Charles University in Prague

Faculty of Social Sciences
Institute of Economic Studies

MASTER THESIS

Comparison of double auction bidding strategies for automated trading agents

Author: **Bc. Daniel Vach**

Supervisor: **Ing. Aleš Maršál M.A.**

Academic Year: **2014/2015**

Declaration of Authorship

The author hereby declares that he compiled this thesis independently, using only the listed resources and literature. This thesis was not used to obtain another academic degree.

The author grants to Charles University permission to reproduce and to distribute copies of this thesis document in whole or in part.

Prague, July 29 , 2015

Signature

Acknowledgments

I am grateful to my supervisor, Ing. Aleš Maršál M.A., for his valuable advice and patience.

Bibliographic record

Vach, D., 2015. *Comparison of double auction bidding strategies for automated trading agents*. Master thesis. Charles University in Prague.

Character count: 119 487

Abstract

In this work, ZIP, GDX, and AA automated bidding strategies are compared in symmetric agent-agent experiments with a variable composition of agent population. ZIPOJA, a novel strategy based on ZIP with Oja's rule extension for updating its optimal price, is introduced. Then it is showed that ZIPOJA underperforms in competition against other strategies and that it underperforms even against the original ZIP. Dominance of AA over GDX and ZIP is questioned and it is showed that it is not robust to composition changes of agent population and that the experimental setup strongly affects the results. GDX is a dominant strategy over AA in many experiments in this work in contrast to the previous literature. Some mixed strategy Nash equilibria are found and their basins of attraction are shown by dynamic analysis.

Keywords	Algorithmic trading, Bidding strategy, ZIP, Zero-Intelligence Plus, GDX, Adaptive Aggressiveness, AA, Agent trading, Autonomous trading, Oja's rule
JEL Classification	C45, C61, C73
Author's e-mail	`daniel.vach@gmail.com`
Supervisor's e-mail	`ales.marsal@seznam.cz`

Abstrakt

V této práci jsou porovnávány automatické obchodní strategie ZIP, GDX a AA v symetrických agent-proti-agentovi experimentech, kde se mění zastoupení jednotlivých strategií v populaci agentů. Také je představena nově vytvořená strategie ZIPOJA, která je založena na ZIP a obohacena Ojovým učícím pravidlem pro aktualizaci optimální ceny. ZIPOJA strategie je porovnávána proti ostatním strategiím, z čehož vychází, že se jí nedaří v porovnání s ostatními strategiemi. Dokonce původní ZIP ji také poráží. Dále je zjištěno, že dominance AA nad GDX a ZIP není robustní ve změnách složení populace agentů. Výsledek lze také silně ovlivnit vlastnostmi experimentu. GDX dominuje AA v mnoha experimentech, které jsou v této práci provedeny, což je v kontrastu s výsledky v předchozí literatuře. Nalezeny jsou také Nashovy rovnováhy ve smíšených strategiích. Dynamická analýza je použita pro nalezení spádových oblastí jednotlivých rovnováh.

Klíčová slova	Algoritmické obchodování, Strategie nabízení, ZIP, Zero-Intelligence Plus, GDX, Adaptivní Agresivita, AA, Obchodování agentů, Autonomní obchodování, Ojovo pravidlo
JEL Klasifikace	C45, C61, C73

E-mail autora	`daniel.vach@gmail.com`
E-mail vedoucího práce	`ales.marsal@seznam.cz`

Master Thesis Proposal

Author	Bc. Daniel Vach
Supervisor	Ing. Aleš Maršál M.A.
Proposed topic	Comparison of double auction bidding strategies for automated trading agents

Motivation

Algorithmic trading is of great importance in markets today. It is a phenomenon which deserves a lot of attention. Trading algorithms for continuous double-auction are studied and developed vastly in literature.

There have been three main strategies examined thoroughly in literature: namely ZIP (zero intelligence plus), the GD (Gjerstad-Dickhaut) class and AA (adaptive aggressiveness). ZIP and AA make use of Widrow-Hoff adaptation and GD is based on belief functions. De Luca and Cliff (2011) compares these three strategies against each other and against humans in artificial market. They come to a conclusion that algorithmic strategies outperform human traders and that AA performs better than the rest of algorithmic strategies.

I am going to challenge the algo-to-algo results of De Luca and Cliff (2011) by repeating their experiment, but with different parameters of the artificial world. Moreover, I will try to test these three families of algos on historical data obtained from the real market.

Last but not least, I will try to develop my own algorithmic strategy. I will test it on ex post real market data, in the artificial world and I will show how these algorithmic strategies perform on generated stochastic process simulating the ex ante real market data.

Hypotheses

1. AA trade strategy does not outperform the other trading strategies in different artificial world setting as significantly as in De Luca and Cliff (2011).
2. There is a difference how trade strategies perform based on number of rival strategies included in experiment.
3. There is a difference how trade strategies perform based on number of agents following each rule.

4. It is possible to rank these algorithmic strategies with sufficient significance and this order is robust to changes in artificial world setting.

Methodology

I am going to test ZIP60, GDX, AA and newly developed trading strategy against each other in competitive environment similar to one used by De Luca and Cliff (2011). I will use OpEx (Open Exchange Experimental Laborator). Compared to them, I will change reasonably the artificial world setting in number of participating strategies and in number of agents of each strategy in one experiment. Based on this I will produce sensitive analysis of outcomes on different number of strategies and agents included. Then I will use real order book data from one day as an input for decision making of strategies and I will compare their successfulness in this experiment. Then I will compare these results with artificial world results. I will also try to model artificial orderbooks with the same generic properties as my real order book simulated by Hawkes process similarly to Hewlett (2006). I will apply all four strategies on these orderbooks and see whether there are changes to the pattern of success. This step should make the results more robust to the randomness.

Expected Contribution

This thesis should discover whether the results of De Luca and Cliff (2011) are robust to some changes in artificial world setting. I will also show whether these results are valid for example of historical data as well. I will challenge the rank order of trading algos introduced in De Luca and Cliff (2011). I will possibly show that it is not a consistent result or at least I will provide some more evidence to this topic. I will introduce completely new algorithmic trading strategy and show whether it performs worse, comparable or better than some known algos.

Outline

1. Introduction
2. Literature Review
3. Definition of Trade Agents
4. Definition of Artificial World
5. Results in artificial world
6. Results with historical prices
7. Results with market simulating stochastic function
8. Comparison and Discussion
9. Conclusion

Probable references

1) Wellman, Michael P. (2011) *"Trading Agents", Synthesis Lectures on Artificial Intelligence and Machine Learning*, Volume.5, Issue.3, pp.1, ISSN: 19394608

2) D. Cliff (1998) *"Genetic optimization of adaptive trading agents for double-auction markets"*, Proc. Comput. Intell. Financial Engineering (CIFEr), pp.252 -258

3) D. Cliff (2006) *ZIP60: Further explorations in the evolutionary design of trader agents and online auction market mechanisms.*

4) M. De Luca and D. Cliff (2011) *Human-agent auction interactions: adaptive-aggressive agents dominate.* In Proc. ICAART, ISBN: 978-1-57735-513-7

5) D. Cliff and J. Bruten (1997) *Minimal intelligence agents for bargaining behaviors in market-based environments.* Technical report, HPL-97-91, HP Labs

6) Vytelingum, P. (2006) *The structure and behaviour of the Continuous Double Auction.* PhD Thesis. University of Southampton.

7) Anonymous: *OpEx: An Open Exchange Experimental Economics Laboratory in a Box.* Technical Report in preparation.

8) Preist, C., Van Tol, M. 1998. *Adaptive agents in a persistent shout double auction.* In Proceedings of the First International Conference on Information and Computation Economies. ACM Press,11-18.

9) Tesauro, G., Das, R. 2001. *High-performance bidding agents for the continuous double auction.* In Proceedings of the Third ACM Conference on Electronic Commerce, 206-209.

10) Foucault, T., O. Kadan, E. Kandel. 2005. *Limit order book as a market for liquidity.* Review of Financial Studies, 18(4), pp. 1171-1217.

11) Hewlett, P. 2006. *Clustering of order arrivals, price impact and trade path optimization.* Workshop on Financial Modeling with Jump processes, Ecole Polytechnique, 6-8 September 2006.

Contents

List of Tables

List of Figures

Acronyms

AA Adaptive Aggressiveness

CDA Continuous Double Auction

GD Gjerstad-Dickhaut

GDX modified Gjerstad-Dickhaut

HFT High-Frequency Trading

IJCAI International Joint Conference on Artificial Intelligence

OEA Optimal Execution Algorithms

OpEx Open Exchange Experimental Laborator

ZIP Zero-Intelligence Plus

ZIPOJA Zero-Inteligence-Plus with Oja's learning rule

Chapter 1

Introduction

Algorithmic trading is a great buzz word on the financial market nowadays. The rise of computer-based algorithmic trading is one of the most significant developments in modern financial markets. Some estimates have algorithmic trading accounting for 30% of UK equity volume and up to 60% of US equity volume. It is a phenomenon which deserves a lot of attention by all participants of the market.

Algorithmic trading is quite a wide concept. It includes all the types of traders which are very different from each other in the basic idea of how they work. There exist High-Frequency Trading (HFT) participants which use extraordinarily high speed and sophisticated programs for generating, routing and executing orders. HFT works in a very short time-frame for establishing and liquidating positions. HFT is opposed by Optimal Execution Algorithms (OEA) employed for example by large asset management firms. OEA work in a way which divides one big chunk trade into a lot of small executed orders during one day in order not to change the price into unfavourable direction. Buying it in one big chunk order would lead into a large price change, because this order would only match with higher limit orders on the opposite site of the order book. The price would shift back close to its original value very shortly after the big chunk order, implying large losses for the executor of this order. These losses can be called transactional costs. In order to minimize these transactional costs, OEA were developed.

OEA work on the basis of different price estimation ideas which recognize when to buy and how much to buy during one day in order to buy a specific amount of stocks in total and to distort the price as little as possible. For this, robots employing automated bidding strategies are used.

The topic of automated bidding strategies is covered by vast literature (Gode and Sunder (1993), Cliff (1997), Preist and van Tol (1998), Cliff et al. (1997), Gjerstad and Dickhaut (1998), Das et al. (2001), Tesauro and Das (2001), Walsh et al. (2002), Tesauro and Bredin (2002), Vytelingum et al. (2008), De Luca and Cliff (2011b) and many more). In this work, I will study 3 of these bidding strategies, namely ZIP, GDX and AA. Moreover, I will try to develop one new bidding strategy myself. These bidding strategies will be tested against each other in the artificial environment of Open Exchange Experimental Laborator (OpEx) v1.2 , convenient simulator of double auctions and one of a few publicly available sources of AA code. I will show which strategy dominates which and how they perform in various experimental setups with a changing share of each strategy in the total population. This should provide more robust results on comparison of bidding strategies and provide a new point of view on comparison of ZIP, GDX and AA since these strategies are now considered to be the most important. ZIP and GDX are benchmarks for testing performance in the world of bidding strategies and AA (as claimed by previous literature) is currently a dominant strategy which outperforms the other two strategies.

Therefore, I will try to test these hypotheses:

1. AA bidding strategy does not outperform the other bidding strategies in different experimental setups as significantly as in De Luca and Cliff (2011b).
2. There is a difference in how bidding strategies perform based on the number of rival strategies included in the experiment.
3. There is a difference in how trade strategies perform based on the number of agents following each rule.
4. It is possible to rank these algorithmic strategies with sufficient significance and this order is robust to changes in the experimental setup.

In the next section, I will recapitulate the research made on the topic of automated bidding strategies in the past, I will provide a general overview on continuous double auctions and I will have a few general remarks on OpEx. Chapter 3 will include definitions of ZIP, GDX and AA bidding strategies. Chapter 4 will be dedicated to the specifics of a newly developed bidding strategy ZIPOJA. In chapter 5, I will elaborate on the methodology of comparison and explain what will be done. Chapter 6 will present the experimental setup and results obtained from experiments. Chapter 7 concludes.

Chapter 2

Background

In this chapter, I will provide some information on the past research on the bidding strategies and agents. There exist some other bidding strategies apart from these mentioned here. I will focus on ZIP, GDX, and AA history. Then I will introduce the continuous double auction topic and describe OpEx trading simulator. The rest of related literature will be mentioned throughout the work when relevant.

2.1 Bidding strategies literature

Multi-agent systems and economics are combined in the research to the benefit of both disciplines. This has led to improvements in several directions. One of the most significant improvements is the creation of agents to trade on behalf of humans in electronic trading environment as highlighted by Preist and van Tol (1998).

Gode and Sunder (1993) have developed Zero-Intelligence agents and showed that the market discipline can be to the large extent responsible for efficiency and convergence to equilibrium in double auction markets rather than intelligence and logic of traders.

Cliff et al. (1997) showed that market discipline is important for efficiency, but Zero-Intelligence is not enough in all cases of supply and demand curves for convergence towards equilibrium.

Cliff (1997) has improved Zero-Intelligence agents into ZIP (more on ZIP in its own section 3.1) and shown that adaptive agents with simple logic are able to participate in a trade of double auction marketplace. Moreover, they are able to trade in such a way that trade prices converge towards the competitive equilibrium price in the market.

Gjerstad and Dickhaut (1998) developed the new algorithm which is called GD nowadays (more on GD and its updates can be found in section 3.2 dedicated to GDX). The GD is centered around belief functions formed by every GD agent using observed market data which assigns probabilities of trade realization and profit to prices and chooses the best combination.

There were several updates to GD. Das et al. (2001) modified the original GD algorithm to fit it in MAGENTA (the real-time asynchronous framework). The difference between GD and this modified version was in the treatment with persistent orders not dealt with in the original GD.

Tesauro and Das (2001) modified the GD in a different way. They called their modification MGD. It solves the problem of prices above the highest price and the lowest price in the recorded history.

There was one more modification of GD algorithm called GDX which will take part in my experiments. Tesauro and Bredin (2002) developed GDX modification to use dynamic programming to price orders. It means that the GDX trader takes into account the effect of trading the current unit immediately, and the effect of trading in the future. The belief function is modified to account for this by discounting the future profit.

The third strategy to be used in this work is Adaptive Aggressiveness developed by Vytelingum (2006) and presented by Vytelingum et al. (2008) as well. The AA strategy is predictive and history-based bidding strategy that software agents can use to bid in Continuous Double Auction (CDA) (More on AA in section 3.4). It is based on short-term and long-term learning behaviour as well as estimation of the equilibrium. In the short-term, it optimizes the aggressiveness in order to react appropriately to each transaction. In the long term, it tries to react to the change in the general equilibrium.

With developing algorithmic strategies, one can wonder how good bidding strategies are against each other and against human traders as well. There is numerous literature on comparison of bidding strategies and basically every time a new one is developed, it is benchmarked against some other so previously mentioned literature is a good source for finding more information on comparison of bidding strategies. Nowadays, mostly ZIP and GDX are used as a benchmark for comparison.

Yet, there are other interesting papers on this topic as well. De Luca and Cliff (2011b) tested bidding strategies, namely ZIP, GDX and AA and proved that they significantly outperform human traders in the CDA market. Their work was inspired by the seminal work reported by IBM at International Joint Conference on Artificial Intelligence (IJCAI) 2001 in which it was demonstrated that two of their tested software-agent traders consistently and robustly outperform human traders in their artificial environment, namely ZIP and GD strategy. Results by IBM were really game-changing because they signaled the start of the era of superiority of the computerized machine traders over the human traders.

Toft (2007) replicates some of trading-agent experiments from previous studies and reports significant difficulty in replicating published results. Therefore, one has to be careful and challenge the results of previous literature. This is done in this work as a result of my experiments.

Another interesting paper written by Walsh et al. (2002) shows the methodology for comparison of bidding strategies based on dynamic analysis. They are able to find mixed strategy Nash equilibria in the three-strategy space with symmetric properties (the same number of buyers as sellers). Vytelingum et al. (2008) present results based on the same approach for two-strategy space where he allows strategies to be asymmetrical. To a large extent, I will follow the approach of Walsh et al. (2002) for symmetric three-strategy games.

2.2 Continuous double auction

The vast majority of financial products are traded electronically. These trades follow exact rules and together create what is known as a common virtual marketplace. These exchanges maintain the system through which traders can execute their trades. The set of rules of trade defining interaction between traders (buyer and seller) on a market forms its market mechanism. Continuous double auction (CDA) is the most prevailing market mechanism used in the real world due to its high efficiency. Smith (1962) demonstrates in his Nobel-prize-winning work that markets governed by the CDA with human traders converge in transaction prices with close to optimal efficiency to the point where the supply curve and the demand curve intersect, to its equilibrium price. He also showed that this result is robust to sudden changes in supply and demand curves which subsequently lead transaction prices to converge to a new equilibrium price.

CDA mechanism is a prevalent way how the electronic trading is done in the financial markets nowadays. CDA can be described as a set of rules in which traders may asynchronously post bids and offers electronically and this electronic system provides information on current outstanding bids and offers to all traders. The information to all traders can be limited only to showing the best (the highest) bid and the best (the lowest) offer or it can show more depth of bids and offers outstanding - called order book.

As De Luca and Cliff (2011b) explain, continuous double auction is specified by the ability of both buyers and sellers to make bids and accept offers asynchronously at any time during the trading period (usually referred to as a trading day). In CDA, all the offers are publicly visible by all market participants in limit order books. The trade is executed whenever there is an existing outstanding bid with price greater or equal to the outstanding ask.

CDA is deeply linked to limit order book. Each of three strategies used here use it to some extent. According to Lorenz and Osterrieder (2009), this makes sense. They say that it is better to work with the order book rather than only with the stock price. Trading strategies based on information from the order book have much larger chance to be superior to those that only use transaction information, in other words stock prices. This makes common sense since you can deduce the whole stock price development from the limit order book and there is some additional information in every order submitted. In other words, stock price is an aggregation of orders and therefore there has to be additional information when you can have more detail on what movements determine the change in the price. For example, there might be huge cancelling of limit buy orders signaling that the price can more easily go down than up.

As you can see, CDA is created by relatively simple set of rules. Despite trading rules being this simple, CDA and its nonlinearities grow really complex when it comes to its analysis. Due to this obstacle, traditional mathematical methods such as game theory are hard to use and as a result, researchers tend to use empirical approaches.

CDA is the main focus of the financial world today. However, a different discrete double auction method will be used in this work. I explain why it is a reasonable simplification, possibly without loss of generality, in subsection 6.1.1 Discrete Event Simulator.

2.3 Open Exchange Experimental Laborator v1.2

My work will be devoted to comparison of trading strategies in the artificial environment where algorithmic agents will interact. This artificial environment will be provided by Open Exchange Experimental Laborator (OpEx) which is experimental artificial trading platform developed by Marco De Luca at the University of Bristol. OpEx is used to compare bidding strategies (and human traders) by De Luca and Cliff (2011b) and has been experimental economics platform in several more studies such as De Luca and Cliff (2011a), De Luca et al. (2011), Cartlidge and Cliff (2012) and Cartlidge et al. (2012).

OpEx was created to be closely similar in structure and the behaviour to commercial financial market electronic trading systems and at the same time to be generic enough to support experimental economics simulations of arbitrary complexity. OpEx can mediate access of both human traders and agent traders into Order Processor (e.g. simple Exchange place) where the trade is executed.

The process in OpEx is as follows, once orders are generated, they are sent to Order Manager, which routes them to the appropriate Order Processor (Exchange). Orders are processed in Exchange according to the specific order matching logic implemented called price-time priority. This matching logic constitutes the foundation of the CDA. Order completion data are then passed back to Order Manager, which in turn inform the appropriate trader. These order data are private and only trade concluding trader is informed about order completion relative to the specific order. Separately, market data are published through Market Data Bus which can be seen by all market participants.

Agents will be included into experiments as individual plug-ins of OpEx running on an instance of the agent host module. OpEx allows running multiple instances of Agent Host and therefore researchers are able to plug different trading agent types into different Agent Host modules and to test them against each other.

Chapter 3

Bidding strategies previously used (ZIP, GDX, AA definition)

I should extensively introduce three automated bidding strategies in order to see the exact differences between these three strategies and the newly developed one described in the later chapter.

3.1 ZIP - Zero-Intelligence Plus

First algorithmic strategy to be explained is Zero-Intelligence Plus (ZIP) algorithm invented by Cliff (1997) by improving Zero-Intelligence agents developed by Gode and Sunder (1993). Zero-Intelligence trading strategy and ZIP both do stochastic bids but there is a difference that ZIP traders employ a very simple form of machine learning which will be described later. Cliff (1997) used ZIP to investigate what are the minimum requirements on intelligence of an algorithm to set the convergence path to achieve market equilibrium price. ZIP became widely used as a benchmark for comparison of performance of other trading algorithms as for example by Tesauro and Das (2001) and Das et al. (2001).

As De Luca and Cliff (2011b) explain, ZIP trading algorithm is based on real-valued profit margin maintained by each agent following this strategy. The ZIP agent employs simple heuristic mechanisms to adjust its margin based on market data. Profit margin is defined for ZIP trader agent as the difference between the agent's limit price and the shout price. Limit price is the maximum amount of money, buyer can spend to buy the unit or the minimum price, for which seller can sell the unit. Shout price is quoted price at which agent is willing to trade the underlying asset.

