University Name

Master's Thesis

Thesis Title

Author: Halfdan Rump

Supervisor: Prof. Toshiharu Sugawara

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

January 2014

UNIVERSITY NAME (IN BLOCK CAPITALS)

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John Smith

XXXX NOT DONE YET XXX

This work proposes a model in which multiple heterogeneous agents use time delayed price information to trade an imaginary financial instrument in a market with a continuous double auction. The main innovation of the model is that, just as in the real world, trading agents do not have access to the market information at the same point in time, which means that the agents generally trade on different information. The model contains agent models of slow human traders using a noisy fundamentalist strategy, and high speed software traders using market maker and chartist strategies. Slow traders base their trading decisions on market information which has been delayed several seconds, while the fast traders observe the market with much smaller delays ranging between a few millisecons to a few hundred milliseconds. By containing agents of such different speeds, the model is relevant to the ongoing discussion of the pros and cons of high frequency trading in financial markets.

The model simulates the market events in the minutes after the advent of bad news represented by sudden negative shock to the fundamental stock price, and does so with a time resolution high enough to register events that unfold from millisecond to millisecond. A genetic algorithm is used to search for model parameters that cause the market to be stable, and parameters that cause the market to crash. Analysis of the results shows that a moderate level of high speed trading activity is not in itself problematic. In fact, high speed market makers are found to reduce price flickering, while high speed chartists are found to decrease the time required for the market to respond to changes in the fundamental price. However, the market is found to respond unfavorably to a large presence of fast traders. First of all, a large presence of fast market makers causes the market to respond sluggishly to the shock, leading to a prolonged disparity between the traded price and the true fundamental price. Secondly, a large presence of fast chartists causes increased flickering of the traded price. Finally it is found that flash-crashes can occur in markets in which ratio of the number of chartists to the number of market makers is high, while at the same time the market makers are faster than the chartists. These results are interesting, as they show both benefits and dangers of having markets where fast computer algorithms trade side by side with human traders.

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

Abbreviations

HFT	**H**igh **F**requency **T**rader
ST	**S**low **T**rader
OB	**O**rder **B**ook
MM	**M**arket **M**aker
SD	**S**tandard **D**eviation
GD	**G**aussian **D**istribution
GMM	**G**aussian **M**ixture **M**odel
GA	**G**enetic **A**lgorithm

Physical Constants

Term	Explanation
ask price	The price of a sell order
bid price	The price of a buy order
model fitnesses	The quantitative measures summarizing how the model behaves
fundamental price	The underlying “true” value of the stock
limit order	Order which is not
market order	
liquidity	The
match	When a sell order and a bu order happen to have the same listed price, they are
order book	
order volume	
partial match	Two orders which match, but have different volumes.
share	A fraction of ownership of an asset, such as a stock
spread	
standing order	A market order registered at an order book and waiting for a matching order
tick	The smallest possible price change of the stock
volatility	

Physical Constants

$\mathrm{E_p}[\mathbf{x}]$	The sample mean of vector $\mathbf{x}$
$\mathrm{Var_p}[\mathbf{x}]$	The sampe variance of vector $\mathbf{x}$
$\mathrm{Cov_p}[\mathbf{x}, \mathbf{y}]$	The sample covariance between vector $\mathbf{x}$ and $\mathbf{y}$
$\mathrm{M_p}[\mathbf{x}]$	The sample median of vector $\mathbf{x}$
p^m	Time delay in rounds from agent i to market j. Note that $\tau_{i,j} = \tau_{j,i}$.
s_t	Spread at the end of round t
p_t^a	The lowest askprice in the order book at the end of round t
p_t^b	The highest bidprice in the order book at the end of round t
p^m	Match price, i.e., the price at which a trade is executed
f_t	Fundamental price at round t
p_{fas}	Fundamental price after shock
m_{stable}	Size in ticks of the stability margin around each side of p_{fas}
ρ_{A}	Ratio between the number of chartists and the number of market makers: $\rho_{\mathrm{A}} = \frac{N_c}{N_m}$
ρ_λ	Ratio between the latency of chartists and the latency of market makers: $\rho_\lambda = \frac{\lambda_{c,\mu}}{\lambda_{m,\mu}}$

