_{dgri}^{-\gamma}\right)^{2}}\right),\\
&0<p_{dgri}<u,\quad \alpha,\gamma>0,
\end{aligned}
\tag{3.15}
$$

$$
\begin{aligned}
&\frac{\delta L\left(\alpha,\gamma;p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}}\right)}{\delta\alpha}\\
&=\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}\frac{-\gamma u^{2\gamma}\left(\alpha^{\gamma}-p_{dgri}^{\gamma}\right)}{\alpha\left(\alpha^{\gamma}-u^{\gamma}\right)\left(-2\left(\alpha p_{dgri}\right)^{\gamma}+\left(\alpha u\right)^{\gamma}+\left(u p_{dgri}\right)^{\gamma}\right)},\\
&0<p_{dgri}<u,\quad \alpha,\gamma>0,
\end{aligned}
\tag{3.16}
$$

$$
\begin{aligned}
&\frac{\delta L\left(\alpha,\gamma;p_{1111},...,p_{tsb_{ts}n_{tsb_{ts}}}\right)}{\delta\gamma}\\
&=\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}\frac{u^{2\gamma}\left(\alpha^{\gamma}+p^{\gamma}+\gamma\left(p_{dgri}^{\gamma}-\alpha^{\gamma}\right)\left(\ln\left(\alpha\right)-\ln\left(p_{dgri}\right)\right)\right)}{\gamma\left(\alpha^{\gamma}-u^{\gamma}\right)\left(u^{\gamma}\left(\alpha^{\gamma}+p_{dgri}^{\gamma}\right)-2\alpha^{\gamma}p_{dgri}^{\gamma}\right)}+\\
&\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}\frac{-\alpha^{\gamma}\gamma u^{\gamma}\left(\alpha^{\gamma}-3p_{dgri}^{\gamma}\right)\left(\ln\left(p_{dgri}\right)-\ln\left(u\right)\right)}{\gamma\left(\alpha^{\gamma}-u^{\gamma}\right)\left(u^{\gamma}\left(\alpha^{\gamma}+p_{dgri}^{\gamma}\right)-2\alpha^{\gamma}p_{dgri}^{\gamma}\right)}+\\
&\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}\frac{\alpha^{\gamma}u^{\gamma}\left(-\alpha^{\gamma}-3p_{dgri}^{\gamma}\right)}{\gamma\left(\alpha^{\gamma}-u^{\gamma}\right)\left(u^{\gamma}\left(\alpha^{\gamma}+p_{dgri}^{\gamma}\right)-2\alpha^{\gamma}p_{dgri}^{\gamma}\right)}+\\
&\sum_{d=1}^{t}\sum_{g=1}^{s}\sum_{r=1}^{b_{dg}}\sum_{i=1}^{n_{dgr}}\frac{2\alpha^{2\gamma}p_{dgri}^{\gamma}\left(1-\gamma\left(\ln\left(p_{dgri}\right)-\ln\left(u\right)\right)\right)}{\gamma\left(\alpha^{\gamma}-u^{\gamma}\right)\left(u^{\gamma}\left(\alpha^{\gamma}+p_{dgri}^{\gamma}\right)-2\alpha^{\gamma}p_{dgri}^{\gamma}\right)},\\
&0<p_{dgri}<u,\quad \alpha,\gamma>0.
\end{aligned}
\tag{3.17}
$$

Because of the non-linearity of the log-likelihood function and its first order partial derivatives, constrained optimization of multiple parameters is not a trivial task. To this end, algorithms like the constrained non-linear optimization algorithm for problems with linear constraints implemented in the `fmincon` function in the `Optimization` toolbox for `MATLAB` [39] could be used. This algorithm is based on a variant of the Newton-Raphson method described in [5] and basically is a trust-region method. This means that the function to be minimized is approximated with a simpler function in a neighborhood (the region of trust) around a point considered to be the minimum. Each iteration, the simplified function is minimized and the alleged minimum of the real function is updated with the minimization result if this result truly yields a lower value for the real function. Otherwise, the region of trust is reduced. This process is iterated until a minimum size of the region of trust is reached.

When the α and γ parameters have been fit, the probability $P(o|p_{dgr})$ that a customer accepts a price p_{dgr} offered by an agent for RFQ r for product g on game day d, given the upper bound price u, can be approximated. Following (3.8), the calculation of acceptance probabilities in case of an underlying log-logistic offer price distribution is detailed in (3.18), where α and γ are the distribution parameters, estimated using (3.15) through (3.17), and n_{dgr} represents the number of valid normalized prices offered on game day d for product g for RFQ r:

$$\begin{aligned} P(o|p_{dgr}) &= 1 - \left(1 - \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + p_{dgr}^{-\gamma}}\right)^{n_{dgr}}\right), \\ & \quad 0 < p_{dgr} < u, \quad \alpha, \gamma, n_{dgr} > 0 \\ &= \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + p_{dgr}^{-\gamma}}\right)^{n_{dgr}}, \\ & \quad 0 < p_{dgr} < u, \quad \alpha, \gamma, n_{dgr} > 0. \end{aligned} \tag{3.18}$$

The probability for an offer for a specific RFQ to be accepted, computed using (3.18), can be theoretically verified and validated using (3.19) and (3.20). Being defined for $0 < p_{dgr} < u$ and $\alpha, \gamma, n_{dgr} > 0$, the range of this function is in the interval $[0, 1]$, which is exactly the desired property. The function is also valid, as the behavior described by the function matches the behavior expected in the game situation. Relatively low prices are more likely to be accepted than relatively high prices.

$$\lim_{p_{dgr} \to 0} \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + p_{dgr}^{-\gamma}}\right)^{n_{dgr}} = 1, \quad \alpha, \gamma, n_{dgr} > 0, \tag{3.19}$$

$$\lim_{p_{dgr} \to u} \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + p_{dgr}^{-\gamma}}\right)^{n_{dgr}} = 0, \quad \alpha, \gamma, n_{dgr} > 0. \tag{3.20}$$

3.4 Summary

In the context of the TAC SCM game, analysis of historical game data in this chapter shows that product price distributions can be very distinctive between market segments, as well as on product level. Price distributions in the TAC SCM game also do not tend to be very stable over time.

Given perfect information, distributions of order prices needed for the product pricing process cannot be accurately determined by directly fitting on the available data on order prices. This is caused by a combination of data sparsity and low dispersion of prices. Enlarging the window size in order to increase the sample size is not an option in this case due to the volatility of price distribution characteristics over time.

Taking these observations into account, a method for determining the probability that a customer accepts an offered price is proposed in this chapter. This approach is based on daily, product-based estimations of distributions of valid offer prices. The assumption here is that for an arbitrary sample of offered prices, the distribution of associated order prices is related to the distribution of all valid offer prices; the distribution of the minimum of the valid offer price distribution equals the order price distribution.

Hence, the proposed approach involves firstly estimating the distribution of prices offered for an arbitrary day for a specific product. These prices are assumed to be distributed in accordance with a double-bounded log-logistic distribution with parameters α (representing the median) and γ (quantifying the distribution tightness). The distribution of order prices can subsequently be derived. Then, the probability that a customer accepts an offered price can be approximated by inverting the cumulative density function of the order price distribution estimation.

The approach considered in this chapter assumes all information to be available; distributions are estimated by fitting onto all valid offer prices for a specific product on a game day. However, in the TAC SCM game, information on prices offered on an arbitrary day is not available on-line. Therefore, Chapter 4 discusses how to translate the proposed approach into a feasible on-line product pricing approach.

Chapter 4

Towards an On-Line Product Pricing Approach

During the TAC SCM game, the MinneTAC agent determines a sales quota for each product type on a daily basis. Each product type is subsequently priced, such that its sales quota is expected to be fulfilled, hereby taking into account the likelihood of customer acceptance of this product price. Multiple RFQs can be filed for a particular product type on an arbitrary game day. All these RFQs are to be taken into account regarding the sales quota for this product type. The number of RFQs filed for aproduct g on day d, m_{dg}, is on-line observable. However, the number of offers for an RFQ r for product g on day d, n_{dgr}, is not observable and should hence be estimated. For now, let this number be approximated by $\overline{n}_{dg}$, the mean number of offers per RFQ for product g on day d.

Using the framework presented in Chapter 3, the probability that a customer accepts a price offered for his RFQ can be estimated. This RFQ-oriented framework can be extended to the product level, in order for it to be applicable in the MinneTAC agent's decision logic. As the customer offer acceptance probability of an arbitrary RFQ is determined using the cumulative density function of the order price of this RFQ (see (3.8) and (3.18)), the probability that an offered price gets accepted in case of multiple similar RFQs can be determined using these RFQs' joint cumulative distribution, hereby assuming the associated order prices to be identically and independently distributed. In this context, the customer offer acceptance probability $P\left(o|p_{dg}\right)$ for product g on game day d in case of m_{dg} RFQs with $\overline{n}_{dg}$ offers per RFQ can be approximated as

$$P\left(o|p_{dg}\right) = 1 - \left(1 - \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + p_{dg}^{-\gamma}}\right)^{\overline{n}_{dg}}\right)^{m_{dg}},$$
$$0 < p_{dg} < u, \quad \alpha, \gamma, m_{dg}, \overline{n}_{dg} > 0. \tag{4.1}$$

Now, let q^*_{dg} be the sales quota for an arbitrary product type g on an arbitrary day d, with m_{dg} associated RFQs, for each of which $\overline{n}_{dg}$ prices are offered. This implies that $P(o|p_{dg})$ for that product type on that game day is required to be q^*_{dg}. Using (4.1), the relation between q^*_{dg} and the offer price p^*_{dg} yielding this quota can be expressed as

$$q^*_{dg} = 1 - \left(1 - \left(1 - \frac{\alpha^{-\gamma} - u^{-\gamma}}{\alpha^{-\gamma} - 2u^{-\gamma} + {p^*_{dg}}^{-\gamma}}\right)^{\overline{n}_{dg}}\right)^{m_{dg}},$$
$$0 < p^*_{dg} < u, \quad \alpha, \gamma, m_{dg}, \overline{n}_{dg} > 0. \tag{4.2}$$

Equation (4.2) enables the normalized price to be offered on RFQs for product g on game day d to be determined. The optimal offer price p^*_{dg} expected to yield the desired quota q^*_{dg} can be approximated as follows:

$$p^*_{dg} = \left(\frac{u^{-\gamma}\left(\alpha^{-\gamma}\left(u^{\gamma} - 2\alpha^{\gamma}\right) \sqrt[\overline{n}_{dg}]{1 - \sqrt[m_{dg}]{1 - q^*_{dg}}} + 1\right)}{1 - \sqrt[\overline{n}_{dg}]{1 - \sqrt[m_{dg}]{1 - q^*_{dg}}}}\right)^{-\frac{1}{\gamma}},$$
$$0 < q^*_{dg} < 1, \quad \alpha, \gamma, m_{dg}, \overline{n}_{dg}, u > 0. \tag{4.3}$$

The number of RFQs m_{dg} used in the product pricing process described above is observable on-line. The mean number of offers per RFQ $\overline{n}_{dg}$ can be estimated based on historical data. Given perfect information, the parameters of the distribution of valid offer prices underlying the customer offer acceptance probability assumed in (4.3), α (representing the median) and γ (quantifying the distribution tightness), can be estimated using (3.15) through (3.17), while accounting for an upper bound price u of 1.25. For the TAC SCM game, let the number of repeated (daily) observations of valid offered prices t considered in this process equal 1, as the observations in Section 3.2 indicate that price distributions are not likely to be stable for more than 1 game day. Furthermore, let the number of considered products s equal 1 as well, as price distributions tend to differ on product level (see Section 3.2).