Cliff (1997) describes ZIP trader as an algorithm altering its profit margin on the basis of four factors. The first of four factors ZIP trader uses is whether the trader is active in the market or not. This is determined by the ability of selling/buying of additional units. If the ZIP trader has already bought/sold the expected amount of units, he turns inactive for the rest of the trading period (trading day). The ZIP trader is designed to eliminate the need for sophisticated memory mechanisms and thanks to simple machine learning it is possible. The rest of four factors are devoted to cope with the machine learning so they watch the most recent shout-price q. One factor contains whether it was a bid or an offer. Second factor contains whether the last offer or bid with shout price q was accepted and resulted in transaction or not. Third factor controls for the price of the last shout q. Based on these factors, ZIP trader decides about its shout price in the next period by a specific set of equations described below.

3.1.1 Bargaining mechanism

Each ZIP trader determines its next shout price p based on its limit price λ times 1 plus profit margin. The whole ZIP algorithm is based on setting the right profit margin based on current conditions (shout price in the market).

These equations can be summed up by the following ZIP trader's bargaining mechanism:

- For ZIP sellers the mechanism is as following: If the last shout was accepted at price q, then any seller si for which $p <= q$ should raise its profit margin. Inaddition, if the last shout was a bid then any active seller si for which $p >= q$ should lower its margin. If the last shout was not accepted at price q, then any active seller s for which $p >= q$ should lower its margin.
- For ZIP buyers the mechanism can be described in the following way: If the last shout was accepted at price q, then any buyer b for which $p <= q$ should raise its profit margin. In addition, if the last shout was an offer then any active buyer b for which $p >= q$ should lower its margin. If the

last shout was not accepted at price q, then any active buyer b for which $p >= q$ should lower its margin.

This qualitative bargaining mechanism has to be specified also quantitatively in order to test it in the real markets or at least in the simulations of the real markets. It is necessary to specify how the profit margin reacts for buyers and sellers.

3.1.2 Quantitative rules for ZIP

Cliff (1997) explains the whole set of quantitative rules in the detail. I will recapitulate the rules for ZIP traders here. At a given time t, each ZIP trader, calculates the shout-price $p(t)$ based on its particular limit price λ and trader's real valued profit-margin $\mu(t)$ according to the following equation:

$$p(t) = \lambda(1 + \mu(t)) \tag{3.1}$$

This equation means that a trader's margin is completely dependent on variable μ. When μ increases, margin and profit for the seller increases and when the μ decreases, margin and consequently profit for the seller decreases. The situation is completely opposite for buyers. They raise their margin by decreasing μ and lower their margin by increasing μ. μ is constrained into interval $\mu(t) \in [0, \infty)$ for sellers for all t and $\mu(t) \in [-1, 0]$ for buyers for all t. Shout-price is defined this way to allow each trader to change its shout-price through μ to alter dynamically in response to the actions of other traders in the market (accepted and rejected bids and offers) to remain a competitive match between trader's shout and the rest of the market.

In order for price to be adaptive there has to be some kind of update rule in operation. Cliff (1997) implemented simple machine learning rule called Widrow-Hoff "delta rule" (Widrow and Hoff, 1960) which is present for example in back propagation adaptation algorithm in neural networks (Rumelhart et al., 1986). The Widrow Hoff "delta rule" is specified by this equation:

$$A(t + 1) = A(t) + \Delta(t) \tag{3.2}$$

Where $A(t)$ is the actual output at time t and $A(t + 1)$ is the actual output on the next time step. $\Delta(t)$ denotes the change in the output. This change is determined by the learning rate coefficient β times the difference between $A(t)$ and the desired output at time t, denoted by $D(t)$:

$$\Delta(t) = \beta(D(t) - A(t)) \tag{3.3}$$

This in practice mean that Widrow-Hoff rule leads the series to asymptotic convergence of $A(t)$ to $D(t)$ at speed of β. Cliff (1997) states that, ZIP traders use this adaptation method to set the shout-price for the next time step ($p(t+1)$) closer to target price $\tau(t)$. In order to incorporate this Widrow-Hoff rule into ZIP trader decision making equation the price setting equation must be rearranged:

$$\mu(t+1) = (p(t) + \Delta(t))/\lambda - 1 \tag{3.4}$$

Where $\Delta(t)$ is the Widrow-Hoff delta value calculated by the following equation with specific β speed for each ZIP trader times the difference between the target price and the shout-price for time t:

$$\Delta(t) = \beta(\tau(t) - p(t)) \tag{3.5}$$

The last not specified variable is target price $\tau(t)$. This one is tricky because simply setting it equal to shout-price would mean that for traders with the trader's current shout-price close to last shout price, there would be a really small or zero change in their shout-prices. This could lead to far from the true competitive equilibrium because any trader does not challenge the current prices. Naturally, buyers want to buy for less and sellers want to buy for more. To overcome this obstacle in setting target price, random variable and absolute variable which will affect the target price have to be introduced. They will affect the target price in a way that will make impossible the scenario of equilibrium far from the true competitive one.

$$\tau(t) = R(t)q(t) + L(t) \tag{3.6}$$

Where R is a "relative" random coefficient that sets the price as a real multiple of the last shout-price $q(t)$. L is small "absolute" random coefficient that alternates the last shout-price. Setting the target price in this way will improve the ZIP trader's behaviour to better reflect human trader in trying to obtain additional profit. When there is a motivation to increase the dealer's shout price, $R \in (1, \infty)$ and $L \in (0, \infty)$, and when there is an intention to decrease the dealer's shout price, $R \in (0, 1)$ and $L \in (-\infty, 0)$. Every time period, the new target price is calculated using newly generated values of relative random coefficient and absolute random coefficient. Both random variables are independent and identically distributed for all traders. Relative variable ensures to change the large shout-prices by greater amounts than small last shout-prices. The use of small absolute variable which alternates the target price ensures that target prices differs by few pips even for shout-prices close to zero. As stated by Cliff (1997), this can be considered as a random noise in the calculation of the target price or the error in determining the right value of target price which can reflect human trader behaviour.

There is one last phenomenon to tackle with to obtain comprehensive set of rules for algorithmic strategy. As Cliff (1997) explains, in case of dynamically variable desired output $D(t)$, Widrow-Hoff rule tends to oscillate around the desired output with high frequency. To prevent this oscillation to happen, the learning system can be damped by the interconnection to the past values of change. We can consider the example of trader whose observations of shouts and transactions lead him to increase his profit margin in this period of time. The next recorded transaction can indicate that the profit margin should be lowered now. However this decrease might be a little bit premature and contradicting the prevalent trend of the increasing shout-price. In this case, there might be good to incorporate some interconnection term which will affect the rate of increase of the margin, rather than the margin itself. It means that when the trajectory of profit margin is increasing, set of observations with lower shout price than the shout price of the trading agent will decrease the rate of increase of the profit margin first and when the new trend of decreasing shout price prevails, it can consequently lead to decrease the change of profit margin below zero or in other words it can lead to decrease in profit margins. Basically, this interconnection defends the trader against the events which oppose the trend but do not prevail, while it allows changes in trader's profit margin based on observed shout-prices and transactions to stay competitive in the ever changing environment.

To incorporate this Cliff (1997) introduces a momentum coefficient for each trader which allows ZIP traders to have a momentum indicating in which way the profit should be altered. The momentum coefficient is denoted by $\gamma \in [0, 1]$

and for $\gamma = 0$ the trader takes no account of past changes in determining the next change to the value of the profit margin μ. With non-zero values of γ, there exists the relevance of past changes in determining the change in the next period. In incorporating of such momentum mechanism, Cliff(1997) was inspired by work of Rumelhart et al. (1986) on back-propagation neural network learning. The following equation shows how the momentum is updated with $\Gamma(0) = 0$:

$$\Gamma(t+1) = \gamma\Gamma(t) + (1-\gamma)\Delta(t) \tag{3.7}$$

When Γ is imputed instead of Δ into equation determining the profit margin of the next period, update rule used by ZIP traders is obtained:

$$\mu(t+1) = (p(t) + \Gamma(t))/\lambda - 1 \tag{3.8}$$

However, there seems to be incompatibility of these equations and explanation behind them. If $\Gamma(t)$ is supposed to replace $\Delta(t)$ in the future profit margin equation =¿ substitute equation 3.7 into equation 3.8, then it would result into the following equation which is dependent on Δ from previous period and not a current one:

$$\mu(t+1) = [p(t) + (\gamma\Gamma(t-1) + (1-\gamma)\Delta(t-1))]/\lambda - 1 \tag{3.9}$$

(3.10)

This would mean that future margin is updated by last period of Δ not taking into account the newest information available (Δ from current period), effectively reacting with one unnecessary period lag. Therefore I believe, the following equation describes the relation between Γ and Δ more accurately.

$$\Gamma(t+1) = \gamma\Gamma(t) + (1-\gamma)\Delta(t+1) \tag{3.11}$$

or for purpose of new strategy described in chapter four the same equation rewritten.

$$\Gamma(t) = \gamma\Gamma(t-1) + (1-\gamma)\Delta(t) \tag{3.12}$$

(3.13)

After this clarification, the whole quantitative definition of ZIP by Cliff (1997) can be therefore summarized by set of equation in the next subsection.

3.1.3 Recapitulation of quantitative rules

The whole quantitative definition of ZIP by Cliff (1997) with small change in notation as explained above:

$$\mu(t+1) = (p(t) + \Gamma(t))/\lambda - 1 \tag{3.14}$$

$$\Gamma(t) = \gamma\Gamma(t-1) + (1-\gamma)\Delta(t) \tag{3.15}$$

$$\Delta(t) = \beta(\tau(t) - p(t)) \tag{3.16}$$

$$\tau(t) = R(t)q(t) + L(t) \tag{3.17}$$

where μ is profit margin, p is price, Γ is momentum of price, λ is limit price, Δ is the distance between target price and current price, τ is target price, q is last shout-price, γ and β are parameters, and and Ł are random variables. For the exact meaning see the previous subsection 3.1.2.

3.1.4 Setting of parameters for ZIP

Setting parameters L, R, β and γ and taking in the account whether the ZIP trader is active/inactive, price q of the last shout in the market, whether the last shout was bid or offer and whether it was accepted, Cliff (1997) is able to construct the ZIP algorithmic trader which is able to participate on trade. Cliff (1997) assigns value of L uniformly from [0.00, 0.05] for increases in price and [-0.05, 0.00] for decreases in price. He assigns R uniformly distributed random value from [1.00, 1.05] for price increases and from [0.95, 1.00] for price decreases. β is set in the beginning of the trade period for all ZIP traders and remains constant for the rest of an experiment. It comes from uniform distribution of range [0.1, 0.5]. γ is generated in the similar way with values from [0.2, 0.8]. Profit margin is set between 5% and 35% for all ZIP traders. I will follow these instructions similarly to De Luca and Cliff (2011b) to keep my study comparable.

3.2 GD/GDX - Gjerstad and Dickhaut

Gjerstad and Dickhaut (1998) developed the new algorithm which is called GD nowadays. The GD trading algorithm is based on the different logic than ZIP algorithm. GD is centered around belief functions formed by every GD agent using observed market data.

3.2.1 Definition of GD

GD agents collect the accepted shouts and rejected shouts which have occurred during the last M trades (recommended value is 4-5). GD trader stores them in a history H, from which a GD agent forms a belief function assigning the probability that an order will be executed at price p. De Luca and Cliff (2011b) show how the belief function of GD looks like:

$$f(p) = \frac{TBL(p) + AL(p)}{TBL(p) + AL(p) + RBG(p)} \tag{3.18}$$

Where $TBL(p)$ represents the number of accepted bids found in recorded history H at price lower than p. $AL(p)$ is the number of asks in recorded history H with price lower than p. $RBG(p)$ is the number of rejected bids in recorded history H at price higher or equal to p. For the seller it is the exact opposite. The belief function decreases with p for buyer whereas buyer belief increases with p. As you can see $f(p)$ heavily depends on chosen H and it change every time a market participant sends an order to the market. The function $f(p)$ would be really computer intensive if evaluated for every p. Therefore $f(p)$ is defined only for some values of p and is interpolated for the rest. These values are knot points and are defined by the prices of orders in recorded history H. Based on this belief function, GD also chooses the right shout-price it should bid or offer. The right value for shout-price is that price where the product of $f(p)$ and profit of the agent is maximized. Profit of the agent is $(l - p)$ limit price minus shout-price for buyer and $(p - l)$ shout-price minus limit price for sellers.

The exact quantitative notation would be for seller:

$$p_{ask} = \arg\max_{P} f_s(p)(p - l) \tag{3.19}$$

And for buyer:

$$p_{bid} = \arg\max_{P} f_b(p)(l - p) \tag{3.20}$$

3.2.2 Updates to GD

There were several updates to GD in order to satisfy the needs of different types of experiments. Das et al. (2001) modified the original GD algorithm to fit it in MAGENTA (the real-time asynchronous framework). The difference between GD and this modified version was in the treatment with persistent orders not dealt with in original GD. They addressed this problem by postponing record of unmatched orders to the history H. Unmatched orders were entered only after a grace period has expired. The length of history recorded was increased to a much larger value as well. The belief function was also exponentially weighted to emphasize the most recent terms more than the older ones. Tesauro and Das (2001) modified the GD in a different way. They called their modification MGD. It adds the probability of 1 to all values above the highest price in the history and 0 to all values below the lowest price in the recorded history for buyers and the exact opposite values for sellers after the interpolation of the belief function. There was one more modification of GD algorithm called GDX. Tesauro and Bredin (2002) developed GDX modification to use dynamic programming to price orders. It means that the GDX trader takes into account the effect of trading the current unit immediately, and the effect of trading in the future. The belief function is modified to account for this by discounting the future trading by a parameter γ. GDX agents therefore do not maximize merely the immediate profit but they maximize the overall profit over the entire trading period. They optimize the pricing process to do so. By the time Tesauro and Bredin (2002) published their work, they suggested that GDX algorithm with γ close to 1 may offer the best performance of any published CDA bidding strategy so far. We will look at it in more detail.

The dynamic programming process developed by Tesauro and Bredin (2002) is solved every time the agent has a possibility to quote a bid or offer. They apply dynamic programming which learns value functions representing long-term reward. They build dynamic programming on approximate formulation of agent-centric state description to overcome problems with unfeasible state-transition model, which would require tracking the history of every observable market event. Their agent-centric model consists of agent's current holdings M, time remaining until the close of trading T and agent's outstanding bids b if such bids constrain the legal bid actions, or if there are any costs of canceling open bids.

3.2.3 Definition of GDX

Tesauro and Bredin (2002) improves GD belief functions $f(p)$ by combining the method mentioned above with standard time-series forecasting methods to estimate future trade probabilities. They define the belief function as a function on the history H of market activity that estimates a scalar probability of an order being traded during some current or future time interval.

Tesauro and Bredin (2002) described their dynamic programming in discrete environment as following: H is the event history and T is the time remaining until the end of a trading day. The agent uses (H, T) to compute $f(p, +-q, t)$, function representing estimates of the probability that an order is traded with p denoting price, q quantity (negative for seller) and t representing time remaining. The agent first estimates the number N of future opportunities for submitting new bids or replacing the current ones. The agent then uses these information to calculate a table of expected values $V(x, n)$, where x is the agent's internal state consisting of agent's holdings M as well as any outstanding bids/asks, and n is the number of remaining bidding opportunities. In the absence of bid-switching costs $x = M$. Calculation of $V(x, n)$ starts by evaluating the terminal states $V(x, 0)$ using agent's private valuation and sunk costs of the holdings or in some markets $f(p, +-q, 0)$ which can be used to estimate fair market value of the holdings. Then the algorithm moves backwards from n to $n-1$ down to 0. Tesauro and Bredin (2002) defines function $s(x, p, +-q)$ as the immediate surplus obtained in state x by trading q at price p, $r(x)$ as the expected return from possession of the holdings and γ as the discount factor. They defined algorithm calculating expected values as:

Algorithm 1 State-space evaluation

for $n = 1$ to N **do**
  for all reachable states $\vec{x}(n)$ **do**
    $V(\vec{x}, n) = \max_{p,\pm\vec{q}}$(/* The value of trading */
      $f(p, \pm\vec{q}, t_n)[s(\vec{x}, p, \pm\vec{q}) + \gamma V(\vec{x} \pm \vec{q}, n-1)]$

      /* The value of not trading */
      $+(1 - f(p, \pm\vec{q}, t_n))\gamma V(\vec{x}, n-1)$

      /* The return on holdings */
      $+r(\vec{x})$)
  end for
end for

Source: Tesauro and Bredin (2002)

Then the agent can choose the optimal bid action at time remaining T, $(p^*, q^*)(T)$ to satisfy:

$$(p^*, \vec{q}^*)(T) = \arg\max_{p,\pm\vec{q}}\Big(f(p, \pm\vec{q}, T)[s(\vec{x}, p, \pm\vec{q}) + \gamma V(\vec{x} \pm \vec{q}, N-1)] + (1 - f(p, \pm\vec{q}, T))\gamma V(\vec{x}, N-1)\Big) \tag{3.21}$$

Where p^* is the optimal price which will be offered by the agent to buy or sell the amount q^*.

To adjust for CDA compatibility, Tesauro and Bredin (2002) make several adjustments. Agents have a fixed role either buyer or seller and bids and asks are for single unit. There is a fixed sequence of limit prices (seller costs or buyer values) for each unit that can be bought or sold represented as a vector L of length M in the binding increasing order for sellers and binding decreasing order for buyers. With these changes,the previous model has to be changed as well. Optimal amount q is set to $+1$ for buyers and -1 for sellers in all cases and the state description x simplifies to a single integer m representing the amount of units that the agent can buy or sell such as $0 <= m <= M$. It means that the agent trading the i-th unit at price p obtains surplus $s_i(p) = L_i - p$ for a buyer, or $s_i(p) = p - L_i$ for a seller. The expected value $V(m, n)$ similarly to discrete environment is computed starting at $V(m, 0) = 0$ for all m. Moreover $V(0, n) = 0$ for all n. Then Algorithm 2 is executed to find $V(m, n)$ for all the combinations.

Algorithm 2 Expected value computation in our model CDA.

for $n = 1$ to N **do**
  for $m = 1$ TO M **do**
    $V(m, n) = \max_p\big(f(p, t_n)[s_m(p) + \gamma V(m-1, n-1)] + (1 - f(p, t_n))\gamma V(m, n-1)\big)$
  end for
end for

Source: Tesauro and Bredin (2002)

The optimal price at time T, p*(T) is then given by

$$p^*(T) = \arg\max_p (f(p,T)[s_M(p) + \gamma V(M-1, N-1)] + (1 - f(p,T))\gamma V(M, N-1)). \tag{3.22}$$

Algorithm 2 and calculation of optimal price $p^*(T)$ is computed each time the agent becomes active and eligible to bid. This time consideration in the algorithm results into interesting behaviour. With more trading opportunities left till the end of a trading period, GDX offers more aggressive price in order to absorb extra-profit by waiting for other traders to make concession. In contrast, for low time remaining, the agent is motivated to trade quickly and therefore it submit more "honest" bids with lower profit closer to original GD strategy bids.

3.3 AA - Adaptive Aggressiveness

Third and the newest from the presented and used in my experiments algorithms, bidding strategy to be explained here is Adaptive Aggressiveness. Adaptive Aggressiveness referred to as AA developed by Vytelingum (2006) in his PhD thesis and presented by Vytelingum et al. (2008) is the dominant bidding strategy nowadays as considered by researchers. The AA strategy is predictive and history-based bidding strategy that software agents can use to bid in CDA. It is based on short-term and long-term learning behaviour as well as estimation of the equilibrium by moving average method. In the shor-term behaviour, it optimizes the aggressiveness in order to react appropriately to each transaction and prevailing market conditions. The level of aggressiveness determines the currently preferred trade-off between the probability that the transaction takes place and the profitability of transaction. In the long-term ,it tries to react to the change in the general equilibrium of the market.

3.3.1 Bidding Aggressiveness

Vytelingum (2006) emphasizes bidding aggressiveness as you can notice from the name of the algorithm as the most important part of this algorithm. It is principal because it decides how the agent is going to decide about the bid price. Whether it will be more aggressive in trying to increase its chance to transact but with not necessarily high profits or it will be more passive and bid at more profitable prices, but with lower chance to actually transact. When the agent is not able to transact it can choose to be more aggressive in order to increase its chance to be part of a transaction. In the opposite case, if the agent is able to transact, he might decide to decrease its aggressiveness to become more profitable. Vytelingum (2006) denotes aggressiveness r which range is [-1, 1]. Agent is aggressive with $r < 0$, neutral with $r = 0$ and passive with $r > 0$.

3.3.2 The Equilibrium Estimator

AA algorithm works with market equilibrium p^*. This information is not known a priori so the AA algorithm implements a moving average method to estimate market equilibrium price p^*. Vytelingum (2006) advocates moving average as an objective analytical tool which is more sensitive to price changes over a short time frame with lower weight on older transactions and which filters out the high-frequency components of the signal within the frame. Moving average estimation therefore provides the information about the direction of a trend with smoothed large price fluctuations. Given a set of N the most recent transaction prices, estimator of p^* is calculated

$$\hat{p}^* = \frac{\sum_{i=T-N+1}^{T} w_i p_i}{\sum_{i=T-N+1}^{T} w_i} \tag{3.23}$$

Where $w_T = 1$ and $w_{i-1} = \lambda w_i$ and where the vector $(w_{T-N+1}, \ldots, w_T)$ is the weight given to N most recent transaction prices $(p_{T-N+1}, \ldots, p_T)$ with T the latest one. Vytelingum (2006) set based on simulations $\lambda = 0.9$ and N to roughly the number of daily transactions in order to spot any converging daily pattern in history.