Symbols

Symbol	Description	Unit
$\texttt{n}_{\texttt{rounds}}$	Number of simulation rounds	
$***\lambda$	Average number of ST orders per round	
N_c	Number of HFT SC agents	
N_m	Number of HFT MM agents	
$\lambda_{c,\mu}$, $\lambda_{c,\sigma}$	Mean and SD parameters of the GD for HFT SC latency	
λ_i^c	Latency of HFT SC agent i where $\lambda_i^c \sim \mathcal{N}(\lambda_{c,\mu}, \lambda_{c,\sigma})$	
$T_{c,\mu}$, $T_{c,\sigma}$	Mean and SD parameters of the GD for HFT SC thinking time	
T_i^c	Thinking time of HFT SC agent i where $T_i^c \sim \mathcal{N}(T_{c,\mu}, T_{c,\sigma})$	
$H_{c,\mu}$, $H_{c,\sigma}$	Mean and SD parameters of the GD for HFT SC time horizon	
$H_{c,\sigma}$	Standard deviation of the Gaussian distribution for HFT SC time horizon	
$W_{c,\mu}$, $W_{c,\sigma}$	Mean and SD of the GD for the HFT chartist wait time	
$\lambda_{m,\mu}$	Mean and SD of the GD for HFT MM latency	
λ_m^i	Latency of HFT MM agent i where $\lambda_m^i \sim \mathcal{N}(\lambda_{m,\mu}, \lambda_{m,\sigma})$	
$T_{m,\mu}$	Mean and SD of the GD for HFT MM thinking time	
T_m^i	Thinking time of HFT MM agent i where $T_m^i \sim \mathcal{N}(T_{m,\mu}, T_{m,\sigma})$	
V_i^c, V_m^i	Volumes of orders submitted by HFT SC and HFT agents, respectively.	

For/Dedicated to/To my...

PREFACE As such, this project turned out to be just as much about software engineering as The work presented in this thesis was the final project in

Chapter 1

Background

1.1 Introduction

One of the great challenges of science today is to model systems in which the system dynamics are influenced human behavior. The purpose of trying to describe such systems through models is to obtain tools which allow prediction, and ultimately makes it possible to enforce control.

Traditional economics tries to understand such systems, by using an analytic approach, requiring many, simplifying assumptions. In spite of these assumptions, classical economic theory has proven itself capable of modeling many of the dynamics of human society where the exchange of goods for money takes place

At its core, economic theory is mostly concerned with how large groups people can be expected to behave on average, by interacting with other human beings. As such, the theories are not suitable for explaining situations arising from recent technological developments. Of interest to this papers are especially two such cases:

1. Situations where humans and machines interact.
2. Situations where a few, exceptional individuals are responsible for a large part of the aggregate activity

In terms of the first item, the financial sector has changed drastically during the last few decades. First of all, the use of computers to handle mundane tasks is no longer news, and has become commonplace. Furthermore, with the rapid development of artificial intelligence and soft computing, algorithms are now also involved in the trade decision process itself, rather than just doing simple trade execution tasks. Hence, rather than

being a system where humans interact with humans and markets, the current financial system is one in which humans and machines interact with markets, humans and machines.

As for the second item, the financial sector has changed as well, due to the invention on a new set of trading strategies called high frequency trading. High frequency trading uses computers and algorithms to analyze market data and trade faster than any human can. By directly accessing the order book information which shows at which prices other traders are currently willing to trade, they are also able to get an accurate picture of the current market state. By placing themselves physically close to the markets (the practice know as co-location), they are able to do so at a very high time resolution. Again, by investing heavily in communication equipment, they are able to process large amounts of information and quickly reach trading decisions.

In other words, economic theory today needs to cope with a reality in which there are a relatively few number of traders who are exceptional in the sense that they have access to technology that other traders do not, and that they use that fact win over other humans traders and less sophisticated trading algorithms. Due to the increased complexity of the analysis required to understand such systems, and due to a number a violent financial crashes in recent years, some economists have started expressing an interest in the use of computational models to understand economic systems as a whole. A few such papers are mentioned in section 1.2.

In parallel, the idea of multi-agent systems has risen in the field of computer science. Because computers are increasingly linked together in networks, they interact with each other, and hence comprise a system in which their aggregated behavior can be analyzed.

Computers, or rather the software that we call agents, are relatively simple to model. If the software is simple enough, it might not even require modeling, and the software agents can be plugged directly into a simulation engine. Much of the work in done in MAS carried out by researchers from the computer science community the field of multi-agent simulation (MAS) has been concerned with modeling and analyzing such system where machines interact with other machines.