Unfortunately, the framework as formulated in Section 3.3 requires data which is not publicly available during a TAC SCM game. Equations (3.15) through (3.17) require offer prices to be available in order to fit distributions, whereas these prices are not known at any given time during a game. This problem could be solved by replacing (3.15) through (3.17) with some model which estimates the α and γ parameter, based on information available on-line to the agent. The mean number of offers per RFQ $\overline{n}_{dg}$ should also be approximated using on-line available information. In this context, an artificial neural network could be used.

Section 4.1 discusses artificial neural networks, whereas their applicability to the parameter estimation problem regarding the customer offer acceptance probability is discussed in Section 4.2. The on-line available information potentially indicative of the customer offer acceptance probability and the associated underlying distributions of valid offer prices is discussed in Section 4.3. Finally, findings are summarized in Section 4.4.

4.1 Artificial Neural Networks

An artificial neural network is a mathematical model inspired by biological neural networks, which provides a general, practical method for learning real-valued, discrete-valued, and vector-valued functions over continuous and discrete-valued attributes from examples in order to facilitate regression or classification [24]. The model consists of interconnecting artificial neurons (nodes), ordered into an input layer, hidden layers, and an output layer. Each node in the hidden layers as well as the output layer computes a linear combination of its inputs. Through an activation function, a threshold may be applied to this weighted sum of inputs.

Due to the ability of an artificial neural network of capturing complex nonlinear relations, which is a useful feature in case of learning functions whose general form is unknown in advance, (3.15) through (3.17) could nicely be represented by such a model, albeit with different inputs (on-line available data). The relation between on-line available data and values of the α and γ parameters is not known in advance. The relation between on-line available data and the $\overline{n}_{dg}$ values is also unknown and could hence also be approximated using an artificial neural network in order to be able to estimate $\overline{n}_{dg}$ values on-line.

Representing the unknown relations between distribution parameters and on-line available data using artificial neural networks also brings the attractive feature of fast evaluation of these (modeled) functions, which is crucial in the TAC SCM game environment. Other advantages include robustness to noise in the training data [24], the possibility to introduce adaptivity by adjusting the weights of each node's inputs on-the-fly using newly obtained examples (if any), and the fact that artificial neural networks have proven to be useful for economic forecasts in various domains [19].

A disadvantage of artificial neural networks is the phenomenon of overfitting, which results in networks that generalize poorly to new data despite excellent performance over training data [24]. However, several methods have been developed in order to cope with this problem. For instance, cross-validation methods can be used to estimate an appropriate stopping point for the training process. Alternatively, a model can be optimized by evaluating the performance of this model on a sufficiently large, representative test dataset.

Despite the disadvantages, an artificial neural network would be a nice substitution for (3.15) through (3.17) in order to be able to estimate offer price distributions underlying customer offer acceptance probabilities on-line. When also using an artificial neural network for estimating the mean number of offers per RFQ, the complete bidding behavior could be modeled.

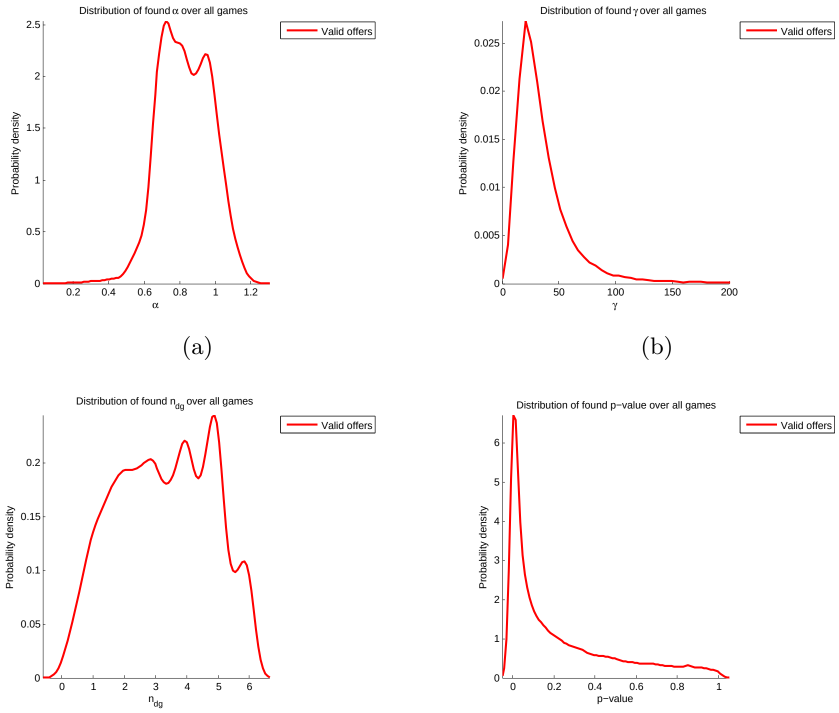
In literature, bidding behavior of agents in the TAC SCM game has already been modeled using artificial neural networks [19]. The latter approach however uses artificial neural networks for modeling the behavior of individual competing agents, whereas the approach proposed in this thesis focuses on modeling the distribution of prices offered by all competitors and the associated distribution of resulting order prices, aggregated on product level, which facilitates decision making processes involving sales quota on product levels (as is the case in the MinneTAC agent).

4.2 Incorporating Artificial Neural Networks into the Framework

In the context of the parameter estimation problem regarding the customer offer acceptance probability, an artificial neural network could be trained to produce the parameters of the distributions underlying the customer offer acceptance probability. To this end, all games in the training set should be analyzed using the framework presented in Section 3.3, which yields data on the α and γ parameters best describing the offer price distributions for each game day, for all games in the training set, on product level. Simultaneously, the mean number of offers per RFQ, $\overline{n}_{dg}$, could easily be derived for each considered sample of prices offered for RFQs for a product by a counting process.

The results of the analysis of the training set as described above can be found in Figure 4.1 and Table 4.1. Figure 4.1 visualizes distributions of the found α and γ parameter values, the associated $\overline{n}_{dg}$ values, and the associated p-values for the Kolmogorov-Smirnov test. Apparently, for over 60% of all considered samples, a log-logistic distribution described the distribution of prices sufficiently, according to the latter test (yielding a p-value higher than 0.05 in these cases). Statistics on the found values can be found in Table 4.1.

These fit parameters can then be used as examples to train the network on. The desired outputs should be presented to the network along with some other features, which are assumed to be somehow related to the outputs. Section 4.3 elaborates on such features, which are available on-line in the TAC SCM game. The constructed artificial neural network thus uses on-line available information to estimate α, γ, and $\overline{n}_{dg}$, hereby facilitating an on-line approximation of the daily customer offer acceptance probabilities per product.



(a) (b) (c) (d)

Figure 4.1: Distributions of the found α and γ parameter values for daily distributions of valid offer prices (depicted in (a) and (b) respectively), the associated mean number of offers per RFQ $\overline{n}_{dg}$ for each day d and product g (depicted in (c)), and the associated p-value for the Kolmogorov-Smirnov test (depicted in (d)), aggregated over all games in the training set.

Variable	Mean	Stdev
α	0.8296	0.1466
γ	43.5372	61.0216
$\overline{n}_{dg}$	3.2663	1.5194
p-value	0.2208	0.2616

Table 4.1: Statistics on the found α and γ parameter values for daily distributions of valid offer prices, the associated mean number of offers per RFQ $\overline{n}_{dg}$ for day d for product g, and the associated p-value for the Kolmogorov-Smirnov test, aggregated over all games in the training set.

A radial basis function network (RBFN) can be considered as a two-layer artificial neural network consisting of a hidden layer and an output layer. The activation function in each hidden unit h is a kernel function $K_h(d(x_h, x))$, the output of which approximates 0 as $d(x_h, x)$ – the (typically Euclidian) distance between an instance x (characterized by a vector of features) and the kernel center x_h – increases. The kernel functions in the hidden layer typically are Gaussians, centered at x_h with variance σ_h^2. The number of Gaussians H is subject to optimization and their centers can be determined by clustering the data, using for example the k-means algorithm [21]. The network's output for an instance x, $\hat{f}(x)$, is a linear combination of the activation units, weighted for their weights w_h, and a bias w_0 [24]:

$$\hat{f}(x) = w_0 + \sum_{h=1}^{H} w_h K_h(d(x_h, x)), \tag{4.4}$$

$$K_h(d(x_h, x)) = e^{\frac{1}{2\sigma_h^2} d^2(x_h, x)}. \tag{4.5}$$

Hence, an RBFN is a global approximation $\hat{f}(x)$ of a target function $f(x)$, represented as a linear combination of local approximations of this target function, as the contribution of each kernel is localized to a region around its center. Because RBFNs can be designed and trained in a fraction of the time it takes to train standard feed-forward backpropagation neural networks [24], a radial basis function network would be a good approximator for α, γ, and $\overline{n}_{dg}$.

However, one characteristic of the found optimal values for the γ parameter should be accounted for. The optimal values found for γ tend to be distributed on an exponential scale. This implies that the relation between the tightness of the distribution and the γ parameter appears to be exponential; because of the formulation of the double-bounded log-logistic distribution (see Section 3.3.2), the increment in γ needed to tighten the distribution increases as the distribution gets tighter. For instance, a distribution with a γ value of 2 is much more different from one with a γ value of 5 than a distribution with a γ value of 200 is from one with a γ value of 500, when all other parameters are fixed. Hence, as the required accuracy decreases for increasing γ values, the network is to be trained to predict the natural logarithm of γ.

4.3 Identifying Potentially Good Inputs

As the focus of this research is on adapting regime-based price distributions in order for them to be useful in the daily product pricing process, these regime-based price distributions should be used as artificial neural network inputs. For now, let these distributions be described by their 10th, 50th, and 90th percentile, as well as the spread of these percentiles.

As already observed in Section 3.2, order price distributions tend to differ per product type. Hence, the product type itself might be indicative of the characteristics of the order price distributions and hence the associated offer price distributions. Furthermore, offered prices might also be related to the game day, as for example in the first phase of the game, prices are more likely to be relatively high due to scarcity of products, which is caused by the fact that agents start with zero inventory.

Another good indicator for the offer price distribution of a specific product type might be the number of RFQs for that product type, as in related research, the number of simultaneously run similar auctions appeared to be affecting the revenue generated from these auctions due to their (partial) substitutivity [42]. This might be the case for the MinneTAC bidding scenario as well, as each agent is restricted by its limited resources. Even more, RFQs for the same product type could be considered to be (partial) substitutes to some of the bidders (depending on their product pricing and RFQ selection strategy). Not only the number of RFQs, but the characteristics of these RFQs as well could be indicators of the pricing behavior they generate. Hence, the mean and standard deviation of requested quantities, requested leadtimes, and reservation prices could be taken into consideration.