3.3.3 The Aggressiveness Model

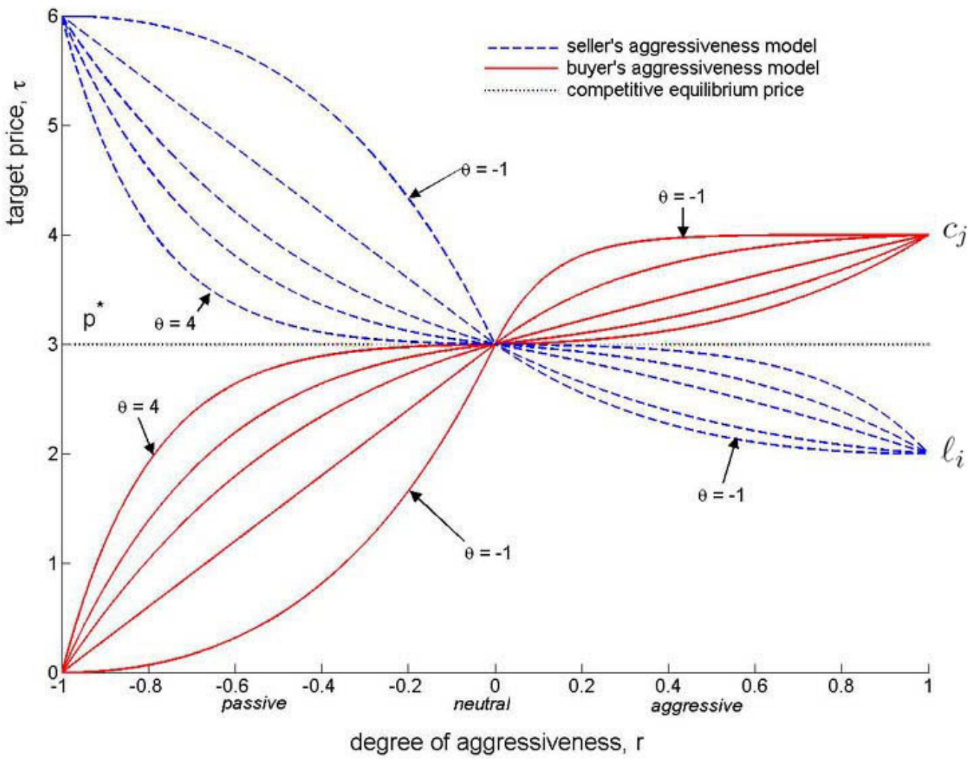
The role of Aggressiveness model in AA algorithm is to set the current target price, τ, given the current aggressiveness r. AA differs between two types of agents: intra-marginal and extra-marginal. Intra-marginal trader has the limit price above (below) the competitive equilibrium price for buyer (seller) and therefore it is probable it will transact. Extra-marginal trader has the limit price below (above) the competitive equilibrium price for buyer (seller) and therefore it is improbable that it will be part of a transaction. In the centralized mechanism with efficient allocation only intra-marginal traders transact while extra-marginal do not. In the decentralized system it might happen that because of inefficiencies even extra-marginal trader will transact while intra-marginal trader is only expected to transact but not sure.

AA differs between these two types of traders and assigns them different functions for target price.

Intra-Marginal Trader

In case of intra-marginal trader, if target price equal to market equilibrium $\hat{p}^*$ then the trader is neutral. If intra-marginal trader becomes passive, it means that it considers target price below the market equilibrium p^* for buyer and above the market equilibrium for seller in order to obtain higher profits than expected at $\hat{p}^*$. In contrast, aggressive intra-marginal buyer (seller) targets bids (asks) above (below) the competitive equilibrium price in order to increase the probability of realizing the transaction although with lower profit margin. The following figure Y shows how a theoretical intra-marginal buyer and seller set their target price based on their aggressiveness. p^* is in this case 3, c_j denotes limit price of buyer and l_i denotes the limit price or cost of seller. Based on aggressiveness, there are 3 certain spots for each intra-marginal trader. If r = 0, then the intra-marginal trader bids or asks at p^*. If $r = -1$ (passive trader), then buyer bids price 0 and seller asks the maximum ask, p_{max}, allowed in the market, although with small chance to transact. If $r = 1$ (aggressive trader), then buyer bids its limit price realizing no profit and seller asks its limit price l_i. Although these 3 points are clear, there is infinite solution space for the rest of values of r. This is solved by incorporating the parameterized function with parameter θ which determines the gradient of the function determining target price.

Figure 3.1: Graph for Intra-marginalAA trader



Source: Vytelingum (2006)

θ parameter allows the agent to specify the properties of the function or in other words to be more or less reactive in setting target price to the changes in aggressiveness. When θ is high, the magnitude of the gradient tends to 0 at $r = 0$ and conversely when θ gets lower. When θ is low, faster update of target price is possible as r changes. This is especially important when there is any market shock affecting the market equilibrium allowing the agent to react appropriately and not to bid or ask inadequately. When θ is high, target price reacts much slowly to change of aggressiveness at $r = 0$. As Vytelingum et al. (2004) describes, the effectiveness of the AA strategy depends on the appropriate θ and the appropriate θ depends on the prevailing market environment. When the AA faces a market with high price volatility, an agent is better off with lower θ to be able to explore the market. In the opposite case, when the market is relatively stable, too large shifts in target price would mean lower efficiency and profitability as well. Therefore, large θ is preferred.

The equation of target price for an intra-marginal buyer i is:

$$\tau = \begin{cases} \hat{p}^*(1 - re^{\theta(r-1)}) & \text{if} \quad r \in (-1, 0) \\ (\ell_i - \hat{p}^*)\ (1 - (r+1)\, e^{r\underline{\theta}}) + \hat{p}^* & \text{if} \quad r \in (0, 1) \end{cases} \tag{3.24}$$

Where $\underline{\theta}$ is defined this way:

$$\underline{\theta} = \frac{\hat{p}^* e^{-\theta}}{l_i - \hat{p}^*} - 1 \tag{3.25}$$

and the equation of target price for an intra-marginal seller j is:

$$\tau = \begin{cases} \hat{p}^* + (p_{max} - \hat{p}^*)\ re^{(r-1)\theta} & \text{if} \quad r \in (-1, 0) \\ \hat{p}^* + (\hat{p}^* - c_j)\ re^{(r+1)\underline{\theta}} & \text{if} \quad r \in (0, 1) \end{cases} \tag{3.26}$$

Where $\underline{\theta}$ is defined this way:

$$\underline{\theta} = \log\left[\frac{p_{max} - \hat{p}^*}{\hat{p}^* - c_j}\right] - \theta \tag{3.27}$$

Marginal traders have their limit price exactly at $\hat{p}^* = l_i = c_j$ and they form the limiting case of the equations above.

Extra-Marginal Trader

For the extra-marginal traders, equations mentioned above are not valid because the seller cannot ask below $\hat{p}^*$ and the buyer cannot bid above $\hat{p}^*$. In such situations the aggressive part of the equation is suppressed into line equal to limit price and the passive part is not related to equilibrium price but limit price instead. The following equation explains the relationship for the extra-marginal buyer:

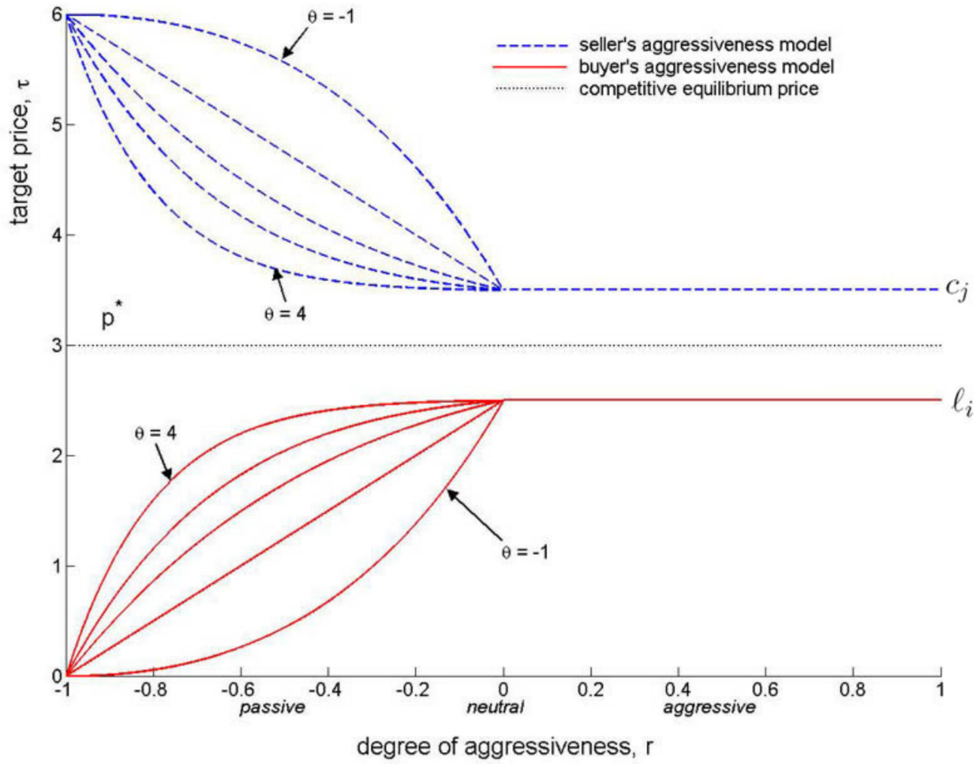
$$\tau = \begin{cases} \ell_i(1 - re^{\theta(r-1)}) & \text{if} \quad r \in (-1, 0) \\ \ell_i & \text{if} \quad r \in (0, 1) \end{cases} \tag{3.28}$$

And the following equation explains the relationship for the extra-marginal seller:

$$\tau = \begin{cases} c_j + (p_{max} - c_j) re^{(r-1)\theta} & \text{if} \quad r \in (-1, 0) \\ c_j & \text{if} \quad r \in (0, 1) \end{cases} \tag{3.29}$$

Aforementioned equations are shown in figure 3.2

Figure 3.2: Graph for extra-marginal AA trader



Source: Vytelingum (2006)

3.3.4 Short Term Learning

The agent pursuing AA strategy uses a set of learning rules to update its aggressiveness as a reaction to the transactions occurred in the market in order to incorporate it into its target price as well as equilibrium price estimation. More specifically, AA uses the Widrow-Hoff learning algorithm (same as used by ZIP) to increase or decrease its aggressiveness $r(t)$ at time t.

$$r(t+1) = r(t) + \beta_1(\delta(t) - r(t)) \tag{3.30}$$

$$\delta(t) = (1+\lambda)r_{shout}; \lambda = -0.05, 0.05 \tag{3.31}$$

Vytelingum (2006) calls $\delta(t)$ desired aggressiveness. Desired aggressiveness has such a value which enables the AA buyer to bid from a range limited by its limit price and the value slightly higher than the outstanding bid. For seller it is between its limit price and value slightly lower than outstanding ask. r_{shout} is the degree of aggressiveness that would force AA trader to bid or ask with the same price as observed last bid, ask, or transaction. As you can see desired aggressiveness $\delta(t)$is a function of r_{shout} with λ changing it by 5% in a profitable direction. The algorithm then adapts the aggressiveness in a direction to desired aggressiveness by the proportion of the difference equal to learning parameter β_1 (set to 0.5 in OpEx application by Marco De Luca).

Simply put, if the traders aggressiveness is below the aggressiveness of the market, trader uses this adaptive mechanism to increase its aggressiveness which will lead to a bid or ask more probable to participate in trade and too large aggressiveness it is decreased.

3.3.5 Long-Term Learning

As described in section 3.3.3, θ is a parameter affecting the long term behaviour and is reviewed after each transaction to improve efficiency of AA. The intuition is that different θ are the best within different market conditions. When the market volatility is high, it is good for AA trader to react with higher magnitude to the changes in aggressiveness and therefore optimal θ is lower. In contrary, if the market volatility is low, it is better to react with lower magnitude which means to have higher θ. This relationship is shown in Figure 3.1. The θ parameter is updated through a learning process based on the price volatility as an approximation of Smith's α-parameter (Smith, 1962). The following equation describes the learning mechanism:

$$\begin{aligned} \theta(t+1) &= \theta(t) + \beta_2(\theta^*(\alpha) - \theta_t) \\ \alpha &= \frac{\sqrt{\frac{1}{N}\sum_{i=T-N+1}^{T}(p_i - \hat{p}^*)^2}}{\hat{p}^*} \end{aligned} \tag{3.32}$$

Where $\beta_2 \in (0, 1)$ is the learning rate that determines how θ adapts in a similar way as aggressiveness r. $\theta^*(\alpha$ is a function determining the desired θ parameter based on the current price volatility α. Current price volatility is calculated from the last N prices where p_i is the price of transaction I and T is the most recent transaction. $\theta^*(\alpha)$ is calculated in the following way

$$\theta^*(\alpha) = (\theta_{max} - \theta_{min})(1 - (\alpha - \alpha_{min})/(\alpha_{max} - \alpha_{min})) e^{2((\alpha-\alpha_{min})/(\alpha_{max}-\alpha_{min})-1)} + \theta_{min} \tag{3.33}$$

Where $[\theta min, \theta max]$ is the range over which θ is updated and α_{max} and α_{min} are maximum and minimum α that occurs in the market. This function is arbitrary chosen by Vytelingum (2006) as a combination of ideal functions for different market environments since this function approximates the optimal θ well.

3.3.6 The Bidding Component

We have covered all the parts of AA strategy except of the bidding component. The bidding component decides based on bidding rules whether or not to bid or ask and if it decides that yes then it has to decide at what price. If the limit price is below the current bid obid for buyer or above the current ask $oask$ for seller, the trader does not perform any action because it is unprofitable and it waits. Otherwise, agent can submit a bid or ask in the market and it considers bidding rules to form a price. The equation to set a bid is as following:

$$bid_i = \begin{cases} o_{bid} + (\min\{\ell_i, o_{ask}\} - o_{bid})/\eta & \text{if first round} \\ o_{bid} + (\tau - o_{bid})/\eta & \text{otherwise} \end{cases} \tag{3.34}$$

And the equation to set an ask

$$bid_i = \begin{cases} o_{bid} + (\min\{\ell_i, o_{ask}\} - o_{bid})/\eta & \text{if first round} \\ o_{bid} + (\tau - o_{bid})/\eta & \text{otherwise} \end{cases} \tag{3.35}$$

Where $\eta \in [1, \infty)$ is a constant that determines the rate of increase (decrease) of the bids (asks) in Vytelingum (2006) work set to value 3. These equations are used only if the conditions of bidding rules are satisfied:

Figure 3.3: Bidding rules for AA

Bidding rule for Buyer *i* is:
If (l_i <= o_{bid}), submit no bid
Else
    If (first trading round) submit bid based on the equation above
    Else
        If (o_{ask} <= τ), accept o_{ask}
        Else submit bid based on the equation above
Bidding rule for Seller *j* is:
If (c_j <= o_{ask}), submit no ask
Else
    If (first trading round) submit ask based on the equation above
    Else
        If (o_{bid} => τ), accept o_{bid}
        Else submit ask based on the equation above

Source: Vytelingum (2006)

These bidding rules are extremely important in the beginning of the trading, where the agent do not have any estimate of the competitive equilibrium price. Therefore sellers start in much above and buyers much below their limit price. η gives a degree of convergence in the beginning of trading period. With a lower η traders might be too hasty and less profitable and the same with η too high. Vytelingum (2006) chooses $\eta = 3$ based on results in different environments as a good compromise. Latter, the agent decides based on prevailing market conditions and target price. It can either form a bid or ask or accept a bid or an ask from the other trader. Agent will do so if it is more profitable than a bid or an ask on target price.

Chapter 4

Newly developed strategy - ZIPOJA

This part of the work is dedicated to developing a new bidding strategy, which will then be compared to the other three strategies described in chapter 3, namely ZIP, GDX, AA. There is a vast literature developing and comparing bidding strategies. For example, Stotter et al. (2014) compare ZIP, ZIP related novel strategy ASAD, and AA. I mentioned other research in chapter 2.

As the name might suggest to the reader, the newly developed strategy is derived from the previously described bidding strategy ZIP. To be more precise, ZIPOJA is Zero-Intelligence-Plus enhanced with Oja's learning rule in its updating mechanism. The motivation is to allow to the algorithm variability in learning rates γ and β of ZIP strategy during the trading period. ZIPOJA starts with weights (described below) equal to the vector of ones which makes ZIPOJA equal to ZIP trader in the beginning of the experiment. Weights are updated during the experiment which distinguishes ZIPOJA from ZIP. To sum it, ZIP could be understood as a special case of ZIPOJA in which weights are constant and equal to ones.

4.1 Oja's learning rule

Oja's learning rule is a mechanism introduced by Oja (1982) falling into category of unsupervised learning. Oja's learning rule comes from Hebbian family of learning mechanisms originally coming from neuroscience. Hebbian family of learning mechanisms are all learning rules which are derived from Hebb's rule described by Hebb (1949).

Hebb's rule can be written in these two formulas:

$$y = \sum_{i=1}^{k} w_i x_i \tag{4.1}$$

$$\Delta w_i = \alpha x_i y \tag{4.2}$$

Where y is output dependent on inputs x_i with weights w_i. Weight of particular input is updated after each stimuli by product of that input and output multiplied by learning rate α. The Hebbian learning mechanism, which attempts to explain associative learning, can be understood as correlation learning because it utilizes directly the product of its input and output to derive the weight updates. In other words, it strengthens the reaction on stimuli which happened, and when the same stimuli happens again in the future, the reaction is then stronger. Inspired by biological neuroscience, Hebbian rule is often used in artificial neural networks to alter the weights between model neurons. It increases the weight (or strength) of interaction between two neurons in case they are activated in the same time and decreases the weight in case neurons are activated separately.

The disadvantage of Hebbian learning rule is that it is highly unstable in its original form as it promotes the dominant type of signal exponentially and suppresses less occurring signals exponentially as well. To overcome this problem, several methods are proposed and used in practice such as BCM model, Sanger's rule or Oja's rule (see Bienenstock et al. (1982), Sanger (1989), and Oja (1982). These models normalize and stabilize the learning rule in slightly different ways. I will only explain Oja's rule more thoroughly since it is the learning rule I have used in the newly developed strategy.

Oja's rule is the Hebb's rule enhanced by the so called 'forgetting' term which ensures that overly intensified connections/weights between inputs and output are discounted more than other weights. This is achieved by multiplicative normalization and solves all stability problems of this learning rule since the weight has a tendency to come back to value $w = 1$ and does so in balanced environment. Oja's rule is a computational expression of an effect which is believed to be an associative learning method in biological neurons. The following equations describe Oja's rule:

$$\Delta w = w_{n+1} - w_n = \alpha y_n (x_n - y_n * w_n) \tag{4.3}$$

$$y_n = x_n * w_n \tag{4.4}$$

The difference from Hebb's rule is in the aforementioned 'forgetting' term (the second term in parenthesis in equation 4.3). I use Oja's rule to expand ZIP bidding strategy and therefore I find the name ZIPOJA the most suitable for describing the content of the bidding strategy.

4.2 Definition of ZIPOJA

As mentioned above, ZIPOJA is based on ZIP bidding strategy. I will briefly review main concepts of ZIP here, but see section 3.1 of this thesis for a more detailed description, or refer directly to the creator of ZIP: Cliff (1997).

4.2.1 From ZIP to ZIPOJA

ZIP is a predictive bidding strategy based on updating its future profit margin μ which the algorithm thinks is optimal for current prevalent market environment. Profit margin is understood as the percentage difference from its limit price which is given to the algorithm in advance. The price is therefore set as following:

$$p(t) = \lambda(1 + \mu(t)) \tag{4.5}$$

where p is price at time t, λ is its limit price and μ is its profit margin positive for sellers and negative for buyers.

Profit margin for the next period according to the following equation)

$$\mu(t+1) = (p(t) + \Gamma(t))/\lambda - 1 \tag{4.6}$$

is based on price at time t and Γ representing the change of price from current period to the next period. These two terms are divided by limit price λ, and one is subtracted to obtain profit margin.

Change of price between current and future period, $\Gamma(t)$, is the discounted (by learning coefficient γ set between 0.2 and 0.8) change in price in the last period ($\Gamma(t-1)$), plus the distance of current price from the target price denoted as $\Delta(t)$, which is discounted by coefficient $(1-\gamma)$. The relation is written in this way :

$$\Gamma(t) = \gamma\Gamma(t-1) + (1-\gamma)\Delta(t) \tag{4.7}$$

Δ, a distance of current price from the target price, is put into equation in the following way:

$$\Delta(t) = \beta(\tau(t) - p(t)) \tag{4.8}$$

Where β is a learning rate coefficient set by Cliff (1997) to 0.5 and τ is the target price which is derived from a current shout q, to which the bidding strategy is reacting to. Target price τ is in the neighbourhood of the shout price q and it is randomized (as you can see in equation below) in order to keep traders active and challenge the current prices in case q is close to p,

$$\tau(t) = R(t)q(t) + L(t) \tag{4.9}$$

where L is random uniformly distributed variable from [0.00, 0.05] for increases in price and [-0.05, 0.00] for decreases in price. R is random uniformly distributed value from [1.00, 1.05] for price increases and from [0.95, 1.00] for price decreases.