Literature also suggests the predictive capabilities of prices realized on the preceding day [19]. Obviously, this information is not available to the agent in a game situation. However, for each product, the minimum and maximum order prices realized on the preceding game day are available. Using these prices, the mid-range price and the spread of the prices can be determined as well. These minimum and maximum prices and their associated mid-range and spread could be considered in a double exponentially smoothed form as well, as this is likely to provide a good approximation of the mean price of the preceding game day [15, 16, 17].

Table 4.2 shows that the on-line available information considered above is indeed more or less (Pearson) correlated with the α and $\overline{n}_{dg}$ parameters. Therefore, these variables are good candidates for artificial neural network inputs. However, the on-line available information is not clearly correlated with the natural logarithm of the γ parameter. Hence, this parameter is probably harder to estimate on-line. This could also hold for $\overline{n}_{dg}$.

4.4 Summary

In this chapter, the framework presented in Chapter 3 is incorporated in an on-line product pricing approach on product level for the TAC SCM game, as required by the current design of the decision logic implemented in the MinneTAC agent. To this end, the customer offer acceptance probability for individual RFQs obtained using the framework is used to formulate a joint distribution of acceptance probability distributions for individual

Variable	α	$\ln(\gamma)$	$\overline{n}_{dg}$
todPredPerc10	0.9123	0.1546	-0.7083
todPredPerc50	0.9187	0.1532	-0.7184
todPredPerc90	0.9111	0.1544	-0.7173
todPredSpread	0.1187	0.0200	-0.1256
productNr	-0.0758	-0.0152	0.0832
day	-0.2895	-0.0885	0.0991
numRFQs	0.4136	0.0699	-0.3569
meanQuantity	0.0134	0.0037	-0.0323
stdevQuantity	0.1377	0.0032	-0.1079
meanLeadtime	-0.0137	0.0070	0.0196
stdevLeadtime	0.1327	0.0071	-0.1266
meanReservePrice	0.0376	-0.0546	0.0940
stdevReservePrice	0.1437	-0.0363	-0.1407
yestMinPrice	0.9156	0.2493	-0.6480
yestMaxPrice	0.9515	0.1738	-0.7124
yestMRPrice	0.9552	0.2121	-0.6980
yestSpread	0.4983	-0.0714	-0.4473
yestESMinPrice	0.9256	0.2400	-0.6540
yestESMaxPrice	0.9535	0.1726	-0.7179
yestESMRPrice	0.9507	0.2048	-0.6964
yestESSpread	0.6388	-0.0676	-0.5892

Table 4.2: Correlations of found values for α and the natural logarithm of γ for daily distributions of valid offer prices and their associated mean number of offers per RFQ $\overline{n}_{dg}$ with on-line available variables for all products for all days in all games in the training set.

RFQs, such that the resulting acceptance probability distribution is defined on product level. Hence, the considered acceptance probabilities still have the desired properties, which include a range in the interval $[0, 1]$ and an acceptance probability which increases as prices decrease. Therefore, the adjusted acceptance probabilities can still be verified and validated.

Product pricing can be done in a straightforward way after rewriting the acceptance probability equation, such that the optimal price is expressed as a function of the desired acceptance probability. The resulting closed-form mathematical expression contains some parameters, the determination of which requires data which is not publicly available during a TAC SCM game: offer prices. In order to enable on-line applicability of the product pricing approach, an RBFN (a type of artificial neural network) is proposed for estimating these parameters, based on on-line available information.

Such a model could adapt regime-based price distributions estimations using additional on-line available information on product type, game day, RFQ characteristics, and observable prices. However, only one out of three parameters (α, representing the median offer price) appears to be rather nicely related to on-line available information. The other two parameters, indicating the width of the offer price distribution (γ) and the mean number of expected offers on day d for product g ($\overline{n}_{dg}$), appear to be harder to approximate using on-line available data. The introduction of adaptivity could help here, as detailed in Chapter 5.

Chapter 5

Introducing Adaptivity

The product pricing approach introduced in Chapter 4 facilitates some form of adaptivity in response to changes in the environment, as the considered price distributions and related acceptance probability distributions are dependent on RFQ characteristics and observable prices, as well as on product type and game day. However, the proposed model itself is static and hence does not adapt the modeled relations between price distributions and characteristics of the environment to changing market conditions and market responses. In order to facilitate a truly adaptive, on-line applicable product pricing approach, this chapter presents several ways of introducing adaptivity into the model for estimating order price distributions and the associated customer offer acceptance probabilities. Section 5.1 discusses how the product pricing model can adapt to market disruptions. Section 5.2 deals with feeding customer responses to the product pricing strategies back into the model, in an attempt to adapt the model to the true customer offer acceptance probabilities. Findings are summarized in Section 5.3.

5.1 Coping With Market Disruptions

In [42], an English auction scenario is considered, in which bidders have independent private values, all originating from the same distribution. These private values result in bids up to the private values. The best (highest) bid wins. A method is proposed for estimating the distribution of the private values of the bidders using averaging and binary search techniques, combined with simulations. Adaptivity to market disruptions is incorporated into the model by assuming changes in bidding (and thus market disruptions) to actually be a shift in the underlying value distribution. The estimated distribution of private values is shifted accordingly.

The problem considered and the approach proposed in this thesis are somewhat similar to the scenario and approach described in [42]. In the TAC SCM game, trading agents bid on an RFQ. The best (lowest) bid wins. However, contrary to the English auction scenario, the TAC SCM RFQ bidding process much more resembles a (reverse) sealed bid, first price auction, as agents are not aware of bids of their competitors and the best bid wins. Combined with the TAC SCM game rules, this introduces the problem of limited visibility to the auctioning process: agents do not know the price resulting from an auction, unless they participate and win. Therefore, changes in bidding behavior of the competing agents cannot be observed.

However, regime information might help here, as realized prices and hence order probabilities tend to vary, depending on the economic regime [15, 16, 17]. Hence, changes in pricing behavior can be accounted for by incorporating regime information into the process of estimating order price distributions and the associated customer offer acceptance probabilities. To this end, individual parameter estimating RBFNs could be trained for each dominant regime.

For dominant regime k, the probability that a customer accepts an offer and hence places an order o_k, given price p_{dgk} for product g on game day d, can now be defined as shown in (5.1), with u the upper bound p_{dgk}. Here, let α_k and γ_k be the parameters of the log-logistic distribution underlying the offer prices, in case of dominant regime k, and $\overline{n}_{dgk}$ the associated number of offers per RFQ, averaged over all m_{dg} RFQs for product g on day d. As the MinneTAC decision logic considers five economic regimes, let $1 \leq k \leq 5$.

$$P\left(o_k|p_{dgk}\right) = 1 - \left(1 - \left(1 - \frac{\alpha_k^{-\gamma_k} - u^{-\gamma_k}}{\alpha_k^{-\gamma_k} - 2u^{-\gamma_k} + p_{dgk}^{-\gamma_k}}\right)^{\overline{n}_{dgk}}\right)^{m_{dg}},$$
$$0 < p_{dgk} < u, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk} > 0. \tag{5.1}$$

The desired sales quota q_{dg}^* for product g on game day d can now be expressed as a function of the offer price p_{dgk}^* yielding this quota, given dominant regime k. The relation between q_{dg}^* and p_{dgk}^* can be defined as

$$q_{dg}^* = 1 - \left(1 - \left(1 - \frac{\alpha_k^{-\gamma_k} - u^{-\gamma_k}}{\alpha_k^{-\gamma_k} - 2u^{-\gamma_k} + {p_{dgk}^*}^{-\gamma_k}}\right)^{\overline{n}_{dgk}}\right)^{m_{dg}},$$
$$0 < p_{dgk}^* < u, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk} > 0. \tag{5.2}$$

Equation (5.2) can be used to determine the optimal product price for each dominant regime. The suggested product prices thus generated could subsequently be weighted for the regime probabilities. To this end, let $P\left(R_{dgk}\right)$ be the probability for product g to be in regime k on game day d. The offer price p_{dg}^* for product g on game day d expected to yield a sales quota q_{dg}^*, weighted for regime probabilities, can then be approximated as shown in (5.3) and (5.4).

$$p^*_{dgk} = \left(\frac{u^{-\gamma_k} \left(\alpha_k^{-\gamma_k} \left(u^{\gamma_k} - 2\alpha_k^{\gamma_k} \right) \sqrt[\overline{n}_{dgk}]{1 - \sqrt[m_{dg}]{1 - q^*_{dg}}} + 1 \right)}{1 - \sqrt[\overline{n}_{dgk}]{1 - \sqrt[m_{dg}]{1 - q^*_{dg}}}} \right)^{-\frac{1}{\gamma_k}},$$

$$0 < q^*_{dg} < 1, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk}, u > 0, \tag{5.3}$$

$$p^*_{dg} = \sum_{k=1}^{5} P\left(R_{dgk}\right) p^*_{dgk}, \quad 0 < p^*_{dgk} < u. \tag{5.4}$$

5.2 Feeding Back Market Responses

The weights in the RBFNs could be updated on-line, based on new data. However, new training samples cannot be presented to the networks during the game, as the target values of these samples would only be available after the end of the game. Therefore, the models estimating the parameters used for the daily approximations of customer offer acceptance probabilities cannot be updated on-line.

However, these approximations can be adjusted by multiplying it by a factor representing the ratio between the number of actually received orders and the number of predicted orders, as proposed in [28]. If more orders have been received than predicted, the acceptance probability is larger than expected, to an extent equal to the ratio between received and predicted number of orders. If less orders have been received than predicted, the acceptance probability should be adjusted downwards. This ratio, which can also be referred to as an error term ϵ, enables market responses to be fed back to the model, as this ratio can be updated on-line. A smoothed error term $\widetilde{\epsilon}$ can be used in order to prevent over- or undercompensation.

Customer offer acceptance probabilities $P\left(o_k|p_{dgk}\right)$ for product g on game day d under dominant regime k range from 0 to 1. Multiplying these probabilities with the suggested ratio $\widetilde{\epsilon}_{(d-1)gk}$ (which depends on regime k and has been updated using performance information up until day $d-1$) yields corrected probabilities $P\left(o_k|p_{dgk}\right)'$ in the range $\left[0, \widetilde{\epsilon}_{(d-1)gk}\right]$. This implies that no suitable price can be found for $q^*_{dg} \geq \widetilde{\epsilon}_{(d-1)gk}$, which is an undesirable feature in case $\widetilde{\epsilon}_{(d-1)gk} < 1$. However, when the corrected customer offer acceptance probability $P\left(o_k|p_{dgk}\right)'$ is defined as

$$P\left(o_k|p_{dgk}\right)' = P\left(o_k|p_{dgk}\right)^{\widetilde{\epsilon}_{(d-1)gk}}, \quad 0 < p_{dgk} < u, \quad \widetilde{\epsilon}_{(d-1)gk} > 0, \tag{5.5}$$

offer acceptance probabilities continue to range from 0 to 1 for $0 < p_{dgk} < u$ after correction, which ensures the validity of the approach. Hence, market responses to modeled acceptance probabilities can be fed back to the model when the probability of acceptance is defined as shown in (5.6).

$$P\left(o_k|p_{dgk}\right)' = \left(1-\left(1-\left(1-\frac{\alpha_k^{-\gamma_k}-u^{-\gamma_k}}{\alpha_k^{-\gamma_k}-2u^{-\gamma_k}+p_{dgk}^{-\gamma_k}}\right)^{\overline{n}_{dgk}}\right)^{m_{dg}}\right)^{\widetilde{\epsilon}_{(d-1)gk}},$$
$$0 < p_{dgk} < u, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk}, \widetilde{\epsilon}_{(d-1)gk} > 0. \tag{5.6}$$

For each dominant regime k, (5.6) can be used to approximate the share of received orders with respect to the total number of RFQs for product g on game day d, generated by a specified price offered on all these RFQs. Similarly, a desired sales quota q_{dg}^* for product g on game day d, under dominant regime k, can be expressed in terms of the associated corrected offer price $p_{dgk}^{*'}$:

$$q_{dg}^* = \left(1-\left(1-\left(1-\frac{\alpha_k^{-\gamma_k}-u^{-\gamma_k}}{\alpha_k^{-\gamma_k}-2u^{-\gamma_k}+p_{dgk}^{*'\,-\gamma_k}}\right)^{\overline{n}_{dgk}}\right)^{m_{dg}}\right)^{\widetilde{\epsilon}_{(d-1)gk}},$$
$$0 < p_{dgk}^{*'} < u, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk}, \widetilde{\epsilon}_{(d-1)gk} > 0. \tag{5.7}$$

Using (5.7), the optimal corrected product price for each dominant regime can be determined, as shown in (5.8). When these corrected prices are subsequently weighted for their associated regime probabilities, the corrected price expected to yield the required quota can be obtained, as shown in (5.9).

$$p_{dgk}^{*'} = \left(\frac{u^{-\gamma_k}\left(\alpha_k^{-\gamma_k}\left(u^{\gamma_k}-2\alpha_k^{\gamma_k}\right)\sqrt[\overline{n}_{dgk}]{1-\sqrt[m_{dg}]{1-\sqrt[\widetilde{\epsilon}_{(d-1)gk}]{q_{dg}^*}}}+1\right)}{1-\sqrt[\overline{n}_{dgk}]{1-\sqrt[m_{dg}]{1-\sqrt[\widetilde{\epsilon}_{(d-1)gk}]{q_{dg}^*}}}}\right)^{-\frac{1}{\gamma_k}},$$
$$0 < q_{dg}^* < 1, \quad \alpha_k, \gamma_k, m_{dg}, \overline{n}_{dgk}, u, \widetilde{\epsilon}_{(d-1)g} > 0, \tag{5.8}$$
$$p_{dg}^{*'} = \sum_{k=1}^{5} P\left(R_{dgk}\right) p_{dgk}^{*'}, \quad 0 < p_{dgk}^{*'} < u. \tag{5.9}$$

The error terms should be assigned values such that under each dominant regime k, the expected customer offer acceptance probabilities $P\left(o_k|p_{(d-1)g}^{*'}\right)$ associated with an offer price $p_{(d-1)g}^{*'}$ are corrected by the unsmoothed exponential error terms to the proportion of actually received number of orders $q_{(d-1)g}$. The found error terms can subsequently be smoothed. Hence, offer price and customer response should be related as follows:

$$q_{(d-1)g} = P\left(o_k|p_{(d-1)g}^{*'}\right)^{\epsilon_{(d-1)gk}},$$
$$0 < q_{(d-1)g} < 1, \quad 0 < p_{(d-1)g}^{*'} < u, \quad \epsilon_{(d-1)gk} > 0. \tag{5.10}$$

The error terms can be smoothed using Brown linear smoothing, where for each new observation, the smoothing factor β is weighted for the associated regime probabilities in order for errors only to be attributed to the models responsible for these errors. For now, let this smoothing factor be equal to the smoothing factor used in smoothing product prices. Equations (5.11) through (5.14) describe the determination of error terms. Smoothing is done using two components, the first of which (defined in (5.12)) consists of a linear combination of the latest error (see (5.11)) and the previous first component. The second component (see (5.13)) is a linear combination of the first component and the previous second component. Then, a linear combination of the two components yields the smoothed error (see (5.14)).

$$\epsilon_{(d-1)gk} = \frac{\ln\left(q_{(d-1)g}\right)}{\ln\left(P\left(o_k | p^{*'}_{(d-1)g}\right)\right)},$$

$$0 < q_{(d-1)g} < 1, \quad 0 < P\left(o_k | p^{*'}_{(d-1)g}\right) < 1, \tag{5.11}$$

$$\widetilde{\epsilon}^{'}_{(d-1)gk} = \beta P\left(R_{(d-1)gk}\right)\epsilon_{(d-1)gk} + \left(1 - \left(\beta P\left(R_{(d-1)gk}\right)\right)\right)\widetilde{\epsilon}^{'}_{(d-2)gk}, \tag{5.12}$$

$$\widetilde{\epsilon}^{''}_{(d-1)gk} = \beta P\left(R_{(d-1)gk}\right)\widetilde{\epsilon}^{'}_{(d-1)gk} + \left(1 - \left(\beta P\left(R_{(d-1)gk}\right)\right)\right)\widetilde{\epsilon}^{''}_{(d-2)gk}, \tag{5.13}$$

$$\widetilde{\epsilon}_{(d-1)gk} = 2\widetilde{\epsilon}^{'}_{(d-1)gk} - \epsilon^{''}_{(d-1)gk}. \tag{5.14}$$

5.3 Summary

The product pricing approach as proposed in Chapter 4 is refined in this chapter, in order for it to be capable of adapting to market disruptions, which in the context of the TAC SCM game can be characterized using economic regimes. The idea is to train separate parameter estimating RBFNs for each dominant regime. This way, product prices can be determined, given dominant regimes. These product prices can subsequently be weighted for their associated regime probabilities in order to determine the optimal product price expected to yield the desired quota. The relations between price distributions and on-line available information are thus dynamically modeled, depending on economic regimes.

Structural errors in the product pricing process can be accounted for by feeding market responses to placed offers back into the product pricing model. In order for the product pricing approach to remain valid, market responses are fed back using an exponential error term, designed to transform the estimated probability of acceptance function into a function better approximating the true acceptance probability. This error term is corrected using daily observations of expected and observed acceptance probabilities, double exponentially smoothed with a smoothing factor weighted for the associated regime probabilities. Through this feedback process, the product pricing model can adapt to the true customer offer acceptance probabilities.

Chapter 6

Adaptive Regime-Based TAC SCM Product Pricing

The final framework as presented in Chapter 5 can be evaluated by implementing the approach in the MinneTAC agent for the TAC SCM game. To this end, product pricing should be done using (5.8) through (5.14). The α_k, γ_k, and $\overline{n}_{dgk}$ parameters for product g on game day d for dominant regime k are to be estimated using RBFNs.

Experiments related to training these types of artificial neural networks are described in Section 6.1. The implementation of the approach in the MinneTAC agent is detailed in Section 6.2. The performance of this upgraded MinneTAC version is benchmarked against the performance of the current model in Section 6.3. Finally, results are summarized in Section 6.4.

6.1 Training the Artificial Neural Networks

As argued in Section 5.1, for each dominant regime k, an RBFN needs to be trained for estimating the α_k, γ_k, and $\overline{n}_{dgk}$ parameters for product g on game day d, using the inputs discussed in Section 4.3 as predictors. Therefore, training and test datasets must be split into datasets per dominant regime. This dominant regime is the dominant regime of the game day, as identified by the current regime model using an exponentially smoothed predicted price as input (as would be the case on-line).

In this case, training data consists of the data resulting from fitting price distributions on historical data of games in the training set, as presented in Section 4.2, along with the associated values for the predictors. After analyzing the test set (specified in Section 1.4) using the framework presented in Section 3.3, the performance of trained models can be evaluated on the games in the test set, as this set is sufficiently large and representative [24]. An average training dataset thus generated contains over 15,000 samples, whereas an average test dataset contains over 8,000 samples.

Using Weka [43], the RBFNs can be trained relatively easily. The results can subsequently be saved as serialized Java objects, which enables them to be used in Java software like the MinneTAC agent. One drawback of using Weka is that the Weka implementation of an RBFN, `RBFNetwork`, can only have one output. Hence, a network is to be trained per dominant regime per parameter.

Some parameters can be adjusted in the `RBFNetwork` implementation. First of all, one can define the random seed used in the clustering process used to determine the centers of the Gaussians in the networks. Let this cluster seed be 0 for all networks. A so-called ridge value can also be specified. This value indicates how much the regression error in estimating model parameters may diverge from the least squares measure. For all networks, this value is left at its default value, 1E-08. Other parameters are the number of clusters and the minimum standard deviation of these clusters.

The configurations of the latter two parameters can be determined by systematically evaluating all combinations of different values. The configurations yielding the lowest RMSD (see (3.1)) on the test set are selected [24]. The optimal number of clusters could be anything between relatively small and rather large. Using too many clusters would cause the model to not generalize very well. Hence, taking into account the size of the dataset, the set of number of clusters considered is $\{25, 50, 100, 150, 200, 300\}$. Depending on the number of clusters and the dataset, the minimum standard deviation of the clusters could also be high or low. Hence, standard deviations in the set $\{1, 2, 5, 10, 15\}$ are considered. Apparently, α_k can be estimated relatively well, whereas $\overline{n}_{dgk}$ and $\ln(\gamma_k)$ cannot (see Table 6.1).

Parameter	Regime	Clusters	MinStdev	RMSD
α_k	1	25	15	0.0448
α_k	2	50	10	0.0346
α_k	3	100	5	0.0366
α_k	4	50	5	0.0386
α_k	5	300	5	0.0400
$\ln(\gamma_k)$	1	100	15	0.7713
$\ln(\gamma_k)$	2	150	5	0.6903
$\ln(\gamma_k)$	3	150	5	0.6481
$\ln(\gamma_k)$	4	200	5	0.6370
$\ln(\gamma_k)$	5	150	2	0.6732
$\overline{n}_{dgk}$	1	50	15	1.0036
$\overline{n}_{dgk}$	2	25	5	1.0773
$\overline{n}_{dgk}$	3	200	5	0.9974
$\overline{n}_{dgk}$	4	300	2	0.9395
$\overline{n}_{dgk}$	5	100	5	0.8090

Table 6.1: Optimized configuration of number of clusters and minimum standard deviation of clusters for radial basis function networks for estimating the α parameter, the natural logarithm of the γ parameter, and the $\overline{n}_{dg}$ parameter, given a dominant regime k. The table also quantifies the root mean squared deviation (RMSD) of parameter values predicted by the trained models from the parameter values in the test set.

6.2 Implementation in the MinneTAC Agent

In an attempt to improve the product pricing process by combining regime information with other on-line available information, the sales model of the 2008 MinneTAC configuration as described in Section 2.3.2 is replaced with a system designed for Product Pricing using Adaptive Real-time Regime-based Probability of Acceptance Estimations: PPARRPAE. The algorithm (described in Algorithm 1) involves parameter estimation using the RBFNs discussed in Section 6.1 and subsequently pricing products using (5.8) and (5.9). An error term is also considered, following (5.11) through (5.14). Figure 6.1 visualizes the relations between logical components involved in this process.