Each ZIP trader also follows the mechanism: if the last shout was accepted at the better price than his calculated optimal price, trader should increase his profit margin. If it was from the opposite side and it was worse, then the trader should decrease his profit margin. If the last shout from the same side was not accepted at worse price than current estimated optimal price of an agent, then this agent should lower his margin.

ZIPOJA differs from ZIP only in the way how future profit margin is calculated. The rest is kept the same including its interaction with the market and activation based on shout prices occurring in the market.

ZIPOJA bidding strategy calculates price in the same way as ZIP (by equation 4.5) and updates its profit margin as seen in equation 4.6. The difference lays in equation 4.7 where Oja's rule is applied. The calculation of price change Γ is explained by the following equation:

$$\Gamma_i(t) = W^T(t)X(t) = \sum_{i=1}^{k=2} w_i x_i = w_1(t)x_1(t) + w_2(t)x_2(t) \tag{4.10}$$

Where w_i for all i is set initially to 1 and represents weight of factor x_i, and x_i is factor affecting change of optimal price Γ. The x_2 can be understood as Δ from equations of ZIP and x_1 is last period Γ. Therefore the equations can be written in the following way:

$$x_1(t) = \xi\Gamma(t-1) \; and \; x_2(t) = \psi(\tau(t) – p(t)) \tag{4.11}$$

Learning rate parameters ψ and ξ are here to affect the size of the effect of statement on current Γ since the weights start at 1 and are normalized by Oja's rule to come back to this value in balanced environment. ψ and ξ are theoretically independent but to make it completely comparable with ZIP bidding strategy I use conversion from ZIP's γ and β. In order to have the same effect of current distance of price and target price and the last change of price:

$$\psi = (1-\gamma)\beta \; and \; \xi = \gamma \tag{4.12}$$

This conversion of parameters is not necessary but it is important to put ZIPOJA into the same notation as ZIP to see the similarity and to set these parameters in a comparable way in experiments. With this change I can basically write equation 4.10 and 4.11 in this way to see that only weights are added into ZIP's equation 4.7:

4.2.2 Recapitulation of ZIPOJA equations

In order to see all equations in one place, this subsection provides the overview. For description, please refer to subsection 4.2.1 above.

$$p(t) = \lambda(1 + \mu(t)) \tag{4.13}$$

$$\mu(t+1) = (p(t) + \Gamma(t))/\lambda - 1 \tag{4.14}$$

$$\Gamma_i(t) = W^T(t)X(t) = \sum_{i=1}^{k=2} w_i x_i = w_1(t)x_1(t) + w_2(t)x_2(t) \tag{4.15}$$

$$x_1(t) = \xi\Gamma(t-1) \; and \; x_2(t) = \psi(\tau(t)–p(t)) \tag{4.16}$$

$$W(t+1) = W(t) + \Delta W(t) \tag{4.17}$$

$$\Delta W(t) = \alpha\Gamma(t)(X - \Gamma(t)W(t)) + \epsilon \tag{4.18}$$

And in order to be comparable with ZIP, I also include equation 4.8. Which means that with $W(1)$ equal to all-ones-vector, this bidding strategy is equal to ZIP in the beginning trading period.

4.3 Imposed restrictions on Oja's rule

In order to avoid very infrequent but possible deviation of weights from calm region around and between 0 and 1, the restriction on weights has been made. After each calculation of appropriate weight for next period code, the weight is set to 2 if it should exceed 2 and it is set to 0 if it should become negative. This is done only very rarely.

Another restriction is imposed on Γ. If the current price together with current Γ are supposed to exceed limit price, Γ is reduced (increased) in case of positive (negative) change for buyers (sellers). The technical restriction of Γ to work well inside OpEx restricts Γ in case it should lead future price to exceed maximum or minimum price allowed for a particular instrument traded. Then Γ is set for such a value that price reaches just the constraint.

4.4 Setting of parameters for ZIPOJA

ZIPOJA decides based on several parameters which have to be set. I will set the common ones with ZIP to values from uniform distribution in the same way that Cliff (1997) did in his work and that De Luca and Cliff (2011b) followed as well. Thanks to this, ZIPOJA will be fully comparable with ZIP and it will be obvious whether adding Oja's learning rule is enhancing this bidding strategy or not.

The parameters and values to be set are $\mu(1)$, $\Gamma(0)$, $W(1)$,R and L, ϵ, α and ψ and ξ which depends in my setup on β and γ to be fully comparable with ZIP strategy.

The following parameters are set in the same way as Cliff (1997) and De Luca and Cliff (2011b) set them. Profit margin $\mu(1)$ will be drawn uniformly from range [0.05; 0.35], $\Gamma(0) = 0$, R is for increases uniformly drawn from [1.00; 1.05] and for decreases from [0.95; 1.00]. L is drawn uniformly from [0; 0.05] for increases and from [-0.05; 0] for decreases. β is set randomly with uniform distribution between 0.1 and 0.5 and γ between 0.2 and 0.8. These two affect new parameters ψ and ξ according to equation 4.12 in subsection 4.2.1.

New parameters for ZIPOJA to be set are vector of weights at first period W(1) equal to all-ones vector, α learning rate of Oja's rule which is recommended to be very low. I have decided to draw it uniformly from range of 0.0005 and 0.005. The lower than 0.0005 values changed weights insufficiently and in relatively short experiments did not change ZIPOJA from ZIP substantially although clear trend was visible. The values above 0.005 meant too quick learning rate and in setting of X factors to the difference from target price and the last change, too large value for α often led to instability and not very competitive estimator of adequate price. The last to be set is ϵ variable introduced in 4.2.1 subsection, taking values uniformly from [-0.01; 0.01] and ensuring that weights will not keep absolutely stable.

Chapter 5

Methodology for comparison of strategies

My research goal is to compare strategies against each other in the agent vs. agent trading experiment. There are a lot of studies comparing trading strategies against each other. Each one of them employs different set of comparison tools. I will present some of them in the next section 5.1 and detail each method used in this work afterwards.

I will vary the number of trading strategies in the experiments. My symmetric experiments will have from one strategy (homogenous environment) to four strategies. I will change the proportions of the strategies in the agent population for two and three-strategy experiments similarly to Walsh et al. (2002) and perform similar dynamic analysis of mixed trading. I will graphically show superior trading strategy for each combination of trading strategies based on percentage of won strategies.

Then I will run an experiment with all four strategies in symmetric and balanced game (2 buyers and 2 sellers for all four strategies) to show how they interact in a common market.

5.1 Comparison methodology used in research

I will take inspiration from these studies about the comparison methods of trading strategies. De Luca and Cliff (2011b) compare pair of trading strategies (AA, ZIP, GDX, GD) in balanced (6 agents of one strategy vs. 6 agents of the second strategy) and symmetric (same number of buyers of one strategy as sellers of this strategy) experiment, and measure the percentage of won rounds by one or the other strategy based on the average profit/surplus.

Tesauro and Bredin (2002) compared GDX, GD and ZIP against each other in balanced symmetric (22 agents vs. 22 agents) and in unbalanced symmetric (1 agent vs. many agents) experiments. They measured the performance of the strategies by number of won rounds and surplus difference between the compared strategies.

Walsh et al. (2002) explain their new method of analyzing complex games on comparison of ZIP, Kaplan and GD trading strategy in three-strategy symmetric experiments with 20 agents and with various composition of agent population. They try to find whether there is an optimal strategy mix of agent population (mixed strategy Nash equilibrium). They use computationally intensive method to assign payoffs to each combination of these three strategies and then they compute the dynamic analysis by replicator dynamics (Weibull 1995). This approach helps them to map dynamic development of strategy implementation in agent population. They find basins of attraction of all equilibria and graphically map the dynamic movement to them.

Vytelingum (2006) and Vytelingum et al. (2008) compare AA against GDX and ZIP in a dynamic analysis in the similar way as Walsh et al. (2002) with 20 agents (10 buyers and 10 sellers). The main difference is in having non-symmetric game (allowing for different number of buyers and sellers from one strategy) and in comparison of only two strategies at each moment in a dynamic analysis using replicator dynamics. Thanks to this, they were able to show disproportions of buyer and seller optimal combinations. Moreover, they show efficiency indicator which they believe is the main driver of success in the real application.

5.2 Trading surplus

In order to identify better performing strategy, the basic indicator of trading surplus is used most often. It serves as a base to develop more advanced comparison methods. Surplus is obtained from trading of assigned units to each trader below their limit price for buyer and above for seller. Surplus then can be compared to obtain surplus difference between two strategies as Tesauro and Bredin (2002) did. Surplus of one trading strategy is the average surplus of all trading agents using this strategy. I will use trading surplus to compute other indicators such as won game rounds or efficiency ratio.

5.3 Won game rounds

The Won game trials metrics speaks for itself. Each specification of proportion of strategies is run multiple times. De Luca and Cliff (2011b) use 1000 trials; Tesauro and Bredin (2002) use 1000 trials; Vytelingum et al. (2008) use 2500 trials. Researchers then report either the number of won rounds or in percentage metrics out of all rounds. Winning strategy is decided based on average surplus of agents performing the same strategy in comparison to average surplus of another strategy. I will use won game rounds metrics in addition to other metrics to report on two, three and four-strategy experiments. The disadvantage of this indicator is that it cannot differentiate between a slightly but consistently better strategy and radically and consistently better strategy. For this reason, efficiency ratio might show more depth in the relationship.

5.4 Efficiency ratio

As Vytelingum et al. (2008) define, the market efficiency is the ratio of all agents' surpluses in the market to the maximum possible surplus that would be obtained in an allocation where the profits of all buyers and sellers are maximized.

The efficiency of a bidding strategy is the ratio of the profits of the agents adopting that strategy to the maximum possible surplus resulting from an efficient, centralized allocation scenario. In the homogenous scenario, efficiency of a bidding strategy is equal to market efficiency. In heterogenous scenario with more strategies implemented, the weighted (by strategy share on total population) mean of all efficiencies of a bidding strategies is the market efficiency.

5.5 Dynamic analysis

The most advanced approach to be used here is a dynamic analysis based on Walsh et al. (2002) approach. Similar approach is implemented by Vytelingum (2006) and Vytelingum et al. (2008) but only in 2 strategy space with possible asymmetric combination (different number of buyers and sellers of one strategy). The main purpose of this analysis is to find the mixed strategy Nash equilibria in the 3 dimensional strategy space or in other words to find whether one strategy purely outperforms all the others or whether there is a combination of strategies, from which no agent wants to differ. Not only equilibria, but paths to them are computed by these researchers. They use replicator dynamics (Weibull 1995) to do that. I will follow their approach with one exception. Our dynamic analysis will be based on ideal change in a particular point, not empirically calculated by replicator dynamics.

These dynamics, equilibria, and saddle points are then shown in a ternary graph representing all combinations of two and three-strategy experiments. It is impossible to graphically show these dynamics for four-strategy experiments since three-dimensional tetrahedron would be needed to show all the combinations. The methods of representation of tetrahedron to two-dimensional space would be insufficient and confusing in this application, so although having four strategies to compare, I will stay with dynamic analysis only to symmetric three-strategy space which is possible to show in two dimensional space in the same way as Walsh et al. (2002) did. However, I will show this ternary graph for all four sets of three-strategy experiments (ZIP, GDX, AA; ZIP, GDX, ZIPOJA; ZIP, ZIPOJA, AA; ZIPOJA, GDX, AA).

5.5.1 Heuristic payoff table

In order to dynamically analyze the multiple-strategy space, there are several steps which have to be done first. First step is to obtain heuristic payoff table that specifies the expected payoff to each strategy in all three-strategy experiments. That means that for each composition of total agent population I will know how any of included strategies performs. Expected payoff of a strategy in an experiment is simply the average of payoffs for all agents following that strategy in a particular experiment. Experiments often have many rounds (in my case

1000) to guarantee some sort of certainty of results.

The heuristic payoff table is a reduced model of the game of multiple strategies followed by multiple trading agents choosing and changing which strategy they follow. Moreover, each agent performs many actions to determine whether it is successful or not. Thanks to the heuristic payoff table, genesis of strategies does not have to be directly analyzed in extremely complicated system as limit order market is.. With this table, complex game as explained is reduced into one-shot game, in which I can just focus on the change of heuristic strategies rather than their development or basic actions. With this model this complex problem in a normal-form game by the standard game-theoretic models. You can find more on the issues of this approach in Walsh et al. (2002).

Thanks to simplifying assumptions on symmetry of experiments and that the strategies are drawn independently from the same distribution, the number of entries reduces dramatically for the heuristic payoff table. The change is from Sa entries where S is number of strategies and a number of agents to merely $\binom{A+S-1}{A}$ entries. That means in my case of 12 agents but in a symmetric experiment(so I can theoretically think only about 6 traders trading in both ways and with restriction to 3 strategies per experiment) only $\binom{8}{6}$= 28 entries. Moreover, I have 4 combinations how to choose 3 strategies out of 4 and therefore there are 4 heuristic payoff tables. Some entries are in more tables so the overall number of entries to be computed is 28+21+15+10=74.

The computation of such a heuristic payoff table is already manageable but it is still very computationally intensive work since every entry is an experiment with 1000 rounds to ensure accuracy and every round consist of 300 steps where actions of all agents are evaluated.

5.5.2 Equilibrium Computation

At the start of the game, strategies are assigned to all agents to match a particular experimental setup and shares of strategies on agent population. The agent i has therefore the probabilities $\hat{p_i} = (\hat{p_{i,1}}, ..., \hat{p_{i,S}})$ to be assigned to one of the strategies $j \in 1, ..., S$, with constraint $\hat{p_{i,j}} \in [0, 1]$ and $\sum_{j=1}^{S} \hat{p_{i,j}} = 1$.

We can also say that agent i plays mixed strategy $\hat{p_i}$. $\hat{p_i}$ is decided before the experiment and reflects the particular combination of strategies on agent population appropriate to a specific entry of heuristic payoff table.

The expected payoff of agent i is a real-valued function u which depends on assigned strategy to the agent i and agent population composition. In other words, it assigns value for each agent based on his average profit in all rounds of a particular experiment. The expected payoff for strategy is then computed as average of expected payoffs for all agents following this strategy. These three strategy expected payoffs are one entry into heuristic payoff table on a position representing shares of strategies on agent population for a given experiment (e.g. 33% of ZIP, 33% of GDX, 33% of AA is one possible combination of strategies on agent population, so average payoffs for all agents will be computed, then averaged into strategy average payoffs and then filled into heuristic payoff table on a position representing the combination 33% of ZIP, 33% of GDX, and 33% of AA).

To put it mathematically, when agent i plays pure strategy j, I denote $\hat{p_i} = e^j$.Then

$$u(e^j, \hat{p}_{l-1}) \tag{5.1}$$

denotes the expected payoff to an agent i for playing pure strategy j, given that all other agents play their mixed strategies $\hat{p_{-i}}$ (or in other words, they play the rest of strategies assigned to this setup in the right proportion). The payoff to agent i of the mixed strategy $\hat{p_i}$ is then

$$u(\hat{p}_l, \hat{p}_{l-1}) = \sum_{j=1}^{S} u(e^j, \hat{p}_{i-1})\hat{p}_{i,j} \tag{5.2}$$

In words, weighted average payoff of pure strategies with weights equal to shares of strategies on total population. The average payoff of a particular strategy is denoted as $u(e^j, p)$ and the average payoff to all agents in this strategy-composition specification p is $u(p, p)$. The difference of $u(e^j, p)$ and $u(p, p)$ is surplus of the strategy j over the average trader in strategy-shares on agent population specification p.

The values can obviously be calculated only for certain strategy-composition specifications given by researcher. The rest of values are interpolated.

Finding Nash equilibria can be computationally challenging problem. Nash equilibrium can be expressed in various equivalent formulations, each suggesting different solution methods as mentioned by McKelvey and McLennan (1996). I will stick to the method used by Walsh et al. (2002). They formulate Nash equilibrium as a minimum of a function:

$$v(p) = \sum_{j=1}^{S} \left(\max\left[u(e^j, p) - u(p, p), 0\right]\right)^2 \tag{5.3}$$

In this formulation, the mixed strategy p^* is a Nash equilibrium if and only if it is a global minimum of v. We can guarantee that a minimum is global if its value is zero (McKelvey and McLennan 1996).

In this work I will use non-linear optimizer Amoeba in the same way as used in Walsh et al. (2002) and explained in Press et al. (1992) to find the zero-points of v in performed experiments. Amoeba optimizer, named downhill simplex method or Nelder-Mead optimization method originally proposed by Nelder and Mead (1965), is an optimization technique which does not require any gradient computation and therefore it can be applied to nonlinear optimization problems for which gradients may not be known. Thanks to these properties, Nelder-Mead method is widely used even though it is computationally more intensive than other optimization algorithms. Nelder-Mead method uses heuristic search method with specified starting point from where it calculates polytype of n+1 vertices in n dimensions. This polytype moves every step based on values to more optimized values. It is possible that this method will not find global but only local optimum. Thanks to the specification of objective function, it is known as mentioned earlier that global minimum occurs only if value is equal to zero. I will use built-in optimization method in Matlab optimization toolbox to find minimums of function $v(p)$.

5.5.3 Dynamic analysis

Nash equilibria found by optimization of v function provide a sufficient view on multi-agent system in static environment. Dynamics behind getting to these equilibria is as important as static properties. In decentralized systems it is usually foolish to assume that all agents have enough information/ common knowledge and the resources to compute equilibria so it is still interesting question which equilibrium is chosen from which place and what path does the system undergoes in reaching it.

The models of adaptation and learning in dynamic environment are studied, developed and applied in voluminous literature. As Walsh et al. (2002) paraphrase Fudenberg and Kresp (1993), Fudenberg and Levine (1993) and Jordan (1993), the positive theoretical properties of equilibrium convergence for general games of A agents and S strategies have not yet been established. Walsh et al. (2002) choose well-developed model from evolutionary game theory showed by Weibull (1995) for their purposes. They posit a large population of N agents from which $A << N$ agents are randomly selected at each time step to play the game. At any given time, selected agents play pure strategy out of a set of strategies S. The share of population playing strategy j is pj. The pj values define strategy shares vector p. If N is large enough, pj can be treated as continuous variable. I will stick to simplified dynamic analysis compared to replicator analysis mentioned above although using its formalism to model the evolution of p in dynamic environment. The change in share of stategy j at each point p in three-strategy space is specified by this equation:

$$\dot{p}_j = [u(e^j, p) - u(p, p)]p_j \tag{5.4}$$

Where $u(p, p)$ is the population average payoff and $u(e^j, p)$ is the expected payoff to agents currently using pure strategy j. This equation models the tendency of strategies with above average payoff to attract more followers and strategies with below average payoff to suffer defections, since $\dot{p}_j$ represents the change in the proportion of strategy j. If $\dot{p}_j = 0$ for all strategies, that means that this point is a stable point, which corresponds to a Nash equilibrium and can be called affixed point. Moreover, if strategies converge to this point, this point can be called an attractor. Nash equilibria with larger basin of attraction are more likely to occur in the population if it isassumed that every initial population state is equally likely (Walsh et al. 2002).

The only disadvantage of this model of dynamic analysis is that agents do not have minimal information requirements beyond their own actions and payoffs since they have to know $u(p, p)$. Based on this, this dynamic analysis does not copy dynamic analysis in reality. The same population dynamics asa change into the ideal direction or replicator dynamics can be achieved by more realistic replication by imitation model (Weibull 1995). As Walsh at al. (2002) explain, an agent switches to the strategy of a randomly chosen better performing opponent in this model.

Chapter 6

Experiments

Results from artificial experiments will be presented in this chapter. I have used OpEx trading simulator in order to compare bidding strategies in different setups. The intention of these simulations is

- to provide more robust results on comparison of ZIP, GDX and AA bidding strategies problematic since AA is reported to be the most efficient and ZIP, GDX are often used as benchmark for comparison in literature.
- the fact that there is frequent literature which compares bidding strategies and the results can be compared to those in the past.
- to provide evidence of newly developed ZIPOJA bidding strategy efficiency and its performance compared to other bidding strategies.

To name only some of relevant literature, ZIP, GDX and AA are compared in Vytelingum et al. (2008) in two strategy game with variation in number of buyers and sellers for each strategy. They calculate dynamic analysis and mixed strategy equilibrium of these simulations as well. De Luca and Cliff (2011b) compare ZIP, GD, GDX and AA against humans and against each other in two strategy game with the same number of buyers and sellers as well as the same number of agents playing each of two strategies. Walsh et al. (2002) compare ZIP, GD, and Kaplan in a symmetric (same number of buyers and sellers playing one strategy) three strategy game with variation in proportions of strategies played by agents. They calculate dynamic analysis and obtain results on how agents change strategies to be more successful. They calculate mixed strategy equilibria in three strategy space and provide graphical interpretation of these relationships in a very informative and intuitive ternary diagram which served as an inspiration for graphical output of this work. Tesauro and Bredin (2002) compare ZIP, GD and GDX in two strategy game with equal share of agents playing both strategies and another scenario with one of one strategy versus many of another strategy.

6.1 Experimental Setup

I will show results of homogenous trading groups and two and three-strategy games with variable shares of agents in population for each strategy and four-strategy games. All experiments will be symmetric, that means they have the same number of buyers and sellers pursuing one strategy. This is the most important in case of three strategies with dynamic analysis since only three strategies with their dynamics can be presented on a two-dimensional paper.