The main idea is to leave the regime model intact and to build an adapter, which combines the characteristics of the price distribution estimated by the regime model with characteristics of RFQs, as well as with more detailed information on prices, the day, and the considered product (see Section 4.3). Using the RBFNs trained in Weka, the adapter transforms this data into a parameterized customer offer acceptance probability distribution function per dominant regime and assigns weights to these distributions, equal to their associated regime probabilities.

```
foreach d in days do
    foreach g in products do
        // Update error using last feedback, following
        // (5.11) through (5.14)
        error = updateError(getFeedback(d − 1, g));
        // Retrieve product-level data from regime model
        regProbs = getRegProbs(d, g);
        regPriceDistr = getRegPriceDistr(d, g);
        trends = getTrends(d, g);
        // Estimate parameters using RBFNs, as detailed
        // in Section 6.1
        priceDistr = estParams(regPriceDistr, getData(d, g));
        // Determine median price using (5.8) and (5.9)
        median = priceForProb(0.5, priceDistr, error, regProbs);
        // Retrieve allocated quota
        quota = getQuota(d, g, median, trends);
        // Determine optimal price expected to yield quota
        // using (5.8) and (5.9)
        price = priceForProb(quota, priceDistr, error, regProbs);
        // Bid optimized price on selected RFQs
        priceProduct(d, g, price);
    end
end
```

Algorithm 1: Product pricing in the PPARRPAE approach.

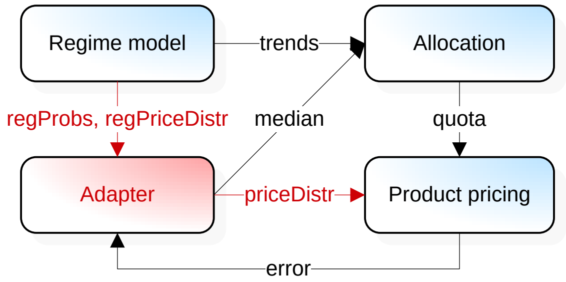


Figure 6.1: Simplified schematic overview of the proposed PPARRPAE sales process configuration for an arbitrary product on an arbitrary game day.

This adapted distribution can subsequently be used in the product pricing process for an arbitrary product on an arbitrary game day, as proposed in Chapter 5. Given a quota specified by the allocation component, the product pricing component uses the adapter to compute the price expected to yield this quota per dominant regime and weighs the suggested prices for their associated regime probabilities. The thus generated optimal price is then offered on all selected RFQs for the considered product. The market responses to these offers are directly fed back to the adapter, which learns from its errors (see Section 5.2). Therefore, in order for this information not to be biased, interval randomization is not applied on the generated optimal price, as opposed to the existing approach (see Section 2.3.2).

The allocation model bases its decisions among other things on price predictions, which consist of an estimate of the median price of the considered game day and trends representing expected future deviations from this median. In the existing sales model, the trends are estimated using the regime model, whereas the median is estimated using a price-follower approach. This price-following component is also used in the estimation of the daily probability of acceptance function and can thus be updated using market responses. Since in the proposed approach, market responses are not related to the price-following median, but are fed back to the adapter, the prediction of the median price should in this case be provided to the allocation component by the adapter.

6.3 Performance in the TAC SCM Game

By running and analyzing a number of games, the performance of the PPARRPAE system proposed in Section 6.2 can be compared with the benchmark configuration of the MinneTAC agent, which implements the regime model for price trend prediction and a price-following approach for estimating median prices, as detailed in Section 2.3.2. In Section 6.3.1, the experimental set-up for analyzing the performance of the proposed model is described. The results of these experiments are presented in Section 6.3.2.

6.3.1 Experimental Set-Up

In this experimental set-up, games are in accordance with the TAC SCM game specifications of 2006 [6], as discussed in Section 2.2. The randomness incorporated in several facets of the game is an inconvenient characteristic for a testing environment in which two approaches are to be compared, as this randomness in market conditions implies that many experiments should be run in order to obtain results with any statistical significance.

The issue of randomness in the testing environment is tackled by a controlled TAC SCM game server, in which random seeds used for generating market conditions can be controlled. Random elements in decision processes of competing agents cannot be controlled. Hence, multiple runs with the same random seeds for market conditions could still yield different results. However, under controlled market conditions, such uncontrolled stochastic behavior of participating trading agents does not have a significant impact on the agent profit levels [34]. The results presented in [34] also indicate that most significant profit differences between agents can already be detected in approximately 40 games.

The performance of the PPARRPAE system can hence be evaluated in 40 experiment sets on a controlled server. Each experiment set consists of a paired evaluation of the performance of the benchmark and the PPARRPAE system under equal market characteristics. For now, let the competitors be Dummy agents, in order for the potential of change in performance to be as apparent as possible, and not to be (partially) concealed by complex behavior of other competitors.

In each evaluation, the final bank account balance can be considered, as well as the sales performance. To this end, the mean and standard deviation of account balances over all games can be computed. Furthermore, the overall deviation of the final account balance of PPARRPAE with respect to the benchmark is to be analyzed. The number of obtained orders should be considered in the analysis as well. As for the sales performance, the RMSD of desired acceptance probabilities to acceptance probabilities derived from actual responses to set product prices can be analyzed. This error can be put into context when the number of times the market is polled – i.e., the number of times the agent proceeds to actually bidding on RFQs, given an expected probability of acceptance – is analyzed as well.

Performance differences should also be assessed with respect to their statistical relevance. This can be done with a paired, two-sided Wilcoxon signed-rank test. This is a non-parametric test, which tests the hypothesis that the differences between paired observations are symmetrically distributed around a median equal to 0. If this null hypothesis is rejected (at a significance level below 0.05), the compared sets of samples can be assumed to be significantly different. This test would be suitable in this experimental set-up, as the distribution of the values to be compared is unknown.

6.3.2 Experimental Results

Over all experiments, the benchmark configuration of the MinneTAC trading agent usually outperforms the Dummy agents with respect to final bank account balance. However, this configuration on average does not manage to win more bidding processes than its competitors. PPARRPAE outperforms the benchmark with respect to final bank account balance and the number of obtained orders.

Tables 6.2 and 6.3 and Figure 6.2 support these observations, as they indicate a clear deviation between the benchmark and the PPARRPAE approach; generally, PPARRPAE yields higher profits and more orders, both values of which tend to be more stable over all games run. This deviation is quantified per game in Tables A.4 and A.5 in Appendix A. Table 6.4 summarizes the overall deviation. Here, the p-value for the Wilcoxon test for both final account balances and number of orders obtained confirms the significance of the difference in performance.

	Benchmark		PPARRPAE	
Agent	Mean	Stdev	Mean	Stdev
MinneTAC	19.2614	12.4207	49.3933	2.7053
Dummy	12.9194	3.2799	14.0436	2.9310
Dummy-2	13.0250	3.3152	14.1313	3.1668
Dummy-3	12.7687	3.3184	14.1034	2.7711
Dummy-4	12.8552	3.4148	14.3529	2.9034
Dummy-5	13.0803	3.2224	14.2307	2.9874

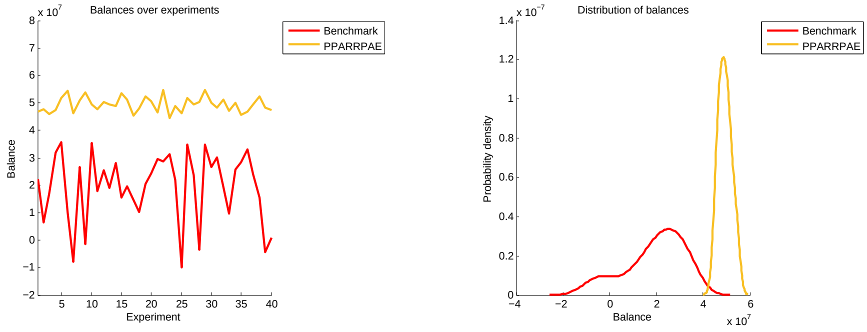
Table 6.2: Mean and standard deviation of final bank account balance per agent, calculated over all experiments. Values are expressed in millions.

	Benchmark		PPARRPAE	
Agent	Mean	Stdev	Mean	Stdev
MinneTAC	3.0865	1.0178	4.6474	0.4507
Dummy	3.2615	0.3367	3.0498	0.3129
Dummy-2	3.2615	0.3291	3.0571	0.3448
Dummy-3	3.2452	0.3386	3.0364	0.3267
Dummy-4	3.2198	0.3292	3.0387	0.3172
Dummy-5	3.2504	0.3480	3.0437	0.3301

Table 6.3: Mean and standard deviation of number of obtained orders per agent, calculated over all experiments. Values are expressed in thousands.

Statistic	Balance	Orders
Deviation	30.1319	1.5608
Relative deviation	1.5644	0.5057
Wilcoxon p-value	0.0000	0.0000

Table 6.4: Overall deviation of values for final bank account balance and number of obtained orders of the PPARRPAE approach, compared to the benchmark. Balance deviation is expressed in millions, whereas the deviation in the number of obtained orders is expressed in thousands.



(a) (b)

Figure 6.2: Final bank account balances for the two considered MinneTAC variants over all experiments. The separate values for each experiment are depicted in (a), whereas (b) shows final bank account balance distributions.

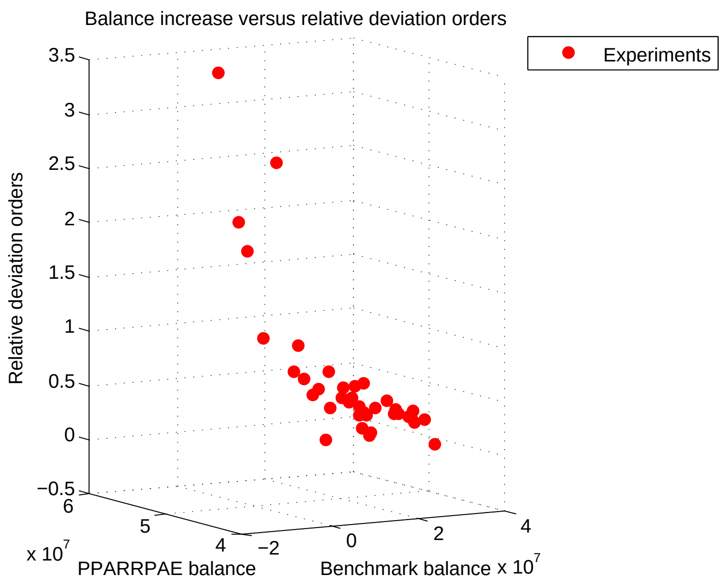


Figure 6.3: Increase in profit of the PPARRPAE approach compared to the benchmark, depicted in relation to the associated relative increase in the number of obtained orders per experiment.

Statistic	RMSD PA	Market polls
Deviation	-0.0011	726.0500
Relative deviation	-0.0024	0.7889
Wilcoxon p-value	0.8297	0.0000

Table 6.5: Overall deviation of values for the root mean squared deviation of estimated probability of acceptance (RMSD PA) and the number of market polls of the PPARRPAE approach, compared to the benchmark.

With PPARRPAE, final bank account balances significantly exceed those of the benchmark in each experiment; balances increase with about 160%. Overall, the number of orders obtained with PPARRPAE significantly exceeds the number of orders obtained by the benchmark with over 50%. Figure 6.3 visualizes the relation between relative deviation of the number of obtained orders and the associated balance increase for each experiment. Balance increase does not appear to be fully explained by an increase in obtained orders; small increases or decreases can also yield higher balances.

With respect to sales performance, the RMSD of desired probabilities of acceptance to realized probabilities, derived from actual responses to set product prices, can be analyzed as well. Table 6.5 summarizes the overall extent to which the error of the model for estimating acceptance probabilities implemented in PPARRPAE deviates from the error in acceptance probability estimates by the benchmark configuration of MinneTAC. Statistics per experiment can be found in Table A.6 in Appendix A. Apparently, the acceptance probability estimation for products actually bid for is not significantly improved by PPARRPAE.

Table 6.5 also summarizes deviations in the number of times the considered MinneTAC configurations actually bid on one or more RFQs for a specific product type and hence are able to poll the market in order to check whether the estimated probability of acceptance is correct. Statistics per experiment can be found in Table A.7 in Appendix A. Overall, the PPARRPAE approach appears to lead to a significant approximately 80% increase of cases in which the estimated acceptance probabilities are good enough for the trading agent to take a chance.

6.4 Summary

In this chapter, the adaptive regime-based product pricing approach proposed in Chapter 5 is implemented and tested in the context of the TAC SCM game. To this end, following suggestions done in Section 5.1, RBFNs are first of all trained for estimating the α_k, $\ln(\gamma_k)$, and $\overline{n}_{dgk}$ parameters for product g on game day d per dominant regime k, using the inputs discussed in Section 4.3 as predictors. The α_k parameter can be estimated relatively well, but $\overline{n}_{dgk}$ and especially $\ln(\gamma_k)$ cannot easily be closely approximated.

The sales model of the 2008 MinneTAC configuration as described in Section 2.3.2 is then replaced with a system designed for Product Pricing using Adaptive Real-time Regime-based Probability of Acceptance Estimations: PPARRPAE. This system basically implements (5.8) through (5.14), the parameters of which are estimated using the trained RBFNs. By doing so, the price distribution estimated by the regime model is transformed into parameterized customer offer acceptance probability distribution estimates per dominant regime, based on on-line available data. When pricing products, these distributions are used in a dynamic combination, which depends on estimated economic regime probabilities. Also, these distributions are corrected using a dynamically updated error term.

The MinneTAC configuration implementing the PPARRPAE approach is benchmarked against the current MinneTAC configuration by a paired evaluation of the performance of these approaches in 40 TAC SCM games run on a controlled server, hereby using fairly non-complex Dummy agents as competitors. PPARRPAE turns out to outperform the benchmark with respect to final bank account balance and the number of obtained orders. Final balances significantly increase with about 160% and the number of orders significantly increases with over 50% with respect to the benchmark. The acceptance probability estimation for products actually bid for does not appear to be significantly improved by PPARRPAE, but using the PPARRPAE approach significantly increases the number usable acceptance probability estimations with approximately 80%.

Chapter 7

Discussion

The relation between the product pricing process and daily price distributions is modeled in Chapter 3. The main idea is that product pricing can be done by taking into consideration the probability that an offered price is accepted by a customer. This probability of acceptance function can be regarded as the inverse cumulative density function of all order prices, as this cumulative density function yields the fraction of order prices realized at or below a specific value, which thus is similar to the probability that an order is placed with another agent offering a similar or better deal.

The proposed approach involves explicit modeling of the distribution of normalized order prices of a game day, as opposed to current TAC SCM product pricing approaches, which directly model acceptance probabilities, explicitly model individual competitor's behavior, or do not explicitly model any sales side behavior at all. Daily distributions of valid offer prices should provide a close approximation of order price distributions, as the latter are distributions of the minimum of the former distributions. Historical game data apparently contains sufficient data to fit daily offer price distributions, as opposed to order price data, which suffers from sparsity and low dispersion. Hence, in the proposed approach, the daily order price distribution is modeled using the daily distribution of valid offer prices.

The log-logistic distribution assumed to be underlying the offer prices (and thus the order prices) is defined in the domain $[0, \infty]$, whereas order prices are only expected to be in the range $[0, 1.25]$. Hence, product pricing using the customer offer acceptance probabilities modeled using this distribution might yield normalized prices higher than 1.25, which is an undesirable feature. After truncating the log-logistic distribution into a double-bounded distribution defined in the domain $[0, 1.25]$, it still sufficiently captures the real distribution of these prices most of the time, according to the Kolmogorov-Smirnov test results.

In order for the product pricing approach to be applicable on-line, artificial neural networks are proposed in Chapter 4. As the relation between product pricing and daily price distribution is modeled in a parametric way, these neural networks can be used for real-time approximation of the parameters of the acceptance probability distribution, using on-line available information. However, the performance of these networks is somewhat disappointing. One out of three parameters (representing the median offer price) can be estimated rather nicely, but the other two parameters, indicating the width of the offer price distribution and the number of expected offers, appear to be hard to approximate using on-line available data.

Limited on-line available information thwarts the introduction of adaptivity in Chapter 5 as well. Observing changes in bidding behavior of competitors is argued to be valuable for estimating price distributions, but the TAC SCM game specifications obstruct agents from doing so. Therefore, regime information is incorporated into the model, as realized prices and hence order probabilities tend to vary with the economic regime. The relations between price distributions and on-line available information are dynamically modeled; using regime probabilities, the overall model uses a mix of models, each of which has been trained for a dominant regime. This enables the model to adapt to market disruptions by adopting different behavior, depending on the expected market conditions.

However, structural errors in the parameter estimating models are not corrected in this way. An obvious way to introduce adaptivity to each individual neural network used for parameter estimation would be updating the internal weights in these networks, but this is impossible due to the lack of on-line available information on the true parameters of daily price distributions. Therefore, for each dominant regime, the probability of acceptance approximations based on parameter estimations of daily offer price distributions are corrected using an error term. This error term is updated on-line using approximations of the real probability of acceptance, based on market responses to offered prices. This way, instead of correcting each parameter individually, the complete distribution is reshaped by one error term.

The thus obtained approach of product pricing using adaptive regime-based probability of acceptance estimations is implemented in the MinneTAC agent in Chapter 6, such that it replaces the current price-following approach of probability of acceptance estimation. Therefore, product pricing is done on product level, as the price expected to yield the desired demand for that product is calculated using a quota (desired acceptance probability) on product level. Product pricing on RFQ level might improve the performance, as acceptance probabilities could then be estimated or fine-tuned on RFQ level. However, the desired acceptance probability should then be provided for each individual RFQ. This would require a redesign of the allocation process as well and hence falls outside the scope of this thesis.

The new pricing approach is benchmarked against the current approach. The significance of differences in performance is assessed using the Wilcoxon test. Even without a mechanism to price on RFQ level, the new product pricing approach yields a significantly higher final bank account balance than the current approach does; profits are more than doubled. The performance of the newly proposed approach appears to be more stable than the current approach as well, as the standard deviation is lower and the profit shows less fluctuation over all experiments.

The performance boost may be explained by the observation that the new approach yields significantly more orders, provided that the prices associated with these orders are high enough. However, the profit increase can only be explained to a certain extent by the increase in the number of obtained orders. In some experiments, a small increase (or even a decrease) in the number of obtained orders still results in doubled profits. This indicates that not only more orders are obtained, but orders are better priced as well. This could be caused by prices of obtained orders to be closer to second-lowest prices, instead of being significantly lower, which results in a reduced margin between customers' reserve prices and realized order prices.

One observation appears to contradict this hypothesis: the quality of probability of acceptance estimations is not significantly improved at all. The key here is in the realization of this statistic. Each game day, the agent estimates a probability of acceptance function for each product and bases its product pricing decisions on that function; the function generates the optimal price, which is assumed to yield the desired demand. If this generated price is higher than the reservation prices associated with the RFQs considered by the agent, the agent does not bid, as the approximation of probability of acceptance would yield an invalid offer in this case.

Hence, each time that the agent does not place any bids on RFQs for a particular product, even though a quota has been allocated, could be interpreted as an indication of a very unrealistic acceptance probability estimation. These occasions do however not directly show up in the statistics, as the magnitude of the error in the acceptance probability estimation cannot be determined, because the estimation is never put to the test in the market. Luckily, the observation that the agent implementing the new product pricing approach polls the market a significant 70% more often could imply that the estimated acceptance probabilities are more often good enough for the trading agent to take a chance.

This does however not explain why acceptance probability estimations that are put to the test in the market are not significantly improved at all. This lack of improvement may be caused by a structural error inherent to the characteristics of the sales process. The true acceptance probability associated with an offered price is assumed to be the ratio of the total quantity of obtained orders to the total quantity associated with the RFQs offered on. Within the TAC SCM game, it is impossible to agree to only partially commit to fulfill a customer's request. Hence, the true acceptance probability tends to be discrete, which makes it hard to match with a desired acceptance probability defined in a continuous approximation of reality.

Despite the discussed imperfections, the in-game performance against fairly non-complex competitors of the approach proposed in this thesis is very promising. The evident performance improvement induced by this approach will hopefully also become apparent when competing against more complex trading agents.

Chapter 8

Conclusions

In the context of product pricing, daily price distributions can be taken into account. In this thesis, the relation between product pricing and price distributions is modeled by a parametric price distribution estimation approach, in which a double-bounded log-logistic function is assumed to be underlying the prices offered (and hence the related order prices) for a product on an arbitrary day. The inverse of the cumulative order price distribution thus approximated can be considered as an estimation of a customer offer acceptance probability function. Using this closed form mathematical expression, the price expected to yield the desired sales quota can easily be computed.

Artificial neural networks can be trained on historical data to determine the parameters of the price distributions in real-time on a daily basis. These networks base their approximations on a price distribution estimated by MinneTAC's regime model, which is based on a Gaussian Mixture Model. Using additional information on product type, game day, RFQ characteristics, and observable prices, the networks transform this distribution into a parametric price distribution and related acceptance probability estimate, hereby facilitating a real-time applicable product pricing process.

The price distribution and related customer offer acceptance probability function thus estimated can be adapted to dynamic market characteristics by incorporating regime information and by introducing an error term, which is updated on-line. Regime information can be incorporated by training separate models per dominant regime. Each game day, outputs as well as error corrections can then be weighted for the associated regime probabilities.

Implementation of the proposed approach in the MinneTAC trading agent leads to promising results in the TAC SCM game. Although apparently not capable of significantly reducing a possibly structural error in customer offer acceptance probabilities used in actual product pricing processes, the proposed approach does yield probability of acceptance estimations of acceptable quality more often than the current price-following product pricing model. This approximately 80% increase in the number of useable estimations results in significantly more obtained orders: the number of bidding processes won turns out to increase with over 50%, when the proposed approach is compared to the current approach under equal market conditions. Even more, obtained orders appear to be associated with higher prices, which implies that margins between customers' reservation prices and realized order prices are reduced due to exploitation of better estimations of behavior of bidding parties, in relation to the customers' reservation prices.

The observed changes yield an evident performance improvement: in the considered experimental set-up with fairly non-complex competitors, final bank account balances and hence profits are overall more than doubled. This performance improvement appears to be structural; in every single experiment, the new approach outperforms the benchmark with respect to profits as well as the number of orders obtained. Moreover, a decreased standard deviation indicates that the MinneTAC agent implementing the approach proposed in this thesis performs more consistently than the benchmark.

Hence, economic regime estimations, which characterize market conditions, turn out to contribute to profit maximization when they are used to differentiate product pricing strategies. To this end, when product pricing strategies are linked to price distribution estimations taking into account on-line available information, the relation between this information and the distribution estimates should depend on economic regimes.