6.1.1 DES - Discrete Event Simulator

I will run experiments in OpEx 1.2 trading simulator offering preprogrammed trading environment to which agents join and just send shouts which are put into order book or executed immediately by trading simulator. More concretely, I will use Discrete Event Simulator (DES) which is turn-based and pick one of each agents at random every turn. This is certainly not Continuous Double Auction (CDA) but with trading agents trading much faster than human traders, one could simply assume that random pick is a representation of the fastest reaction on stimuli in CDA. One might nearly agree that DES could be taken as sufficient for CDA representation. In DES, one trading round consists of 300 turns in which agents have the opportunity to trade their assigned units.

DES simplification is necessary when one takes into account the vast amount of experiments to be run and that CDA should be run in real-time. More specifically I will run 74 experiments having 1000 rounds each in 5 groups of 200 rounds. Calculation of results consisting approximately from 5 million rows of transaction data took fair amount of time in DES as well. It took around 3 days of computation to run all the experiments on my laptop with 1.5GHz processor.

6.1.2 Agents and strategies

Regarding number of agents, it will stay strictly constant except for the experiment with all four strategies present. The number of buyers is the same as sellers and is equal to 6. That accounts for 12 traders in total. The number of agents following one particular strategy varies across the experiments but not in one experiment. The number of agents following one bidding strategy varies from 0 to 12 by step of two (the same number of buyers and sellers all the time). This represents all the possibilities of homogenously populated experiments, two-strategy games and three-strategy games as well, having 12 traders in total in each experiment. The only exception to this rule is a four-strategy experiment which has two buyers and two sellers following each strategy. That means 16 agents in total for this experiment.

Regarding strategies, I will use prebuilt ZIP, GDX and AA in OpEx 1.2. ZIPOJA is built based on ZIP strategy. Depending on OpEx pre-coded strategies might be risky, since there might be minor differences against the versions of strategies developed by their creators. For example, neither Vytelingum (2006) nor Vytelingum et al. (2008) provided source code for its novel AA strategy, so researchers and traders had to construct AA again by themselves.

OpEx, used for example by De Luca and Cliff (2011b), provides one of a few publicly available versions of AA. Nevertheless, OpEx offers sufficient foundation for running intended experiments which will follow and which extend agent versus agent experiments done not only by De Luca and Cliff (2011b) in OpEx, too.

6.1.3 Units to trade

All traders get the same amount of units to trade with the same limit prices with no regard to which strategy they follow. The only difference is between buyers and sellers which is a necessary condition to get market moving. All the buyers get eleven units for trading with limit price 250, 240, 230, 220, 210, 200, 190, 180, 170, 160 and 150. Sellers get the same limit prices for their units but the sequence starts from the lowest one. When the trader trades his most profitable unit, he starts offering the second most profitable one and so on. Usually traders sell 6 units and the equilibrium on the market is found around 200 with no one willing to buy (or sell) a unit above (below) their limit price. Sometimes some traders sell more units than 6 which means they are effectively grasping profit from other agents on the same side who will not be able to participate on trade because there will not be any counter-party. Another option is that trader will be quick in trading his first 6 units and will enter market with his less profitable units at time when there will still be possible counter-party at that price.

Maximum average surplus is artificially calculated value maximizing surplus for all traders together. Inmy setup the optimal equilibrium price is 200 for all experiments and therefore maximum surplus is 150 for all buyers and sellers as well since sellers have mirrored prices of buyers. Efficiency of traders is calculated from successfulness of reaching this amount.

6.1.4 Instrument traded

OpEx allows to choose properties of an instrument traded. The instrument was kept constant with price tick 1 (trades could be traded only at integer prices). There is a set minimum price of 125 and maximum price of 275 and therefore no trade can be made behind these values.

6.2 Homogenous environment and two-strategy games

First results to be presented here are from homogenous environment and from two-strategy game. As explained in subsection 6.1.2, Iwill vary number of buyers and sellers of one strategy from 0 to 6 and assign another bidding strategy to the rest of agents. Therefore, the possible setupss are:

- 6 buyers and 6 sellers from one strategy - Homogenous environment - I will indicate it 6:0 or 0:6.
- 5 traders for each side using one strategy and 1 trader on each side using another strategy - One vs. many setup - Indication 5:1 or 1:5

- 4 buyers and sellers with one strategy and 2 buyers and seller using another strategy - Unbalanced setup - Indication 4:2 or 2:4
- 3 traders buying and 3 traders selling on both sides - Balanced setup- Indication 3:3

There are 6 pairs of strategies to be presented to cover all combinations. For each pair, there will be a short subsection presenting results and relevant comments.

6.2.1 ZIP vs. GDX

The first pair to be introduced here is the pair of the oldest strategies tested in this work. ZIP was invented by Cliff (1997) and GDX was invented by Tesauro and Bredin (2002). The reader can see an overview of seven relevant experiments in the following table. From the left side, there ZIP pure strategy but share of ZIP in population decreases and share of GDX increases till GDX pure strategy.

We can see that GDX clearly dominates ZIP in all scenarios where GDX is present. This is in line with previous literature (Tesauro and Bredin (2002) and Vytelingum et al. (2008)). The number of won rounds based on average profit extracted by agent pursuing the particular strategy is quite stable for GDX and ZIP for all experiments with obvious exception of homogenous environment. Efficiency of market remains quite stable and reaches very high values above 99% with one exception. Homogenous environment filled with GDX has the lowest efficiency ratio out of all conducted experiments. Based on check of the data, the reason is that some GDX buyers set their price incredibly low in the beginning. The rest of buyers build the belief function based on first shouts and then all the bids get down. It usually takes almost all turns of that round for the price to recover, which results in less total number of trades and consequently and logically lowers efficiency since efficiency is measured against the maximum theoretical surplus. An important thing is that as long as there are different bidding strategies, GDX is capable of competitive trading so this behaviour does not affect comparison with other strategies at all.

6.2.2 ZIP vs. AA

The next pair to evaluate is Zero-Intelligence Plus against Adaptive Aggressiveness. As you can see, AA dominates ZIP almost clearly, but with not prevalent dominance. Nonetheless, the domination is not that strong as in case of ZIP vs. GDX and it has one exception with one ZIP vs. many AA experiment. ZIP trader is able to gather more profit than the rest of AA traders in 589 rounds out of 1000.

This is not happening by a coincidence. We will see in GDX vs. AA and ZIPOJA vs. AA that AA is generally weak in the case when large share of buyers follow this bidding strategy or in other words, one trader of the competing strategy is able to use the homogenous competition of AA and gather more profit. This might be just the result of poor performance in this setup (one vs. many setup or different aspect of the setup - for example total number of traders or order of limit prices of units) or the result of the way how AA is designed in

Table 6.1: **Two-strategy experiments - ZIP vs. GDX**

ZIP vs. GDX	**6:0**	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6**
ZIP won rounds	1000	344	353	273	342	382	0
GDX won rounds	0	656	647	727	658	618	1000
ZIP efficiency	99.32%	98.35%	97.85%	95.43%	95.18%	94.81%	-
(s.s.d.)	(0.68%)	(2.74%)	(4.39%)	(6.98%)	(10.44%)	(18.56%)	-
GDX efficiency	-	105.32%	103.29%	104%	102.07%	100.63%	55.31%
(s.s.d.)	-	(13.15%)	(8.61%)	(6.94%)	(5.28%)	(3.94%)	(32.6%)
Total efficiency	99.32%	99.51%	99.66%	99.72%	99.77%	99.66%	55.31%
(s.s.d.)	(0.68%)	(0.59%)	(0.51%)	(0.47%)	(0.49%)	(0.66%)	(32.6%)
Surplus difference	-	10.46	8.17	12.86	10.34	8.73	-
Winner	**ZIP**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy.

OpEx 1.2. AA as defined in OpEx 1.2 cannot initialize first trade and therefore it has to wait for another strategy to build starting order-book. This certainly makes AA slower at the start of round. According to data, AA is not recovering for the rest of 300 turns of trading day/round. Even though AA is able to gain much more from each trade on average than ZIP trader in this setup, ZIP trader completes more trades. Nevertheless, this might not be the only reason why ZIP performs better in this setup. As you will see later (see 6.2.3 AA vs. GDX subsection), just different market setup can have its impact as well.

The efficiency of the market is high for homogenous ZIP populated market and decreases with increasing AA share in the population. We can only conclude that adding AA into ZIP population decreases efficiency of ZIP linearly except the one ZIP vs. many AA. The development of overall efficiency might be attributed to the inability of AA to open first trade. Because of this I was unable to get results for homogenous AA environment. To keep this AA comparable with AA from De Luca and Cliff (2011b), no change to AA was made. Value of 1000 won rounds for homogenous AA is just logical although not backed by data. That is the reason why the rest of values in this column is missing.

6.2.3 GDX vs. AA

The third pair of strategies to be compared is GDX vs. AA. This pair is the most interesting one since it shows the most balanced results. This pair is also important since according to results in the following table, GDX challenges the overall supremacy attributed to AA from previous research.

We can see that AA dominates clearly one AA vs. many GDX and unbalanced 2 AA vs.

Table 6.2: **Two-strategy experiments - ZIP vs. AA**

ZIP vs. AA	**6:0**	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6***
ZIP won rounds	1000	406	421	369	395	589	0
AA won rounds	0	594	579	631	605	411	1000
ZIP efficiency	99.32%	98.12%	97.28%	95.02%	95.02%	99.5%	-
(s.s.d.)	(0.68%)	(3.34%)	(4.97%)	(6.9%)	(10.1%)	(17.84%)	-
AA efficiency	-	103.18%	100.17%	99.45%	96.75%	88.47%	-
(s.s.d.)	-	(19.08%)	(12.89%)	(10.57%)	(9.78%)	(13.34%)	-
Total efficiency	99.32%	98.97%	98.24%	97.24%	96.17%	90.31%	-
(s.s.d.)	(0.68%)	(1.27%)	(1.99%)	(3.04%)	(4.3%)	(9.04%)	-
Surplus difference	-	7.59	4.34	6.64	2.6	16.54	-
Winner	**ZIP**	**AA**	**AA**	**AA**	**AA**	**ZIP**	**AA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation

Table 6.3: **Two-strategy experiments - GDX vs. AA**

GDX vs. AA	**6:0**	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6***
GDX won rounds	1000	176	366	696	913	976	0
AA won rounds	0	824	634	304	87	24	1000
GDX efficiency	55.31%	94.7%	97.92%	100.48%	104.18%	114.29%	-
(s.s.d.)	(32.6%)	(9.6%)	(3.85%)	(3.72%)	(5.19%)	(8.83%)	-
AA efficiency	-	112.95%	101.15%	94.26%	87.93%	77.38%	-
(s.s.d.)	-	(17.26%)	(8.87%)	(7.78%)	(8.56%)	(10.76%)	-
Total efficiency	55.31%	97.74%	99%	97.37%	93.35%	83.53%	-
(s.s.d.)	(32.6%)	(7.2%)	(1.57%)	(2.56%)	(4.54%)	(7.93%)	-
Surplus difference	-	27.39	4.84	9.33	24.36	55.37	-
Winner	**GDX**	**AA**	**AA**	**GDX**	**GDX**	**GDX**	**AA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation

4 GDX experiments. For the rest of mixed experiments, GDX is a more favourable one and for unbalanced 2 GDX vs. 4 AA and one GDX vs. many AA experiments, GDX wins in an overwhelming number of rounds. The surplus difference shows the clear effect of changing population shares from one strategy to another one. The minority dominates in average profit. The reason for lower total efficiency with increasing share of AA in population is the inability of AA to initiate the first trade of the round decreasing its competitiveness especially in experiments with large AA share in population. More on this matter is in 6.2.2 subsection.

Nevertheless, I have used the same GDX and AA specification as De Luca and Cliff (2011b) and yet GDX wins in the balanced experiment which is in contrast with their results. Since the difference in specification of strategies can be excluded as a reason of this difference, the only logical answer is that the winning strategy depends on the experiment specification such as number of traders, limit prices of each unit and the amount of traded units. This is a very strong conclusion answering some of my hypotheses which I want to test in this work.

6.2.4 ZIPOJA against the other three strategies in two-strategy games

Novel strategy ZIPOJA competed against other strategies bravely but as you will see in following tables, it didn't performed well. In the following table, you can see the most important results for ZIPOJA; ZIPOJA compared to its father strategy ZIP.

Table 6.4: **Two-strategy experiments - ZIP vs. ZIPOJA**

ZIP vs. ZIPOJA	**6:0**	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6**
ZIP won rounds	1000	727	799	774	805	720	0
ZIPOJA won rounds	0	273	201	226	195	280	1000
ZIP efficiency	99.32%	101.28%	103.92%	105.78%	109.97%	110.47%	-
(s.s.d.)	(0.68%)	(4.01%)	(5.73%)	(8.87%)	(12.9%)	(20.3%)	-
ZIPOJA efficiency	-	88.17%	89.09%	91.84%	92.89%	96.18%	98.42%
(s.s.d.)	-	(19.91%)	(11.95%)	(9.14%)	(6.63%)	(4.36%)	(1.26%)
Total efficiency	99.32%	99.1%	98.98%	98.81%	98.58%	98.56%	98.42%
(s.s.d.)	(0.68%)	(0.89%)	(0.98%)	(1.05%)	(1.18%)	(1.19%)	(1.26%)
Surplus difference	-	19.68	22.24	20.91	25.62	21.44	-
Winner	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIPOJA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy.

ZIPOJA with the same setup of parameters as ZIP underperform ZIP in all metrics. It wins fewer rounds no matter the share in population, obtains less profit on average by approximately 20 and as a consequence ZIPOJA is less efficient, too. The overall efficiency in homogenous environment is lower to the one of homogenous ZIP population as well. We can therefore unfortunately conclude that ZIPOJA is worse in all measured metrics than ZIP.

The question is: Why is that so, if ZIP is a special case of ZIPOJA with all weights equal to one and constant. The reason is in the development of weights for Oja's rule. Although no clear pattern was found in development of weights based on time, generally all the weights felt from 1 to a value between 0and 1. This led ZIPOJA to discount the last changes more (effectively forget faster) and take the current difference from target price less into account. For some traders w_2 was closer to 1 and w_1 close to 0 and vice versa. It was not a rule to have one larger than the other. This development of weights led ZIPOJA to stay more stable than ZIP and in case some agent challenged the price or there was some kind of disturbance, ZIPOJA was not able to reap profits as efficiently as ZIP.

The following table shows how ZIPOJA stands against GDX and the table below this one completes the picture of performance of ZIPOJA against other bidding strategies.

Table 6.5: **Two-strategy experiments - GDX vs. ZIPOJA**

GDX vs. ZIPOJA	**6:0**	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6**
GDX won rounds	1000	705	834	864	847	818	0
ZIPOJA won rounds	0	295	166	136	153	182	1000
GDX efficiency	55.31%	101.67%	105.29%	109.16%	111.75%	114.78%	-
(s.s.d.)	(32.6%)	(3.89%)	(5.96%)	(9.14%)	(12.26%)	(18.59%)	-
ZIPOJA efficiency	-	90.11%	88.66%	90.04%	93.29%	95.98%	98.42%
(s.s.d.)	-	(19.18%)	(11.95%)	(9.29%)	(6.18%)	(3.89%)	(1.26%)
Total efficiency	55.31%	99.74%	99.74%	99.6%	99.44%	99.12%	98.42%
(s.s.d.)	(32.6%)	(0.49%)	(0.47%)	(0.62%)	(0.67%)	(0.87%)	(1.26%)
Surplus difference	-	17.35	24.94	28.68	27.7	28.19	-
Winner	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**ZIPOJA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy.

We can see similar results as GDX vs. ZIP although GDX performs even better against ZIPOJA than ZIP. The same story is valid for ZIPOJA vs. AA. There is even the same effect for one ZIPOJA vs. many AA as in one ZIP vs. many AA. The only difference is that ZIPOJA does not make it into winning region of 500 won rounds and more. In other setup, ZIPOJA loses remarkably.

An interesting fact is also how performance of strategies against ZIPOJA in two-strategy games is ordered. For all types of experiments except one (ZIPOJA against many other agents), AA is the best competitor followed by GDX and then ZIP. In case of one vs. many, AA performs the worst, GDX follows and the best competitor is ZIP - completely reversed. The reason is, as mentioned earlier, the changed interaction of strategies because one might

Table 6.6: **Two-strategy experiments - AA vs. ZIPOJA**

AA vs. ZIPOJA	**6:0***	**5:1**	**4:2**	**3:3**	**2:4**	**1:5**	**0:6**
AA won rounds	1000	582	871	921	930	895	0
ZIPOJA won rounds	0	418	129	79	70	105	1000
AA efficiency	-	93.24%	103.45%	109.28%	116.02%	123.03%	-
(s.s.d.)	-	(11%)	(7.21%)	(8.85%)	(12.94%)	(21.1%)	-
ZIPOJA efficiency	-	92.26%	85.12%	86.9%	89.55%	93.81%	98.42%
(s.s.d.)	-	(17.91%)	(10.68%)	(7.97%)	(6.45%)	(4.22%)	(1.26%)
Total efficiency	-	93.07%	97.34%	98.09%	98.37%	98.68%	98.42%
(s.s.d.)	-	(7.26%)	(3.12%)	(2.1%)	(1.66%)	(1.21%)	(1.26%)
Surplus difference	-	1.46	27.49	33.57	39.71	43.83	-
Winner	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**	**ZIPOJA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation

be better prepared for one vs. many experiments than the other strategy. This is for example a case for AA bidding strategy which is unable to initiate first trade in OpEx 1.2 (see 6.2.2 for explanation).

6.3 Three trading strategies competing

The most interesting part of this work (at least in graphical terms) is detailed in this section. As explained in chapter 5, I show which strategy dominates which one and efficiency results. Moreover I show mixed strategy Nash equilibria obtained by the same approach as Vytelingum et at. (2008) and Walsh et al. (2002) did. I use dynamic analysis in order to find the way how strategies converge into Nash equilibria. Then I put these data into understandable ternary graphs with each corner representing one pure strategy.

The motivation is to provide another point of view on comparison of ZIP, GDX and AA strategies since to the best knowledge of author, no one performed experiments with these three strategies in three-strategy game and variable share of these strategies in population. This approach will provide valuable information on how strategies perform in more heterogenous environment which can possibly be dramatically different from two-strategy environment and can possibly challenge the results from the previous research.

Another goal is to use this methodology to introduce ZIPOJA into competition of two other strategies at once, which can speak a completely different story than two-strategy games. We have four strategies to be compared in three-strategy game and therefore four different triplets of strategies:

- ZIP, GDX, AA
- ZIP, GDX, ZIPOJA
- ZIP, ZIPOJA, AA
- ZIPOJA, GDX, AA

I will dedicate one subsection to each of the triplets.

6.3.1 How to read these ternary graphs

All of the information depicted graphically in figures 6.1, 6.3, 6.5, 6.7 and some more information can be found in the same order in tables A.1, A.2, A.3, A.4 in Appendix A. In each corner of all ternary diagrams in this section there is a pure strategy (that means 6 buyers and 6 sellers) and with each further step from the corner (steps are visible in the first picture) there is one buyer and one seller less or in other words the share of this strategy in population mix decreases. For example, in the middle of the ternary graph there is a balanced three strategy game with two buyers and sellers from each of these three strategies. One can check that this point is exactly four steps from each corner resulting in two buyers and sellers for each strategy. This rule is valid for all the points on the graph.

Edges have meaning as well. All the edges represent two-strategy games presented in the previous 6.2 section. From one corner to another setup moves from a pure strategy experiment with six buyers and six sellers in six steps to another pure strategy experiment.

Colours in the first graph in each subsection sign experiments (only in black dots) which were won by a particular strategy. The coloured area and its asymmetry has some minor information too. It is the result of plotting surface of strategy share of won rounds to the whole triangle with values in positions of experiments. Then the highest value for each experiment is the only one visible since it covers all the competitor's surfaces and you are looking on this ternary graph 'from the above'. The coloured area therefore signals the relation with the winning strategy in the neighbouring experiment - large area = much larger percentage of rounds won compared to neighbouring experiment and vice versa.

Colours in the other three ternary graphs in the first figure of each subsection represent share of won games for each strategy with colour bar on the right (0 - no rounds won, 1 - all rounds won). The values are only in experiment nodes so to get smooth surface with colour, values were interpolated into finer grid by Matlab function griddata.

The last type of figure shows found equilibria and dynamic analysis of convergence into these equilibria. Arrows represent s teckou, change in shares of strategies on the total population (see subsection 5.5.3). The three-dimensional change is then transformed into 2D representation to be shown on a paper. Strategies which earn less than average are abandoned in favor of more profitable strategies. Full black dots show pure strategy equilibria in corners and empty circles show mixed strategy Nash equilibria. The colour inside the ternary graph shows the degree of change in any particular point in the space of three strategies. Again, to obtain finer results, function of change for each strategy was interpolated (by Matlab function griddata) and then summed to gain total vector.

6.3.2 Triplet ZIP, GDX, AA

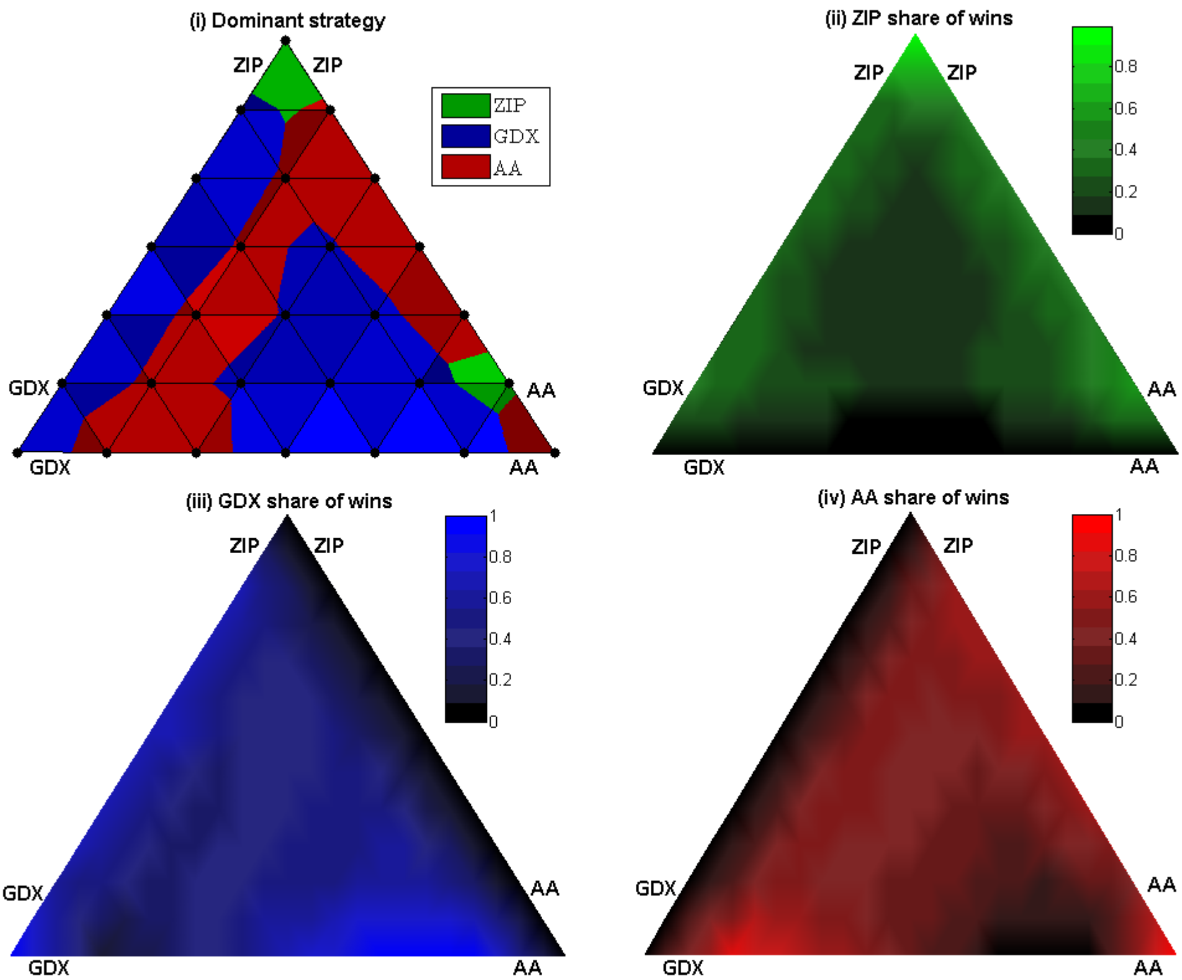
The first triplet to be presented here is ZIP, GDX, AA triplet. As the current state of knowledge says, GDX outperforms ZIP and AA outperforms GDX in two-strategy game. I have shown in subsection 6.2 that the relationship is not that clear especially between AA and GDX even for two-strategy games. This triplet analysis will bring some more information on this topic.

In first picture of figure 6.1 (see data behind and more in table A.1), you can see the overview of dominant strategy for each point representing share of strategies in population mix. Obviously corner experiments belong to pure strategy residing in that corner since there is no competition. The edges, and especially the inner part, are more interesting. We can see that winning position belongs to GDX or AA all the time expect one two-strategy experiment between ZIP and AA explained in 6.2.2 in more detail. The pattern is obvious. AA wins all the experiments where other strategies build the market and there is only one AA buyer and seller to reap above average profits for himself from others. Then it wins experiments only against ZIP on the right edge and one experiment against GDX on the edge (2 AA vs. 4 GDX). The rest of the games (where GDX is present) are ruled by GDX. The winner of balanced game of 2-2-2 buyers and sellers is GDX as well, very narrowly though.

This is a quite interesting result since it shows that the share of strategy on the overall population has a major impact on the result. Moreover, specifications of the experiment have a large impact as well since these results do not fully correspond to the results obtained by the previous research. The finding is not less relevant even with the inability of AA in OpEx 1.2 to open first trade (as explained in 6.2.2 subsection in more detail) since De Luca and Cliff (2011b) use the same specification of agents and get AA dominance as a result of their agent-to-agent experiments. Therefore the difference affecting the result is in experimental setup such as the number of agents, limit orders of units to trade and their number as well.

All the rest of ternary graphs in figure 6.1 represent share of wins in each node by the intensity of colour (ZIP = green, GDX = blue, AA = red). These three graphs show layers of the first graph one by one since in the first graph you see only the highest value for each node (therefore the winning strategy). We can see that ZIP has constant but poor performance across all the mixed three-strategy games, GDX is doing particularly well everywhere especially in experiments with larger share of AA in population and except of experiments with one AA buyer and seller. AA performs badly in three-strategy experiments where it has a large share in population, but well in one AA versus many others type of game. The share of won rounds is steadily increasing with decreasing share in population. However, this effect is mainly done by competition of GDX since it copies the development of two-strategy experiments between GDX and AA. ZIP is the weakest one in this trio and does not affect competition between GDX and AA with its stable results much. GDX and AA fight for victory.

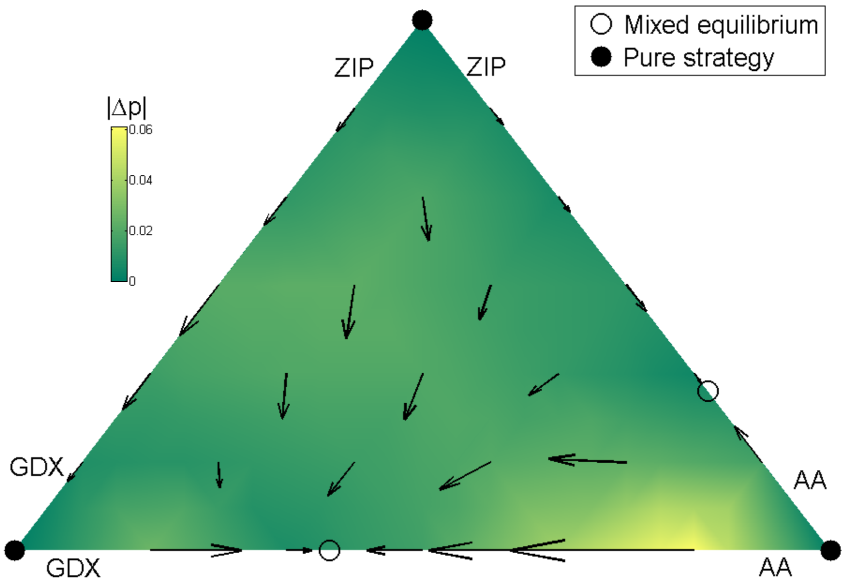
Figure 6.1: 3-Strategy Games Overview - ZIP, GDX, AA



All experiments are run 1000 times. Check tables in in Appendix A to see values.

The next figure shows dynamic analysis based on calculation described in chapter 5. There are 2 mixed strategy Nash equilibria found in this scenario. First one is attracting the whole inner part of triangle and is on GDX-AA edge without ZIP. The share of ZIP, GDX and AA is [0, 61.4%, 38.6%]. That means that ZIP is inferior strategy and optimal shares of strategies are found in this mixed strategy Nash equilibrium. The reason behind this conclusion is that pure strategy equilibria and another equilibrium point have its basin of attraction only on edges or even only in its point (that means that any point close to it directs into different equilibrium point). This is case for ZIP and AA pure strategy. These points do not attract any points except themselves. Pure GDX strategy attracts ZIP-GDX edge mix of strategies. The reason is that since on the edge AA is not introduced, dynamics can move only in two directions - pure ZIP or GDX. Last equilibrium point is mixed strategy Nash equilibrium at ZIP-AA edge and attracting all the points on this edge. The share of ZIP, GDX, AA for this point is [70%, 0, 30%]. We can also see that the fastest change based on colour is on the GDX-AA edge closer to AA pure strategy. This is because GDX outperformed AA very strongly in these experiments.

Figure 6.2: Dynamic analysis for ZIP, GDX, AA



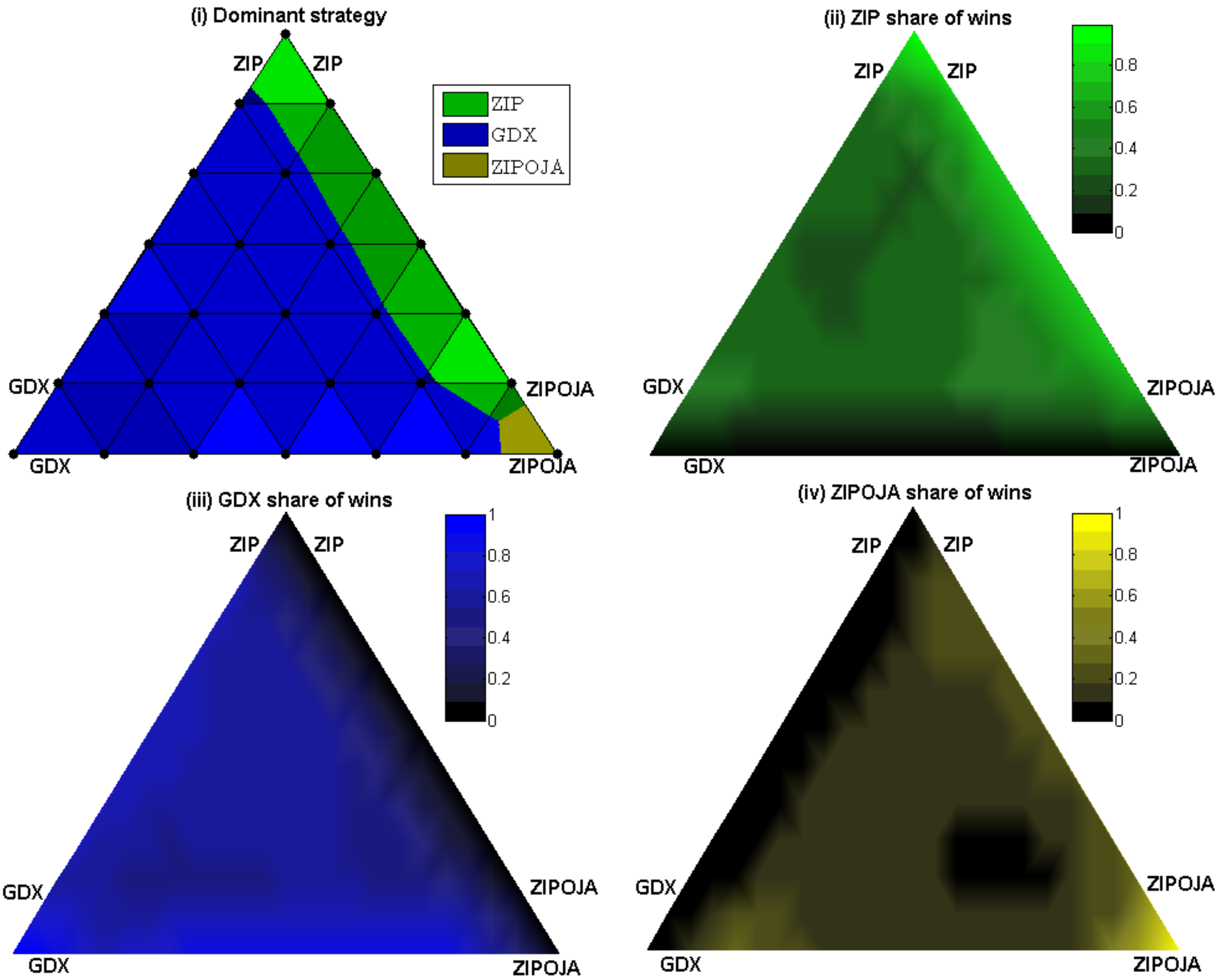
Dynamic analysis for three-strategy experiments with ZIP, GDX, AA.

6.3.3 Ternary ZIP, GDX, ZIPOJA

Next triplet to be put into more detail is ZIP, GDX, ZIPOJA, the first of triplets for introducing ZIPOJA among bidding strategies. In figure 6.3, you can see graphical interpretation of share of won rounds in this set of experiments. After exploring results from two-strategy experiments in section 6.2, one would guess that ZIPOJA will perform badly even in three-strategy experiments. He would be right. ZIPOJA does not win any mixed strategy game. It performs constantly badly but with large share in population it gets even worse. You can see exact numbers for this set of experiments in table A.2. The other two strategies compete more evenly but not evenly enough. GDX outperforms ZIP in all possible scenarios where it is present but the difference is not that sharp.

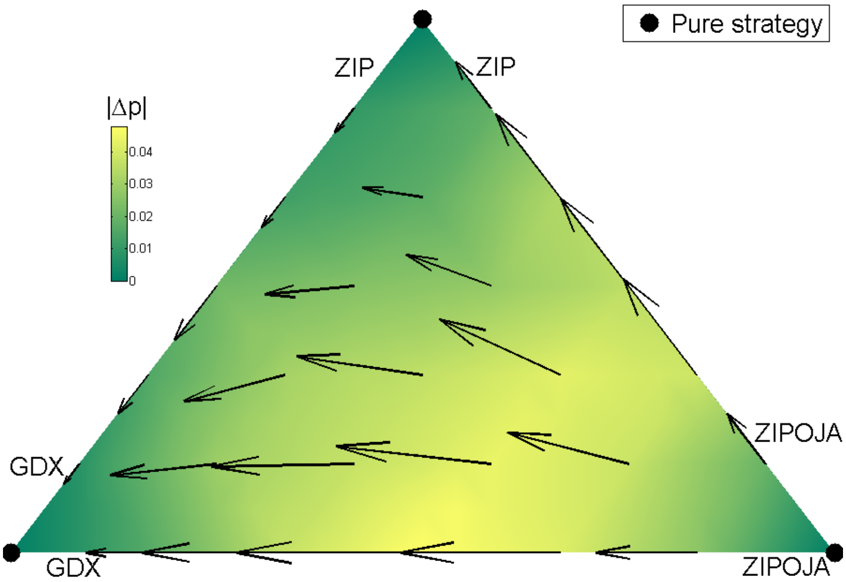
Dynamic analysis constantly moving outward from ZIPOJA clearly shows that under this definition of ZIPOJA, it is not a preferred strategy against ZIP or GDX. There is no mixed stragy Nash equilibrium in this setup and logically there are 3 pure strategy equilibria. ZIPOJA pure strategy equilibrium does not attract any followers except its initial state of pure strategy. ZIP attracts ZIP-ZIPOJA edge since GDX is not introduced there and therefore cannot affect the dynamics by definition. All the remaining points are attracted to pure strategy GDX equilibrium. The more ZIPOJA in the population mix, the faster is the change.

Figure 6.3: 3-Strategy Games Overview - ZIP, GDX, ZIPOJA



All experiments are run 1000 times. Check tables in in Appendix A to see values.

Figure 6.4: Dynamic analysis for ZIP, GDX, ZIPOJA

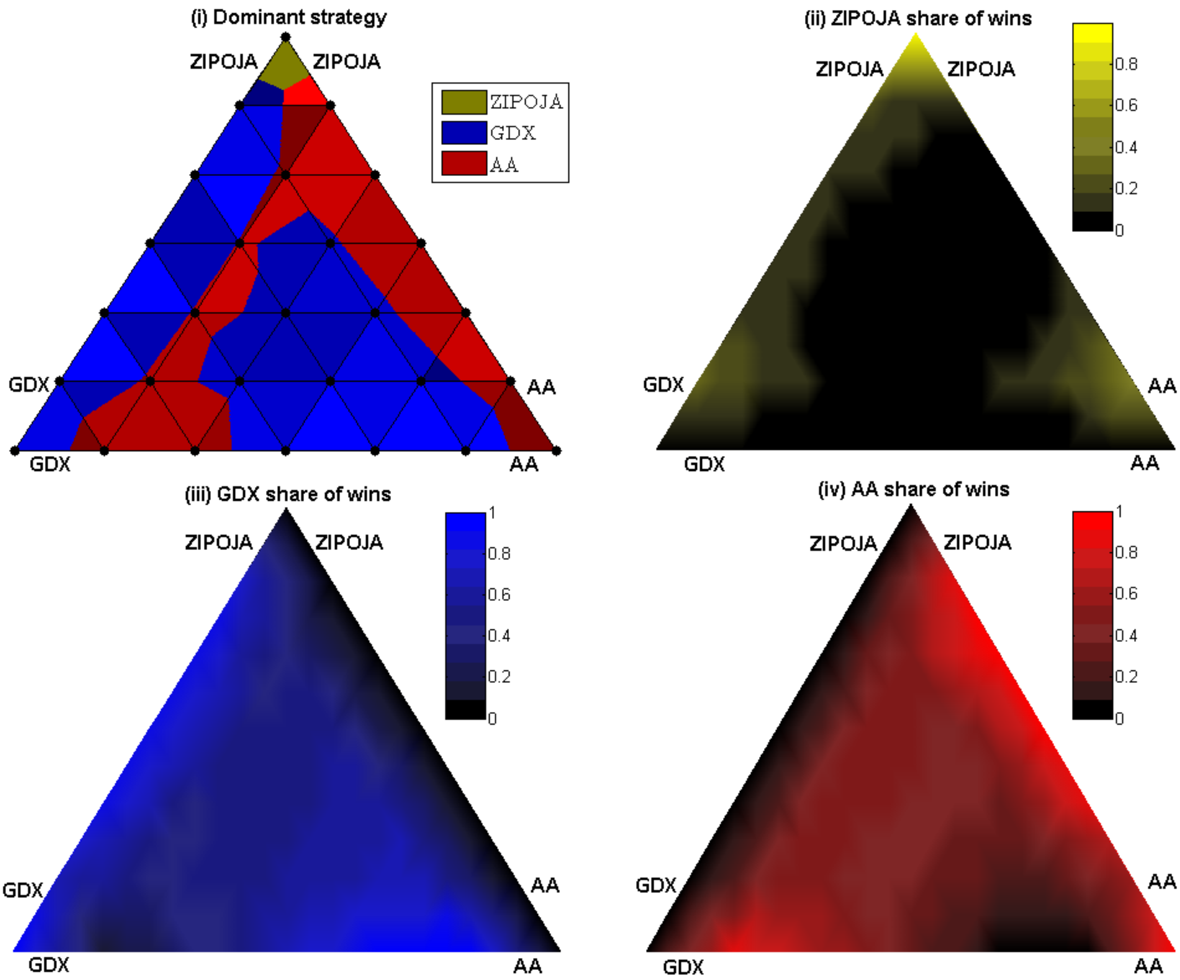


Dynamic analysis for three-strategy experiments with ZIP, GDX ,ZIPOJA.

6.3.4 Triplet ZIPOJA, GDX, AA

Next triplet to describe is ZIPOJA, GDX, AA. Results are visible in 6.5 and more detailed numbers are in A.4. GDX and AA are the most successful strategies in results presented so far. Therefore, ZIPOJA as a weak strategy shows strong underperformance in this scenario. It did not manage to succeed notably in any three-strategy experiment. ZIPOJA performs better in two-strategy games (see results in 6.2 section). Since ZIPOJA does not affect the results much, the resulting pattern of dominant strategy is very similar to triplet ZIP, GDX, AA. As presented in 6.3.2, AA wins in all settings where it is one AA buyer and seller against many others, everywhere on ZIP-AA edge, and in one 2 AA vs. 4 GDX case as well. The rest of setups is dominated by GDX. This (balanced 3-3 experiment), as mentioned in 6.3.2, is in contrast to previous research results) and since De Luca and Cliff (2011b) use the same agent specification, the reason for the difference in results has to be the experimental setup such as the number of agents, limit prices of units traded and their amount. The difference is dependent on strategy share on population as well, which was proven by results in previous sections of this chapter. The dominance of AA over GDX is therefore not so clear as it is presented in the literature. The contra-argument of different specification of AA in OpEx 1.2 (as elaborated more in subsection 6.2.2) might be of high importance but comparison with De Luca and Cliff (2011b) results show that the question of whether setting of experiment and the share of strategy in population of traders really matters as well.
Dynamic analysis of this set of experiments shows already known mixed strategy Nash equilibrium at [61.4%, 38.6%] GDX and AA shares in trader population. It attracts all of the inner part of triangle and all the points on GDX-AA edge. Pure strategy AA attracts ZIPOJA-AA edge and pure strategy GDX attracts ZIPOJA-GDX edge. The only point not attracted to any of these points is pure strategy ZIPOJA which stays in its own equilibrium. The movement is

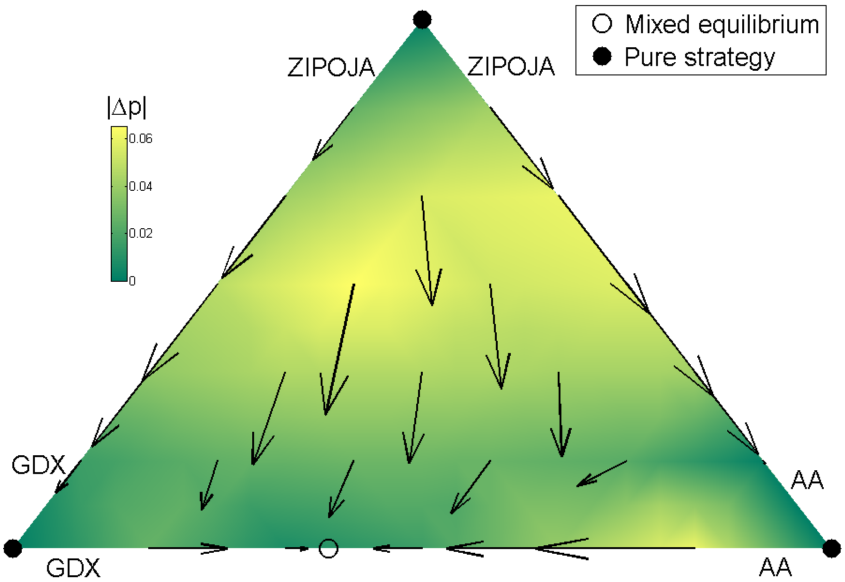
Figure 6.5: 3-Strategy Games Overview - ZIPOJA, GDX, AA



All experiments are run 1000 times. Check tables in in Appendix A to see values.

faster with a larger share of ZIPOJA population since traders have larger motivation to leave the ZIPOJA and obtain above average profit with other strategies.