Even though the performance of the proposed model already is very promising, some aspects still require more research. First of all, the type and parameterization of models for on-line price distribution and acceptance probability approximation could be revised, as the artificial neural networks used in this research do not perform very satisfactorily. To this respect, other possible predictors for acceptance probabilities could be considered as well. These predictors do not necessarily have to be data directly related to sales; procurement information might be a good candidate, as costs associated with specific orders could theoretically easily influence the price, dependent on the cost allocation applied in the participating trading agents.

Another option for future research is in the design of the MinneTAC trading agent. For instance, a model for product pricing on RFQ level instead of on product level could be taken into consideration. Furthermore, as the adapter introduced in this thesis appears to perform so well, the adapter could maybe be extended in order for it to encapsulate the entire regime model, which could improve the quality of predicted price distributions and price trends. Also, the improved acceptance probability estimations could be used in the allocation or RFQ selection process.

Finally, the approach of product pricing using adaptive regime-based acceptance probability estimations proposed in this thesis could be challenged in a situation with strong competitors with complex decision logics. If the MinneTAC agent could deal with those agents as with the agents considered in this research, MinneTAC would be more competitive than ever.

Bibliography

[1] Apache Excalibur project. Apache Excalibur and Fortress IOC Container. Website, 1997-2008. Available online, `http://excalibur.apache.org/`; last visited January 13, 2008.

[2] R. Becker, S. Hurn, and V. Pavlov. Modelling Spikes in Electricity Prices. *The Economic Record*, 82(263):371–382, 2007.

[3] M. Benisch, A. Greenwald, I. Grypari, R. Lederman, V. Naroditskiy, and M. Tschantz. Botticelli: A Supply Chain Management Agent Designed to Optimize under Uncertainty. *ACM SIGecom Exchanges*, 4(3):29–37, 2004.

[4] M. Benisch, A. Sardinha, J. Andrews, and N. Sadeh. CMieux: Adaptive Strategies for Competitive Supply Chain Trading. *ACM SIGecom Exchanges*, 6(1):1–10, 2006.

[5] T. Coleman and Y. Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[6] J. Collins, R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. The Supply Chain Management Game for the 2006 Trading Agent Competition. Technical report, Carnegie Mellon University, November 2005.

[7] J. Collins, W. Ketter, and M. Gini. Flexible decision control in an autonomous trading agent. *Electronic Commerce Research and Applications*, Forthcoming 2009.

[8] P. Dasgupta and Y. Hashimoto. Multi-Attribute Dynamic Pricing for Online Markets Using Intelligent Agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 277–284, Washington, DC, USA, 2004. IEEE Computer Society.

[9] G. Ghiani, G. Laporte, and R. Musmanno. *Introduction to Logistics Systems Planning and Control*. John Wiley & Sons, Ltd, 2004.

[10] J. D. Hamilton. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57(2):357–384, March 1989.

[11] R. Hathaway and J. Bezdek. Switching Regression Models and Fuzzy Clustering. *IEEE Transactions on Fuzzy Systems*, 1(3):195–204, 1993.

[12] M. He, A. Rogers, E. David, and N. R. Jennings. Designing and Evaluating an Adaptive Trading Agent for Supply Chain Management Applications. In H. L. Poutré, N. Sadeh, and S. Janson, editors, *IJCAI Workshop on Trading Agent Design and Analysis*, pages 35–42. Springer, 2005.

[13] R. Huisman and R. Mahieu. Regime Jumps in Electricity Prices. *Energy Economics*, 25(5):425–434, September 2003.

[14] A. S. Kapadia, W. Chan, and L. A. Moyé. *Mathematical Statistics with Applications*. CRC Press, 2005.

[15] W. Ketter, J. Collins, M. Gini, A. Gupta, and P. Schrater. Identifying and Forecasting Economic Regimes in TAC SCM. In H. L. Poutré, N. Sadeh, and S. Janson, editors, *AMEC and TADA 2005, LNAI 3937*, pages 113–125. Springer Verlag Berlin Heidelberg, 2006.

[16] W. Ketter, J. Collins, M. Gini, A. Gupta, and P. Schrater. A Predictive Empirical Model for Pricing and Resource Allocation Decisions. In *Proceedings of the Ninth International Conference on Electronic Commerce (ICEC 2007)*, pages 449–458, Minneapolis, MN, USA, 2007.

[17] W. Ketter, J. Collins, M. Gini, A. Gupta, and P. Schrater. Detecting and Forecasting Economic Regimes in Multi-Agent Automated Exchanges. *Decision Support Systems*, Forthcoming 2009.

[18] C. Kiekintveld, M. P. Wellman, S. P. Singh, J. Estelle, Y. Vorobeychik, V. Soni, and M. R. Rudary. Distributed Feedback Control for Decision Making on Supply Chains. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 384–392, 2004.

[19] Y. Kovalchuk and M. Fasli. Adaptive Strategies for Predicting Bidding Prices in Supply Chain Management. In *Proceedings of the Tenth International Conference on Electronic Commerce (ICEC 2008)*, Innsbruck, Austria, 2008.

[20] C. Li, J. Giampapa, and K. Sycara. Bilateral Negotiation Decisions with Uncertain Dynamic Outside Options. In *First IEEE International Workshop on Electronic Contracting*, pages 54–61, Washington, DC, USA, July 2004. IEEE Computer Society.

[21] J. MacQueen. Some Methods of Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[22] E. Mamdani and S. Assilian. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

[23] C. Massey and G. Wu. Detecting Regime Shifts: The Causes of Under- and Overreaction. *Management Science*, 51(6):932–947, June 2005.

[24] T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997.

[25] A. Mood, F. Graybill, and D. Boes. *Introduction to the theory of statistics*. McGraw-Hill, 1974.

[26] T. D. Mount, Y. Ning, and X. Cai. Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters. *Energy Economics*, 28(1):62–80, January 2006.

[27] A. Murari, G. Vagliasindi, M. Zedda, R. Felton, C. Sammon, L. Fortuna, and P. Arena. Fuzzy Logic and Support Vector Machine Approaches to Regime Identification in JET. *IEEE Transactions on Plasma Science*, 34:1013–1020, June 2006.

[28] D. Pardoe and P. Stone. Predictive Planning for Supply Chain Management. In *Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling*, pages 21–30, June 2006.

[29] D. Pardoe and P. Stone. TacTex-05: A Champion Supply Chain Management Agent. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 1489–94, July 2006.

[30] S. Saha, A. Biswas, and S. Sen. Modeling Opponent Decision in Repeated One-Shot Negotiations. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 397–403, New York, USA, 2005. ACM.

[31] SICS. Game History for tac3.sics.se. Website, 2004-2008. Available online, `http://tac3.sics.se:8080/tac3.sics.se/history/`; last visited March 14, 2008.

[32] SICS. Game History for tac5.sics.se. Website, 2004-2008. Available online, `http://tac5.sics.se:8080/tac5.sics.se/history/`; last visited March 14, 2008.

[33] R. Slonim and E. Garbarino. The Effect of Price History on Demand as Mediated by Perceived Price Expensiveness. *Journal of Business Research*, 45(1):1–14, 1999.

[34] E. Sodomka, J. Collins, and M. L. Gini. Efficient Statistical Methods for Evaluating Trading Agent Performance. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 770–775, Vancouver, British Columbia, Canada, 2007. AAAI Press.

[35] S. Y. Sohn, T. H. Moon, and K. J. Seok. Optimal Pricing for Mobile Manufacturers in Competitive Market Using Genetic Algorithm. *Expert Systems with Applications*, 36(2):3448–3453, 2009.

[36] S. Srinivasan, P. T. L. P. Leszczyc, and F. M. Bass. Market Share Response and Competitive Interaction: The Impact of Temporary, Evolving and Structural Changes in Prices. *International Journal of Research in Marketing*, 4:281–305, December 2000.

[37] M. Stan, B. Stan, and A. M. Florea. A Dynamic Strategy Agent for Supply Chain Management. In *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 227–232, Washington, DC, USA, 2006. IEEE Computer Society.

[38] C. S. Tang and R. Yin. Responsive Pricing under Supply Uncertainty. *European Journal of Operational Research*, 182(1):239–255, October 2007.

[39] The MathWorks. *Optimization Toolbox User's Guide*, 4th edition, October 2008.

[40] University of Minnesota. Game History for tac01.cs.umn.edu. Website, 2008. Available online, `http://tac01.cs.umn.edu:8080/tac01.cs.umn.edu/history/`; last visited July 17, 2008.

[41] University of Minnesota. Game History for tac02.cs.umn.edu. Website, 2008. Available online, `http://tac02.cs.umn.edu:8080/tac02.cs.umn.edu/history/`; last visited July 17, 2008.

[42] W. E. Walsh, D. C. Parkes, T. Sandholm, and C. Boutilier. Computing Reserve Prices and Identifying the Value Distribution in Real-world Auctions with Market Disruptions. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1499–1502, 2008.

[43] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Fransisco, 2nd edition, 2005.

Appendices

Appendix A

Detailed Experimental Results

This appendix contains tables with results from the experiments described in Chapters 3, 4, and 6. These results are specified per game or experiment.

	RFQs		Offers		Minimum offers		Orders	
Game	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
9323tac5	12.0764	6.3269	32.1651	17.9330	11.2807	6.1414	9.8185	5.1603
9324tac5	9.4676	5.2862	30.2744	18.2357	9.1668	5.2918	8.8060	5.0906
9325tac5	11.5886	6.0218	34.3358	18.1536	10.8798	5.7728	10.0043	5.2015
9326tac5	12.8906	6.6669	34.0080	20.1186	11.9830	6.5375	10.5736	5.6623
9327tac5	10.3804	6.0814	33.6080	17.4197	9.8733	5.6831	9.1602	4.9966
7308tac3	13.3730	6.0040	40.8477	23.0691	11.7756	5.7080	9.9813	4.8246
7309tac3	13.4250	6.2315	43.1662	26.2167	12.2449	6.2381	9.7406	5.2744
7310tac3	12.2420	6.2338	41.3926	25.0991	11.3315	6.0380	9.9023	5.3857
7311tac3	11.1483	6.2349	46.5145	28.0801	10.6946	6.0885	10.2770	5.9160
7312tac3	11.6991	5.7714	45.1634	26.2521	10.9125	5.7379	9.7923	5.2786
763tac02	10.5429	6.4178	20.4696	14.3590	8.3224	5.0486	8.3224	5.0486
764tac02	12.9278	6.1818	19.4688	13.6319	9.0071	4.9651	9.0071	4.9651
765tac02	11.5832	6.3899	26.2659	17.6138	9.5537	5.4567	9.5537	5.4567
766tac02	13.8310	6.1400	27.0628	15.5303	10.2844	4.8285	10.2844	4.8285
767tac02	10.7131	6.3423	24.2165	16.3958	8.5793	5.1237	8.5793	5.1237
768tac02	11.4946	6.0492	26.7875	18.5988	9.4722	5.4665	9.4722	5.4665
794tac01	10.2946	6.1801	36.7739	23.0839	9.0227	5.5712	9.0227	5.5712
795tac01	10.0074	5.6986	41.2977	26.0354	9.4071	5.4980	9.4071	5.4980
796tac01	11.1804	6.5138	38.3506	24.7362	9.9457	6.0086	9.9457	6.0086
797tac01	11.3318	6.1276	41.8895	26.1851	10.0875	5.6947	10.0875	5.6947
798tac01	8.4523	4.6021	37.9932	20.6528	8.1608	4.5408	8.1608	4.5408
799tac01	9.4216	6.0347	36.6420	22.4467	8.5324	5.3909	8.5324	5.3909
Aggregated	11.3669	6.2385	34.4861	22.6736	10.0235	5.7305	9.4741	5.3440

Table A.1: Statistics on the number of RFQs, offers, minimum offers, and orders per product per day: mean and standard deviation (Stdev) per game, as well as over all games in the training set aggregated.

Game	Total	Invalid	RMSD valid	RMSD invalid
9323tac5	113,221	0.1015	0.2265	0.1770
9324tac5	106,566	0.0429	0.2814	0.1410
9325tac5	120,862	0.0555	0.2731	0.1688
9326tac5	119,708	0.1003	0.2105	0.1732
9327tac5	118,300	0.0473	0.3164	0.1689
7308tac3	143,784	0.0613	0.2451	0.1719
7309tac3	151,945	0.0807	0.2452	0.1876
7310tac3	145,702	0.0554	0.2411	0.1848
7311tac3	163,731	0.0190	0.2878	0.1402
7312tac3	158,975	0.0416	0.2731	0.1660
763tac02	72,053	0.0000	0.2497	0.0000
764tac02	68,530	0.0000	0.1502	0.0000
765tac02	92,456	0.0000	0.2455	0.0000
766tac02	95,261	0.0000	0.1775	0.0000
767tac02	85,242	0.0000	0.3044	0.0000
768tac02	94,292	0.0000	0.2667	0.0000
794tac01	129,444	0.0000	0.2992	0.0000
795tac01	145,368	0.0000	0.3256	0.0000
796tac01	134,994	0.0000	0.2714	0.0000
797tac01	147,451	0.0000	0.2920	0.0000
798tac01	133,736	0.0000	0.3437	0.0000
799tac01	128,980	0.0000	0.3342	0.0000
Aggregated	2,670,601	0.0297	0.2750	0.1730

Table A.2: Statistics on the number of offers and the extent to which these offers are invalid. This table also quantifies the root mean squared deviation (RMSD) from the normalized reservation prices for both valid and invalid normalized offer prices. These statistics are calculated per game, as well as over all games in the training set aggregated.

Game	Total	Invalid	RMSD valid	RMSD invalid
9323tac5	39,708	0.1296	0.2348	0.2174
9324tac5	32,267	0.0394	0.2971	0.1813
9325tac5	38,297	0.0805	0.2802	0.2003
9326tac5	42,180	0.1176	0.2246	0.2042
9327tac5	34,754	0.0722	0.3310	0.2066
7308tac3	41,450	0.1524	0.2587	0.1893
7309tac3	43,102	0.2045	0.2598	0.2124
7310tac3	39,887	0.1261	0.2587	0.2076
7311tac3	37,645	0.0390	0.3169	0.1744
7312tac3	38,412	0.1027	0.2902	0.1964
763tac02	29,295	0.0000	0.2605	0.0000
764tac02	31,705	0.0000	0.1638	0.0000
765tac02	33,629	0.0000	0.2548	0.0000
766tac02	36,201	0.0000	0.1936	0.0000
767tac02	30,199	0.0000	0.3155	0.0000
768tac02	33,342	0.0000	0.2859	0.0000
794tac01	31,760	0.0000	0.3245	0.0000
795tac01	33,113	0.0000	0.3594	0.0000
796tac01	35,009	0.0000	0.2896	0.0000
797tac01	35,508	0.0000	0.3167	0.0000
798tac01	28,726	0.0000	0.3846	0.0000
799tac01	30,034	0.0000	0.3629	0.0000
Aggregated	776,223	0.0548	0.2878	0.2034

Table A.3: Statistics on the number of minimum offers and the extent to which these minimum offers are invalid. This table also quantifies the root mean squared deviation (RMSD) from the normalized reservation prices for valid and invalid normalized minimum offer prices. These statistics are calculated per game, as well as over all games in the training set aggregated.

Experiment	Benchmark	PPARRPAE	Relative deviation
1	22.3808	46.8987	1.0955
2	6.3948	47.8112	6.4765
3	16.9663	45.9162	1.7063
4	31.9214	47.3017	0.4818
5	35.5960	51.8664	0.4571
6	9.8478	54.4973	4.5339
7	-7.9483	46.3227	6.8280
8	26.5347	50.9251	0.9192
9	-1.5760	53.7920	35.1314
10	35.3067	49.4758	0.4013
11	17.8575	47.6696	1.6695
12	25.4310	50.4246	0.9828
13	19.1308	49.4091	1.5827
14	28.0204	48.7326	0.7392
15	15.5909	53.6205	2.4392
16	19.4944	51.1878	1.6258
17	14.6102	45.4497	2.1108
18	10.1712	48.1132	3.7303
19	20.4764	52.3903	1.5586
20	24.2783	50.7384	1.0899
21	29.5051	46.5686	0.5783
22	28.6080	54.6886	0.9117
23	31.4340	44.3995	0.4125
24	21.9394	48.7604	1.2225
25	-9.9027	46.1636	5.6617
26	34.9400	51.7080	0.4799
27	23.5693	49.5133	1.1008
28	-3.5125	50.2345	15.3017
29	34.7362	54.6371	0.5729
30	26.6605	50.0827	0.8785
31	30.1376	48.3689	0.6049
32	19.2111	51.1211	1.6610
33	9.5348	47.0312	3.9326
34	25.6683	49.9390	0.9456
35	28.2874	45.5603	0.6106
36	33.1365	46.7110	0.4097
37	23.9364	49.4190	1.0646
38	15.3969	52.3362	2.3991
39	-4.2309	48.4070	12.4414
40	0.9167	47.5390	50.8611

Table A.4: Final bank account balances per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. Balance values are expressed in millions.

Experiment	Benchmark	PPARRPAE	Relative deviation
1	3.0600	4.2480	0.3882
2	2.1990	4.3860	0.9945
3	3.0630	4.3970	0.4355
4	3.6990	4.4760	0.2101
5	3.7790	4.6700	0.2358
6	2.6140	5.0600	0.9357
7	1.4480	4.2710	1.9496
8	3.7750	4.7400	0.2556
9	1.6090	7.1290	3.4307
10	3.6000	4.5270	0.2575
11	3.0190	4.3580	0.4435
12	4.1090	4.6250	0.1256
13	3.1870	4.5850	0.4387
14	3.2290	4.5090	0.3964
15	3.0190	4.9020	0.6237
16	3.6600	4.8220	0.3175
17	2.9020	4.4250	0.5248
18	2.8420	4.3360	0.5257
19	4.7180	4.7060	-0.0025
20	3.2250	4.8620	0.5076
21	3.4180	4.3970	0.2864
22	3.5290	4.9420	0.4004
23	3.9860	4.2300	0.0612
24	3.4870	4.4930	0.2885
25	1.3340	4.2930	2.2181
26	3.9790	4.7670	0.1980
27	3.6860	4.7240	0.2816
28	0.3130	4.6360	13.8115
29	4.9730	4.8470	-0.0253
30	4.4150	4.6800	0.0600
31	3.5290	4.5160	0.2797
32	2.8470	4.7130	0.6554
33	2.8720	4.5760	0.5933
34	3.7670	4.7090	0.2501
35	3.1980	4.1490	0.2974
36	3.5890	4.4630	0.2435
37	3.0550	4.7610	0.5584
38	3.0220	4.7700	0.5784
39	0.5030	4.7280	8.3996
40	1.2030	4.4660	2.7124

Table A.5: Number of orders obtained per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value. Values for the number of obtained orders are expressed in thousands.

Experiment	Benchmark	PPARRPAE	Relative deviation
1	0.4494	0.4536	0.0093
2	0.4233	0.4775	0.1278
3	0.4699	0.4758	0.0125
4	0.5091	0.4584	-0.0996
5	0.4821	0.4677	-0.0299
6	0.4502	0.4539	0.0083
7	0.4035	0.4609	0.1423
8	0.4866	0.4725	-0.0291
9	0.3954	0.4321	0.0927
10	0.4928	0.4487	-0.0895
11	0.4654	0.4686	0.0070
12	0.4837	0.4898	0.0127
13	0.4364	0.4677	0.0719
14	0.4461	0.4516	0.0122
15	0.4413	0.4731	0.0721
16	0.4890	0.4612	-0.0568
17	0.4809	0.4600	-0.0434
18	0.4308	0.4846	0.1248
19	0.5032	0.4927	-0.0207
20	0.4740	0.4489	-0.0531
21	0.4526	0.4654	0.0284
22	0.4529	0.4717	0.0416
23	0.4732	0.4486	-0.0520
24	0.5216	0.4815	-0.0767
25	0.4222	0.4731	0.1205
26	0.4598	0.4574	-0.0053
27	0.4927	0.4755	-0.0350
28	0.4524	0.4659	0.0298
29	0.4818	0.4834	0.0033
30	0.4815	0.4711	-0.0216
31	0.4793	0.4662	-0.0274
32	0.4310	0.4562	0.0585
33	0.5622	0.4637	-0.1752
34	0.4931	0.4651	-0.0568
35	0.4981	0.4619	-0.0726
36	0.4445	0.4442	-0.0007
37	0.4700	0.4646	-0.0114
38	0.4700	0.4652	-0.0102
39	0.4182	0.4739	0.1332
40	0.5033	0.4744	-0.0576

Table A.6: Root mean squared deviation of estimated probability of acceptance per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value.

Experiment	Benchmark	PPARRPAE	Relative deviation
1	1,091	1,463	0.3410
2	693	1,613	1.3276
3	807	1,679	1.0805
4	1,375	1,705	0.2400
5	1,398	1,700	0.2160
6	681	1,724	1.5316
7	506	1,491	1.9466
8	1,143	1,661	0.4532
9	523	789	0.5086
10	1,579	1,700	0.0766
11	931	1,531	0.6445
12	1,085	1,717	0.5825
13	857	1,695	0.9778
14	1,081	1,529	0.4144
15	816	1,811	1.2194
16	851	1,751	1.0576
17	924	1,696	0.8355
18	739	1,547	1.0934
19	845	1,702	1.0142
20	1,072	1,800	0.6791
21	1,258	1,582	0.2576
22	1,232	1,834	0.4886
23	1,238	1,616	0.3053
24	1,082	1,626	0.5028
25	494	1,552	2.1417
26	1,232	1,656	0.3442
27	1,154	1,702	0.4749
28	113	1,675	13.8230
29	835	1,754	1.1006
30	700	1,719	1.4557
31	1,249	1,654	0.3243
32	1,075	1,686	0.5684
33	490	1,774	2.6204
34	996	1,768	0.7751
35	1,303	1,542	0.1834
36	1,387	1,595	0.1500
37	1,093	1,740	0.5919
38	840	1,709	1.0345
39	10	1,706	169.6000
40	37	1,663	43.9459

Table A.7: Number of market polls per experiment for the two considered MinneTAC variants, along with the relative deviation of the PPARRPAE value from the benchmark value.