Figure 6.6: Dynamic analysis for ZIPOJA, GDX, AA



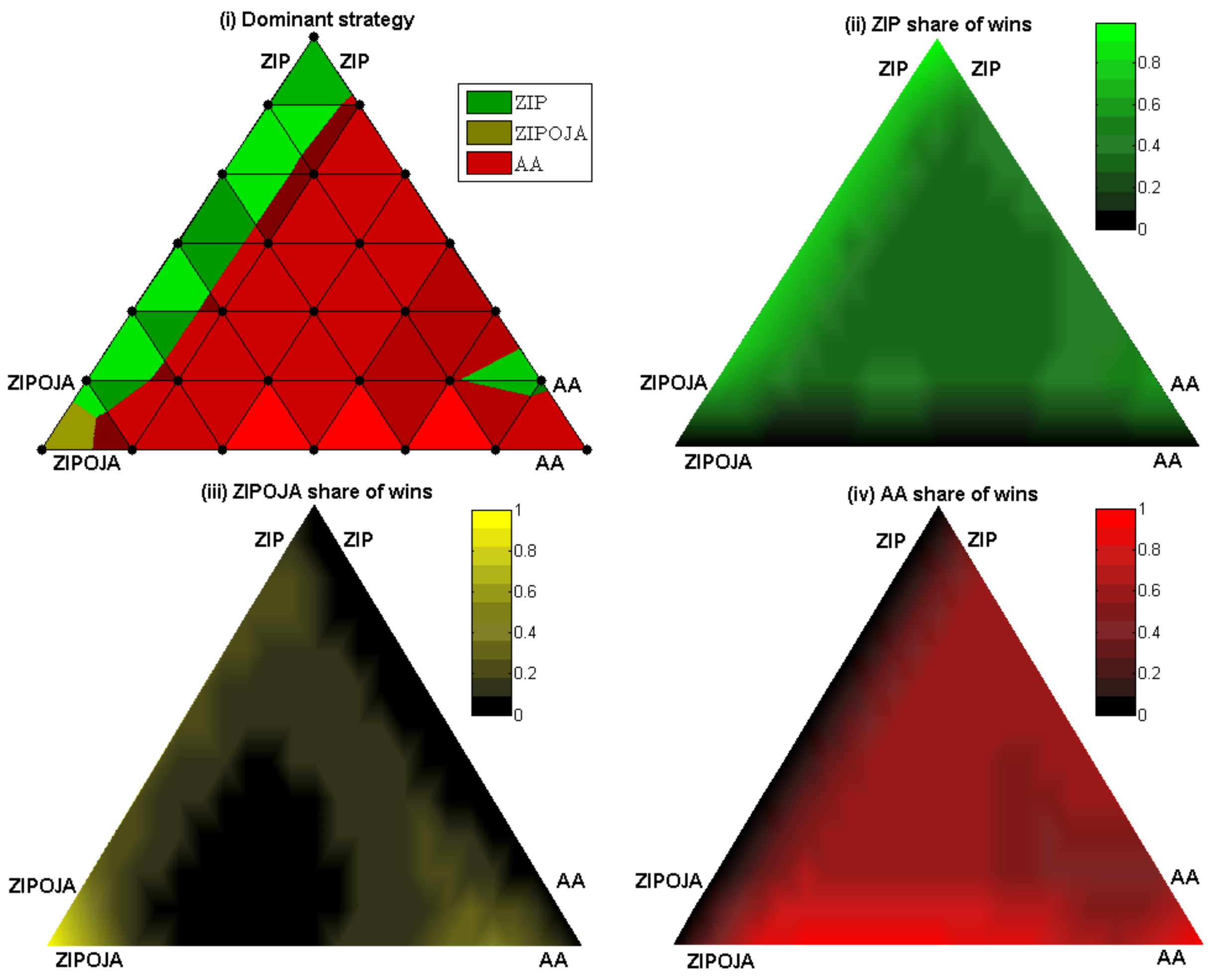
Dynamic analysis for three-strategy experiments with ZIPOJA, GDX, AA.

6.3.5 Triplet ZIP, ZIPOJA, AA

The last of triplets is ZIP, ZIPOJA, AA triplet. The effect of introducing ZIPOJA among ZIP and AA is the same as among GDX and AA. ZIPOJA underperforms and ZIP and AA has quite stable results with ZIP losing in all experiments but one which is elaborated on in subsection 6.2.2 since it falls into two-strategy game. We can see that AA clearly dominates ZIP and ZIPOJA as well. ZIP dominates ZIPOJA as well.

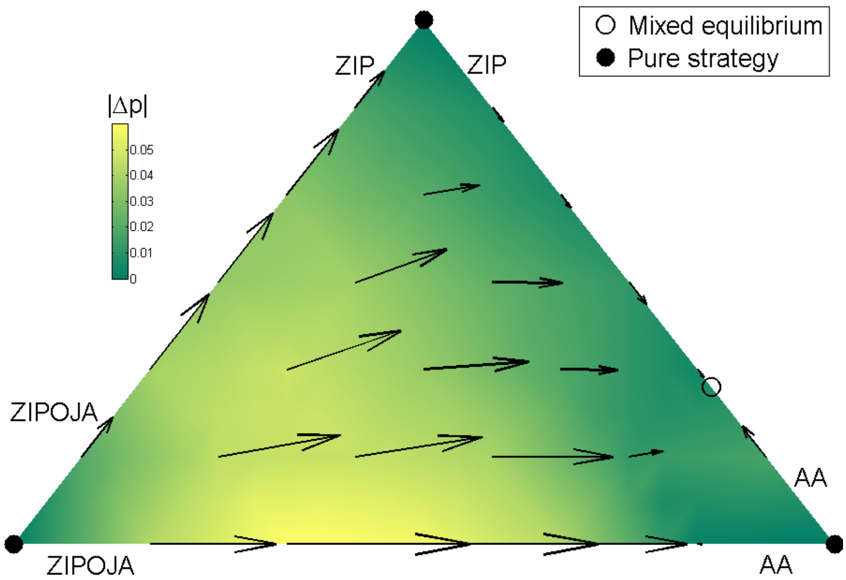
Dynamic analysis tells the already known story. There is the same equilibrium point on ZIP-AA edge at [70%, 30%] of ZIP, AA mix as was found in ZIP, GDX, AA triplet analysis. The dynamics move out from ZIPOJA. All the inner points end up in the mentioned mixed strategy Nash equilibrium point. Pure strategy AA attracts ZIPOJA-AA edge and pure strategy ZIP attracts ZIPOJA-ZIP edge. ZIPOJA pure strategy is equilibrium point with no basin of attraction.

Figure 6.7: 3-Strategy Games Overview - ZIP, ZIPOJA, AA



All experiments are run 1000 times. Check tables in in Appendix A to see values.

Figure 6.8: Dynamic analysis for ZIP, ZIPOJA, AA



Dynamic analysis for three-strategy experiments with ZIP, ZIPOJA, AA.

6.4 Four trading strategies competing

To complete the overview of four strategies ZIP, GDX, AA, and ZIPOJA competing, four-strategy balanced experiment will be run. As explained before, it is not possible to show results of four-strategy experiments with changing share of strategies on population mix graphically and therefore I will just stick to the balanced experiment with 1000 rounds as well. We included 2 buyers and 2 sellers from all 4 strategies so 16 traders competed overall.

Table 6.7: **Four-strategy experiment**

	ZIP	**GDX**	**AA**	**ZIPOJA**	**Market**
Won rounds	228	367	360	45	1000
Efficiency	99.55%	106.19%	105.20%	85.01%	98.99%
(s.s.d.)	(11.40%)	(11.31%)	(11.21%)	(11.99%)	(1.12%)
Surplus	0.84	10.80	9.33	-20.97	0
Winner					**GDX**

Results of 1000 rounds of balanced four-strategy experiment. Surplus is the average of all round surpluses.

The results as seen in previous table confirm the underperformance of ZIPOJA followed by not that low underperformance of ZIP. However, GDX and AA got almost tie with GDX only slightly winning this experiment. This is in line with results obtained in previous sections. GDX and AA are in many cases comparable in dominance and the winner depends on the experimental setup such as the number of agents, number of agents following one strategy, limit price of units to be traded and their amount. As a result it can be said that there is no clear order for GDX and AA, but generally GDX and AA outperforms ZIP almost all the time and all three outperforms ZIPOJA all the time.

Chapter 7

Conclusion

In this thesis, I have covered previous research on automated bidding strategies ZIP, GDX and AA. I have developed novel automated bidding strategy called ZIPOJA, based on ZIP with an extension for Oja's rule (Oja, 1982), which is believed to be a computational form of the learning process in biological neurons and it is used in artificial neural networks. I have run altogether 74 computationally intensive experiments in the trading simulator OpEx 1.2. Experiments varied share of strategies in population of traders but kept the symmetry of game in the sense that one strategy had as many buyers as sellers no matter what the share on population was. This approach provided me with a tool to compare already known strategies ZIP, GDX, and AA from a different perspective, effectively adding a value to the research community since the same approach was applied only to a different set of strategies in Walsh et al. (2002), whose approach simplifies overly complex game with multiple agents into analyzable game-theoretic matrix of payoffs. Thanks to this approach, I could introduce ZIPOJA into the competition of ZIP, GDX, and AA automated bidding strategies.

The performed experiments could be divided into four categories by the number of strategies included in the population of traders. There was a homogenous environment with all traders having the same strategy, two-strategy games, three-strategy games and four-strategy games. Two and three-strategy games were of the most added value since the share of strategies in total population was varied, which gave us an interesting view on how results change based on the share of strategies in the total population. Three-strategy games were of the most importance, since their properties allowed to show results on paper which would not be possible for four-strategy games due to the limitation of two dimensions. This reason together with even more extensive computational intensity led to focusing on three-strategy game rather than four-strategy game which was performed only in the balanced setup to complete the overview.

The main findings of this work are that AA does not outperform other strategies (especially GDX) in all experimental setup, which is in contrast with results of previous literature. GDX is the dominant strategy in many experimental setups. The newly introduced ZIPOJA underperforms other strategies, including original ZIP. As a result of my experiments, I am able to elaborate on my hypotheses:

Hypothesis 1: AA bidding strategy does not outperform the other bidding strategies in different experimental setups as significantly as in De Luca and Cliff (2011b).

As shown in comparison of AA against GDX in two-strategy games, the dominance of AA changes significantly based on the experimental setup, such as limit prices for units to be traded and the number of them, as well as on the number of agents included in the experiment. In my balanced (3 buyers and 3 sellers from each strategy) experiments of GDX vs. AA, GDX outperformed AA. This testifies about the importance of experimental setup since De Luca and Cliff (2011b) use the same strategy specifications as I do, but AA outperforms GDX in their experimental setup. Therefore, conclusion can be drawn that experimental setup parameters such as the number of agents, limit prices of units to be traded and their amount affects the outcome significantly.

Hypothesis 2: There is a difference in how bidding strategies perform based on the number of rival strategies included in experiment.

The experiments showed that the results vary between scenarios and that strategies interact with each other. The answer based on these results would be that it matters more which strategies are competing than how many of them are present. Nonetheless, this factor has an effect as well.

Hypothesis 3: There is a difference in how trade strategies perform based on the number of agents following each rule.

This is probably the most important hypothesis to be answered. The results are truly different based on the share of strategy in the total population. Strategies AA and GDX were competing for dominance most evenly. It was shown that AA dominates other strategies in experiments in which AA has only one buyer and seller and the rest of the market (5 buyers and 5 sellers) is ruled by different strategies, and in one case of two-strategy experiment of GDX vs. AA where AA has two buyers and sellers. GDX dominates all the rest of experiments in which it is present. Moreover, ZIP wins one two-strategy game between ZIP and AA but loses all the others. Based on this evidence, conclusion can be drawn that it really significantly matters what the share of strategy on the total population is.

Hypothesis 4: It is possible to rank these algorithmic strategies with sufficient significance and this order is robust to changes in the experimental setup.

The answer to this hypothesis remains still ambiguous to some extent. As mentioned earlier, results depend significantly on the experimental setup. Therefore I can answer this question only regarding experiments performed in this work. Moreover, there were some mixed Nash equilibria found which suggest that the optimum might be to have a mixed strategy. These equilibria can be a result of dynamic adaptation even from a different initial stage as shown by the dynamic analysis.

In all my experiments, ZIPOJA was not a preferred strategy so it is ranked fourth amongst bidding strategies. ZIP lost all the games against GDX and AA except one against AA. We might therefore say that ZIP is ranked third. However, there is not a clear winner since GDX and AA dominate others in different setups. The answer to this hypothesis is that it is possible to rank these automated bidding strategies and the rank might be without changes for some alterations in experimental setup but it is not robust to all setup alterations. Not only the share in population affects the results, but the experimental setup affects the results too, as explained in the answer to the second hypothesis.

It has to be mentioned that divergence of AA in OpEx 1.2 compared to theoretical AA was found and it might affect the absolute results. Nonetheless, it does not affect the findings about the importance of the experimental setup and a changed share of strategies on the total population.

One direction of future work might lead to testing these strategies on a real-world data which was originally intended to be done in this work but turned out to be over the scope of this work. Another possible extension is to rethink the specification of ZIPOJA learning rule or to optimize the parameters which might significantly improve the performance of this strategy, but it was not included here to keep ZIPOJA comparable with the original ZIP as defined in Cliff (1997). The last but definitely not the least relevant way to create a more robust information set on this topic is to test the specification of agents in OpEx 1.2 as this program provides one of a few publicly available versions of AA bidding strategy.

Bibliography

Bienenstock, E.L., Cooper, L. and Munro, P., 1982. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience 2 (1): 32-48.*

Cartlidge, J. and Cliff, D., 2012. Exploring the "robot phase transition" in experimental human-algorithmic markets. Foresight, The Future of Computer Trading in Financial Markets, Driver Review DR25.

Cartlidge, J., De Luca, M., Szostek, C. and Cliff, D., 2012. Too fast too furious: faster financial-market trading agents can give less efficient markets. In: J. Filipe and A.L.N. Fred, eds., *4th Int. Conf. on Agents & Artificial Intelligence, Vol. 2 - Agents (ICAART-2012).* Vilamoura, Algarve, Portugal: SciTePress, 126–135.

Cliff, D., 1997. Minimal-Intelligence Agents for Bargaining Behaviors in Market-Based Environments. *Technical Report HPL-97-91, Hewlett-Packard Laboratories.*

Cliff, D., Bruten, J. and Road, F., 1997. Zero is Not Enough: On The Lower Limit of Agent Intelligence for Continuous Double Auction Markets.

Das, R., Hanson, J., Kephart, J. and Tesauro, G., 2001. Agent-human interactions in the continuous double auction. *In: Nebel, B. (ed.) 17th Int. JointConf. Artif. Intell. (IJCAI-01)*, pp. 1169–1176.

De Luca, M. and Cliff, D., 2011a. Agent-human interactions in the continuous double auction, redux: Using OpEx lab-in-a-box to explore ZIP and GDX. *In: Filipe, J., Fred, A. (eds.) 3rd Int. Conf. Agents & Artif. Intell. (ICAART-2011).*, pp. 351–358.

De Luca, M. and Cliff, D., 2011b. Human-Agent Auction Interactions: Adaptive-Aggressive Agents Dominate. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, Barcelona, Spain, July 2011, pp. 178–185.

De Luca, M., Szostek, C., Cartlidge, J. and Cliff, D., 2011. Studies of Interactions Between Human Traders and Algorithmic Trading Systems. Foresight, The Future of Computer Trading in Financial Markets, Driver Review DR13.

Fudenberg, D. and Kresp, D.M., 1993. Learning mixed equilibria. *Games and Economic Behavior 5*, pp. 320–367.

Fudenberg, D. and Levine, K.K., 1993. Steady state learning and Nash equilibrium. *Econometrica 61(3):320-367.*

Gjerstad, S. and Dickhaut, J., 1998. Price Formation in Double AAuction. *Games and Economic Behavior*, 22(1), pp. 1–29.

Gode, S. and Sunder, S., 1993. Allocative Efficiency of Markets with Zero-Inteligence Traders: Markets as a Partial Substitute for Individual Rationality. *Journal of Political Economy*, 101(1), pp. 119–137.

Jordan, J.S., 1993. Three problems in learning mixed strategy Nash equilibria. *Games and Economic Behavior 5: 368-386.*

Lorenz, J. and Osterrieder, J., 2009. Simulation of a Limit Order Driven Market. *The Journal of Trading*, Vol. 4, No. 1: pp. 23-30.

McKelvey, R.D. and McLennan, A., 1996. Computation of equiliria in finite games. *Handbook of Computational Economics, 1.*

Nelder, J.A. and Mead, R., 1965. A simplex method for function minimaization. *Computer Journal*, 7(4), pp. 308–313.

Oja, E., 1982. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15, 267–273.

OpEx, 2012. SourceForge.net open-source download homepage for Open Exchange (OpEx) project. *http://sourceforge.net/p/open-exchange/home/Home/. Last accessed 29th July., 2015.*

Preist, C. and van Tol, M., 1998. Adaptive agents in a persistent shout double auction. *In: 1st Int. Conf. Inform. & Computation Economies.*, pp. 11–18. ACM Press.

Press, W., Teukolsky, S., Vetterling, W. and Flannery, B., 1992. Numerical recipes in C. *Cambridge University Press.*

Sanger, T.D., 1989. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks 2 (6): 459-473.*

Smith, V., 1962. An experimental study of comparative market behavior. *The Journal of Political Economy*, 70, pp. 111–137.

Stotter, S., Cartlidge, J. and Cliff, D., 2014. Behavioral investigations of financial trading agents using Exchange Portal (ExPo). *In: N. T. Nguyen, R. Kowalczyk, A. Fred, and F. Joaquim eds. Transactions on Computational Collective Intelligence XVII.*, Berlin Heidelberg: Springer, pp. 22–45.

Tesauro, G. and Bredin, J.L., 2002. Strategic sequential bidding in auctions using dynamic programming. *Proc. First International Joint Conference on Autonomous Agents and MultiAgent Systems*, pp. 591–598.

Tesauro, G. and Das, R., 2001. High-Performance Bidding Agents for the Continuous Double Auction. *In Proceedings of IJCAI 2001 Economic Agents Models and Mechanisms Workshop and/or Proceedings of the Third ACM Conference on Electronic Commerce*, pp. 206–209.

Toft, I., 2007. *Adaptive Bidding Agents for Single Seller and Continuous Double Auctions.* Ph.D. thesis, Department of Computer Science, University of East Anglia, UK.

Vytelingum, P., 2006. *The Structure and Behaviour of the Continuous Double Auction.* Ph.D. thesis, University of Southampton School of Electronics and Computer Science, Southampton, UK.

Vytelingum, P., Cliff, D. and Jennings, N., 2008. Strategic bidding in continuous double auctions. *Artificial Intelligence.*

Vytelingum, P., Dash, R.K., David, E. and Jennings, N.R., 2004. A risk-based bidding strategy for continuous double auctions. *In 16th European Conference on Artificial Intelligence*, pp. 79–83.

Walsh, W.E., Das, R., Tesauro, G. and Kephart, J.O., 2002. Analyzing complex strategic interactions in multi-agent games. *Proc. AAAI Workshop on Game-Theoretic and Decision-Theoretic Agents.*

Weibull, J.W., 1995. Evolutionary Game Theory. *MIT Press, Cambridge, MA.*

Widrow, B. and Hoff, Jr., M.E., 1960. Adaptive switching circuits. *Inst. Radio Engineers, Western Electron. Show and Conv. (IRE WESCON), Conv. Rec.*, 4, pp. 96–104.

Appendix A

Supplementary tables

You can see tables for three-strategy experiments in this appendix. Tables are too large but beautiful. See next page.

Table A.1: **Three-strategy experiments - ZIP vs. GDX vs. AA**

ZIP vs. GDX vs. AA	**6:0:0**	**5:1:0**	**4:2:0**	**3:3:0**	**2:4:0**	**1:5:0**	**0:6:0**
ZIP won rounds	1000	344	353	273	342	382	0
GDX won rounds	0	656	647	727	658	618	1000
AA won rounds	0	0	0	0	0	0	0
ZIP efficiency	99.32%	98.35%	97.85%	95.43%	95.18%	94.81%	-
(s.s.d.)	(0.68%)	(2.74%)	(4.39%)	(6.98%)	(10.44%)	(18.56%)	-
GDX efficiency	-	105.32%	103.29%	104%	102.07%	100.63%	55.31%
(s.s.d.)	-	(13.15%)	(8.61%)	(6.94%)	(5.28%)	(3.94%)	(32.6%)
AA efficiency	-	-	-	-	-	-	-
(s.s.d.)	-	-	-	-	-	-	-
Total efficiency	99.32%	99.51%	99.66%	99.72%	99.77%	99.66%	55.31%
(s.s.d.)	(0.68%)	(0.59%)	(0.51%)	(0.47%)	(0.49%)	(0.66%)	(32.6%)
Winner	**ZIP**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**

ZIP vs. GDX vs. AA	**5:0:1**	**4:1:1**	**3:2:1**	**2:3:1**	**1:4:1**	**0:5:1**	**5:0:2**
ZIP won rounds	406	130	123	148	205	0	421
GDX won rounds	0	375	398	322	314	176	0
AA won rounds	594	495	479	530	481	824	579
ZIP efficiency	98.12%	96.36%	94.89%	93.69%	93%	-	97.28%
(s.s.d.)	(3.34%)	(4.96%)	(6.57%)	(9.06%)	(14.79%)	-	(4.97%)
GDX efficiency	-	103.81%	103.55%	101.65%	100.31%	94.7%	-
(s.s.d.)	-	(12.28%)	(8.94%)	(6.37%)	(4.9%)	(9.6%)	-
AA efficiency	103.18%	105.85%	104.93%	104.6%	103.01%	112.95%	100.17%
(s.s.d.)	(19.08%)	(18.77%)	(16.7%)	(16.14%)	(16.26%)	(17.26%)	(12.89%)
Total efficiency	98.97%	99.18%	99.45%	99.49%	99.54%	97.74%	98.24%
(s.s.d.)	(1.27%)	(1.23%)	(0.88%)	(0.94%)	(0.95%)	(7.2%)	(1.99%)
Winner	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**

ZIP vs. GDX vs. AA	**3:1:2**	**2:2:2**	**1:3:2**	**0:4:2**	**3:0:3**	**2:1:3**	**1:2:3**
ZIP won rounds	156	144	176	0	369	191	172
GDX won rounds	457	437	450	366	0	534	541
AA won rounds	387	419	374	634	631	275	287
ZIP efficiency	95.57%	92.89%	90.28%	-	95.02%	94.46%	90.47%
(s.s.d.)	(6.39%)	(9.18%)	(13.77%)	-	(6.9%)	(9.38%)	(13.68%)
GDX efficiency	104.75%	103.21%	101.77%	97.92%	-	104.87%	103.46%
(s.s.d.)	(13.52%)	(8.44%)	(6.08%)	(3.85%)	-	(12.06%)	(7.86%)
AA efficiency	99.76%	100.25%	98.81%	101.15%	99.45%	96.47%	95.87%
(s.s.d.)	(12.44%)	(11.95%)	(11.25%)	(8.87%)	(10.57%)	(10.51%)	(9.79%)
Total efficiency	98.5%	98.78%	98.87%	99%	97.24%	97.2%	97.5%
(s.s.d.)	(1.98%)	(1.78%)	(1.69%)	(1.57%)	(3.04%)	(3.06%)	(2.76%)
Winner	**GDX**	**GDX**	**GDX**	**AA**	**AA**	**GDX**	**GDX**

ZIP vs. GDX vs. AA	**0:3:3**	**2:0:4**	**1:1:4**	**0:2:4**	**1:0:5**	**0:1:5**	**0:0:6***
ZIP won rounds	0	395	216	0	589	0	0
GDX won rounds	696	0	620	913	0	976	0
AA won rounds	304	605	164	87	411	24	1000
ZIP efficiency	-	95.02%	95.34%	-	99.5%	0%	-
(s.s.d.)	-	(10.1%)	(14.71%)	-	(17.84%)	-	-
GDX efficiency	100.48%	-	107.06%	104.18%	-	114.29%	-
(s.s.d.)	(3.72%)	-	(11.42%)	(5.19%)	-	(8.83%)	-
AA efficiency	94.26%	96.75%	91.19%	87.93%	88.47%	77.38%	-
(s.s.d.)	(7.78%)	(9.78%)	(11.17%)	(8.56%)	(13.34%)	(10.76%)	-
Total efficiency	97.37%	96.17%	94.53%	93.35%	90.31%	83.53%	-
(s.s.d.)	(2.56%)	(4.3%)	(5.26%)	(4.54%)	(9.04%)	(7.93%)	-
Winner	**GDX**	**AA**	**GDX**	**GDX**	**ZIP**	**GDX**	**AA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation

Table A.2: **Three-strategy experiments - ZIP vs. GDX vs. ZIPOJA**

ZIP vs. GDX vs. ZIPOJA	**6:0:0**	**5:1:0**	**4:2:0**	**3:3:0**	**2:4:0**	**1:5:0**	**0:6:0**
ZIP won rounds	1000	344	353	273	342	382	0
GDX won rounds	0	656	647	727	658	618	1000
ZIPOJA won rounds	0	0	0	0	0	0	0
ZIP efficiency	99.32%	98.35%	97.85%	95.43%	95.18%	94.81%	-
(s.s.d.)	(0.68%)	(2.74%)	(4.39%)	(6.98%)	(10.44%)	(18.56%)	-
GDX efficiency	-	105.32%	103.29%	104%	102.07%	100.63%	55.31%
(s.s.d.)	-	(13.15%)	(8.61%)	(6.94%)	(5.28%)	(3.94%)	(32.6%)
ZIPOJA efficiency	-	-	-	-	-	-	-
(s.s.d.)	-	-	-	-	-	-	-
Total efficiency	99.32%	99.51%	99.66%	99.72%	99.77%	99.66%	55.31%
(s.s.d.)	(0.68%)	(0.59%)	(0.51%)	(0.47%)	(0.49%)	(0.66%)	(32.6%)
Winner	**ZIP**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**

ZIP vs. GDX vs. ZIPOJA	**5:0:1**	**4:1:1**	**3:2:1**	**2:3:1**	**1:4:1**	**0:5:1**	**5:0:2**
ZIP won rounds	727	265	268	248	350	0	799
GDX won rounds	0	556	560	577	498	705	0
ZIPOJA won rounds	273	179	172	175	152	295	201
ZIP efficiency	101.28%	99.93%	98.93%	96.56%	96.91%	-	103.92%
(s.s.d.)	(4.01%)	(5.04%)	(7.89%)	(10.99%)	(17.32%)	-	(5.73%)
GDX efficiency	-	107.64%	107.01%	105.97%	104.21%	101.67%	-
(s.s.d.)	-	(13.87%)	(10.63%)	(7.39%)	(5.21%)	(3.89%)	-
ZIPOJA efficiency	88.17%	89.05%	86.81%	87.28%	84.77%	90.11%	89.09%
(s.s.d.)	(19.91%)	(18.31%)	(18.86%)	(18.28%)	(18.5%)	(19.18%)	(11.95%)
Total efficiency	99.1%	99.4%	99.6%	99.72%	99.75%	99.74%	98.98%
(s.s.d.)	(0.89%)	(0.69%)	(0.61%)	(0.46%)	(0.45%)	(0.49%)	(0.98%)
Winner	**ZIP**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**ZIP**

ZIP vs. GDX vs. ZIPOJA	**3:1:2**	**2:2:2**	**1:3:2**	**0:4:2**	**3:0:3**	**2:1:3**	**1:2:3**
ZIP won rounds	318	320	320	0	774	387	365
GDX won rounds	553	586	541	834	0	522	546
ZIPOJA won rounds	129	94	139	166	226	91	89
ZIP efficiency	101.74%	101.69%	98.95%	-	105.78%	105.88%	103.62%
(s.s.d.)	(7.58%)	(10.52%)	(17.47%)	-	(8.87%)	(12.05%)	(18.84%)
GDX efficiency	108.65%	108.16%	107.1%	105.29%	-	111.25%	110.36%
(s.s.d.)	(15.23%)	(10.04%)	(7.82%)	(5.96%)	-	(16.73%)	(11.32%)
ZIPOJA efficiency	90.86%	88.72%	88.86%	88.66%	91.84%	90.85%	90.79%
(s.s.d.)	(11.89%)	(11.95%)	(12.57%)	(11.95%)	(9.14%)	(9.37%)	(9.12%)
Total efficiency	99.27%	99.52%	99.66%	99.74%	98.81%	99.26%	99.45%
(s.s.d.)	(0.79%)	(0.63%)	(0.54%)	(0.47%)	(1.05%)	(0.78%)	(0.64%)
Winner	**GDX**	**GDX**	**GDX**	**GDX**	**ZIP**	**GDX**	**GDX**

ZIP vs. GDX vs. ZIPOJA	**0:3:3**	**2:0:4**	**1:1:4**	**0:2:4**	**1:0:5**	**0:1:5**	**0:0:6**
ZIP won rounds	0	805	371	0	720	0	0
GDX won rounds	864	0	540	847	0	818	0
ZIPOJA won rounds	136	195	89	153	280	182	1000
ZIP efficiency	-	109.97%	106.57%	-	110.47%	-	-
(s.s.d.)	-	(12.9%)	(19.75%)	-	(20.3%)	-	-
GDX efficiency	109.16%	-	114.14%	111.75%	-	114.78%	-
(s.s.d.)	(9.14%)	-	(18.72%)	(12.26%)	-	(18.59%)	-
ZIPOJA efficiency	90.04%	92.89%	93.52%	93.29%	96.18%	95.98%	98.42%
(s.s.d.)	(9.29%)	(6.63%)	(6.55%)	(6.18%)	(4.36%)	(3.89%)	(1.26%)
Total efficiency	99.6%	98.58%	99.13%	99.44%	98.56%	99.12%	98.42%
(s.s.d.)	(0.62%)	(1.18%)	(0.86%)	(0.67%)	(1.19%)	(0.87%)	(1.26%)
Winner	**GDX**	**ZIP**	**GDX**	**GDX**	**ZIP**	**GDX**	**ZIPOJA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy.

Table A.3: **Three-strategy experiments - ZIP vs. ZIPOJA vs. AA**

ZIP vs. ZIPOJA vs. AA	**6:0:0**	**5:1:0**	**4:2:0**	**3:3:0**	**2:4:0**	**1:5:0**	**0:6:0**
ZIP won rounds	1000	727	799	774	805	720	0
ZIPOJA won rounds	0	273	201	226	195	280	1000
AA won rounds	0	0	0	0	0	0	0
ZIP efficiency	99.32%	101.28%	103.92%	105.78%	109.97%	110.47%	-
(s.s.d.)	(0.68%)	(4.01%)	(5.73%)	(8.87%)	(12.9%)	(20.3%)	-
ZIPOJA efficiency	-	88.17%	89.09%	91.84%	92.89%	96.18%	98.42%
(s.s.d.)	-	(19.91%)	(11.95%)	(9.14%)	(6.63%)	(4.36%)	(1.26%)
AA efficiency	-	-	-	-	-	-	-
(s.s.d.)	-	-	-	-	-	-	-
Total efficiency	99.32%	99.1%	98.98%	98.81%	98.58%	98.56%	98.42%
(s.s.d.)	(0.68%)	(0.89%)	(0.98%)	(1.05%)	(1.18%)	(1.19%)	(1.26%)
Winner	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIP**	**ZIPOJA**

ZIP vs. ZIPOJA vs. AA	**5:0:1**	**4:1:1**	**3:2:1**	**2:3:1**	**1:4:1**	**0:5:1**	**5:0:2**
ZIP won rounds	406	294	303	317	307	0	421
ZIPOJA won rounds	0	134	98	64	58	105	0
AA won rounds	594	572	599	619	635	895	579
ZIP efficiency	98.12%	99.57%	102.02%	104.58%	105.12%	-	97.28%
(s.s.d.)	(3.34%)	(5.71%)	(8.27%)	(12.34%)	(19.39%)	-	(4.97%)
ZIPOJA efficiency	-	86.63%	87.43%	89.42%	91.94%	93.81%	-
(s.s.d.)	-	(18.86%)	(12.78%)	(9.49%)	(6.59%)	(4.22%)	-
AA efficiency	103.18%	108.12%	111.43%	114.82%	118.86%	123.03%	100.17%
(s.s.d.)	(19.08%)	(18.84%)	(18.95%)	(20.44%)	(19.83%)	(21.1%)	(12.89%)
Total efficiency	98.97%	98.84%	98.73%	98.71%	98.62%	98.68%	98.24%
(s.s.d.)	(1.27%)	(1.27%)	(1.23%)	(1.32%)	(1.29%)	(1.21%)	(1.99%)
Winner	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**

ZIP vs. ZIPOJA vs. AA	**3:1:2**	**2:2:2**	**1:3:2**	**0:4:2**	**3:0:3**	**2:1:3**	**1:2:3**
ZIP won rounds	296	282	385	0	369	330	326
ZIPOJA won rounds	155	86	65	70	0	173	98
AA won rounds	549	632	550	930	631	497	576
ZIP efficiency	98.28%	99.61%	104.24%	-	95.02%	97.61%	98.08%
(s.s.d.)	(8.17%)	(10.29%)	(19.21%)	-	(6.9%)	(11%)	(17.49%)
ZIPOJA efficiency	84.93%	87.7%	89.13%	89.55%	-	86.88%	86.34%
(s.s.d.)	(18.88%)	(11.64%)	(8.62%)	(6.45%)	-	(16.72%)	(11.37%)
AA efficiency	105.25%	107.97%	109.07%	116.02%	99.45%	101.39%	105.84%
(s.s.d.)	(12.48%)	(11.44%)	(12.02%)	(12.94%)	(10.57%)	(9.68%)	(8.43%)
Total efficiency	98.38%	98.43%	98.3%	98.37%	97.24%	97.71%	98.05%
(s.s.d.)	(1.83%)	(1.79%)	(1.7%)	(1.66%)	(3.04%)	(2.45%)	(2.08%)
Winner	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**

ZIP vs. ZIPOJA vs. AA	**0:3:3**	**2:0:4**	**1:1:4**	**0:2:4**	**1:0:5**	**0:1:5**	**0:0:6***
ZIP won rounds	0	395	383	0	589	0	0
ZIPOJA won rounds	79	0	214	129	0	418	0
AA won rounds	921	605	403	871	411	582	1000
ZIP efficiency	-	95.02%	98.16%	-	99.5%	-	-
(s.s.d.)	-	(10.1%)	(18.11%)	-	(17.84%)	-	-
ZIPOJA efficiency	86.9%	-	89.05%	85.12%	-	92.26%	-
(s.s.d.)	(7.97%)	-	(17.25%)	(10.68%)	-	(17.91%)	-
AA efficiency	109.28%	96.75%	97.86%	103.45%	88.47%	93.24%	-
(s.s.d.)	(8.85%)	(9.78%)	(8.8%)	(7.21%)	(13.34%)	(11%)	-
Total efficiency	98.09%	96.17%	96.44%	97.34%	90.31%	93.07%	-
(s.s.d.)	(2.1%)	(4.3%)	(3.88%)	(3.12%)	(9.04%)	(7.26%)	-
Winner	**AA**	**AA**	**AA**	**AA**	**ZIP**	**AA**	**AA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation

Table A.4: **Three-strategy experiments - ZIPOJA vs. GDX vs. AA**

ZIPOJA vs. GDX vs. AA	**6:0:0**	**5:1:0**	**4:2:0**	**3:3:0**	**2:4:0**	**1:5:0**	**0:6:0**
ZIPOJA won rounds	1000	182	153	136	166	295	0
GDX won rounds	0	818	847	864	834	705	1000
AA won rounds	0	0	0	0	0	0	0
ZIPOJA efficiency	98.42%	95.98%	93.29%	90.04%	88.66%	90.11%	-
(s.s.d.)	(1.26%)	(3.89%)	(6.18%)	(9.29%)	(11.95%)	(19.18%)	-
GDX efficiency	-	114.78%	111.75%	109.16%	105.29%	101.67%	55.31%
(s.s.d.)	-	(18.59%)	(12.26%)	(9.14%)	(5.96%)	(3.89%)	(32.6%)
AA efficiency	-	-	-	-	-	-	-
(s.s.d.)	-	-	-	-	-	-	-
Total efficiency	98.42%	99.12%	99.44%	99.6%	99.74%	99.74%	55.31%
(s.s.d.)	(1.26%)	(0.87%)	(0.67%)	(0.62%)	(0.47%)	(0.49%)	(32.6%)
Winner	**ZIPOJA**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**	**GDX**

ZIPOJA vs. GDX vs. AA	**5:0:1**	**4:1:1**	**3:2:1**	**2:3:1**	**1:4:1**	**0:5:1**	**5:0:2**
ZIPOJA won rounds	105	40	24	51	106	0	70
GDX won rounds	0	414	473	450	398	176	0
AA won rounds	895	546	503	499	496	824	930
ZIPOJA efficiency	93.81%	90.79%	86.25%	85.7%	84.81%	-	89.55%
(s.s.d.)	(4.22%)	(6.34%)	(8.64%)	(11.38%)	(16.31%)	-	(6.45%)
GDX efficiency	0%	113.58%	111.76%	106.16%	102.2%	94.7%	-
(s.s.d.)	-	(18.61%)	(12.9%)	(8.03%)	(5.09%)	(9.6%)	-
AA efficiency	123.03%	117.41%	113.43%	106.96%	104.01%	112.95%	116.02%
(s.s.d.)	(21.1%)	(20.3%)	(19.87%)	(17.6%)	(16.33%)	(17.26%)	(12.94%)
Total efficiency	98.68%	99.02%	99.28%	99.47%	99.6%	97.74%	98.37%
(s.s.d.)	(1.21%)	(1.19%)	(0.97%)	(0.89%)	(0.79%)	(7.2%)	(1.66%)
Winner	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**	**AA**

ZIPOJA vs. GDX vs. AA	**3:1:2**	**2:2:2**	**1:3:2**	**0:4:2**	**3:0:3**	**2:1:3**	**1:2:3**
ZIPOJA won rounds	45	52	62	0	79	28	78
GDX won rounds	539	496	513	366	0	617	611
AA won rounds	416	452	425	634	921	355	311
ZIPOJA efficiency	88.24%	85.55%	82.03%	-	86.9%	85.53%	82.45%
(s.s.d.)	(8.69%)	(10.07%)	(14.72%)	-	(7.97%)	(10.55%)	(15.17%)
GDX efficiency	112.17%	107.1%	103.52%	97.92%	-	110.09%	105.96%
(s.s.d.)	(16.74%)	(10.84%)	(6.75%)	(3.85%)	-	(15.77%)	(9.81%)
AA efficiency	107.64%	104.67%	101%	101.15%	109.28%	102.58%	98.38%
(s.s.d.)	(12.91%)	(10.81%)	(10.81%)	(8.87%)	(8.85%)	(8.9%)	(8.09%)
Total efficiency	98.7%	99.11%	99.09%	99%	98.09%	98.14%	98.25%
(s.s.d.)	(1.59%)	(1.35%)	(1.54%)	(1.57%)	(2.1%)	(2.18%)	(2.05%)
Winner	**GDX**	**GDX**	**GDX**	**AA**	**AA**	**GDX**	**GDX**

ZIPOJA vs. GDX vs. AA	**0:3:3**	**2:0:4**	**1:1:4**	**0:2:4**	**1:0:5**	**0:1:5**	**0:0:6**
ZIPOJA won rounds	0	129	114	0	418	0	0
GDX won rounds	696	0	687	913	0	976	0
AA won rounds	304	871	199	87	582	24	1000
ZIPOJA efficiency	-	85.12%	88.31%	-	92.26%	-	-
(s.s.d.)	-	(10.68%)	(15.11%)	-	(17.91%)	-	-
GDX efficiency	100.48%	-	109.33%	104.18%	-	114.29%	-
(s.s.d.)	(3.72%)	-	(14.08%)	(5.19%)	-	(8.83%)	-
AA efficiency	94.26%	103.45%	94.54%	87.93%	93.24%	77.38%	-
(s.s.d.)	(7.78%)	(7.21%)	(9.28%)	(8.56%)	(11%)	(10.76%)	-
Total efficiency	97.37%	97.34%	95.97%	93.35%	93.07%	83.53%	-
(s.s.d.)	(2.56%)	(3.12%)	(4.21%)	(4.54%)	(7.26%)	(7.93%)	-
Winner	**GDX**	**AA**	**GDX**	**GDX**	**AA**	**GDX**	**AA**

Results of 1000 rounds for each experiment. First row shows how many buyers and sellers following one strategy compete against traders following the other strategy (e.g. 2:4 means 2 buyers and 2 sellers following the first strategy compete against 4 buyers and 4 sellers following the second strategy. *This experiment was not run - see subsection 6.2.1 for explanation