MAS is a well suited tool to deal with the complexities of the modern trading world, because it allows for the construction of models with complex dynamics. Naturally, the creation of realistic agent models is difficult. However, as is done in this paper, it is possible to capture essential characteristics of the problem under investigation. In this paper, we create a model of a simple trading environment with one market, and several agents trading on that market. The aim of creating this model is to find out how artificial markets behave when there is a communication delay between the agents and

the market. The contents of this paper is as follows. First, in section 1.2, we summarize the main influences of this work, and explain a few of the ideas that were central to the development of our own model. Next, in section

1.2 Related work

Artificial market simulation is a popular field with many different viewpoints. Due to the multi-disciplinary nature of the background of the researchers who contribute (economists, computer scientists, ecologists, sociologists, etc.), it is not within the scope of this section to give comprehensive review of the literature. Rather, the aim of this section is to show which influences lie behind this work, and where we have tried to add new contributions.

[1] proposes a round-based model in which a fast trader using a market maker strategy trades against a large number of slow traders, implemented as the stylized trader model laid out in [6]. The paper shows a link between the frequency with which the traders trade and their trading success. The notion of a trading frequency is also present in our model, as assign each agent with a time delay to the market which means that the agent can never trade faster than it takes for information to flow between itself and the market.

In [4], the authors show that as the time resolution with which information from stock market is observed is increased, the the distribution of price movements of the traded asset changes from the usually observed power law. suggesting that the presence of very fast traders creates markets which behave in fundamentally different ways. The authors then proceed to discuss the phenomenon of strategy crowding, which they argue is one reason why many flash-crashes can be observed in recent years. The crux of the argument is that the faster the agents have to be, the less information they have to trade on, and this means that there are only so possible conclusions that can be drawn. As the agents react similarly, they can cause crashes, especially if those agents are responsible for trading large volumes. Our model adds an extra condition for this to happen. Not only must the agents be similar, but they must also receive the same information at the same time, which may or may not happen.

In cite [7], the authors point out that the share of the volume traded by high frequency traders vary widely from market to market, but that in some markets it is estimated to be as high as 70%. The same paper also goes to great lengths to specify exactly what high frequency trading is, and lists a number of defining characteristics. A few of them are “Very high number of orders”, “rapid order cancellations”, “profit from buying

and selling (as middlemen)", "low latency requirement", "use of co-location/proximity services and individual data feeds". These are (some of) the typical characteristics that high frequency traders exhibit, and they have all been replicated by our model. If we want to understand markets in which high frequency trading is taking place, these characteristics should be replicated, rather than trying to mimic specific strategies which are used in the real world. Another argument for this is that, unlike the more traditional chartist strategies, modern algorithmic trading trading strategies use complex probabilistic models, which are very difficult (if not impossible) to reverse-engineer. In many cases, the only way to judge the efficacy of a trading algorithm is to implement several types of algorithms and let them trade against each other, as was done in [5].

Because of the high degree of secrecy surrounding which algorithms are actually being used for high frequency trading, and since the practice is relatively new, [2] argues that there is a general lack of knowledge about the topic, even among professionals in the financial field. To remedy that fact, it tries to establish a broad taxonomy of strategies employed by high frequency traders, and basically divides them into market making strategies, arbitrage strategies, and market rebate trading strategies.

[8] is another work arguing for a taxonomy of the trading strategies, but does so from an ecological perspective by assigning roles of prey and predator to traders. This way of thinking is closely related to the discussion of whether or not HFT is beneficial or harmful to markets, as it is intuitive to think of HFTs as the predators and other traders as they prey. The default assumption seems to be that it is not beneficial, partly because of the large profits, but some findings seem to indicate that HFT actually improve liquidity in markets, and argue that the profit gained by the HFT is the cost imposed on other traders. Indeed, since financial institutions expected of employing HFTs themselves seem to affected by flash crashes in their company stock, as was pointed out in [4]. [3] indirectly presents an argument for this notion of predator vs. prey. It is interested in the phenomenon of flickering quotes, i.e., small, momentary differences in trade prices between markets. The paper uses a simple, round-based model, and presents equations for when the fast traders show use their speed advantage to deliberately sell at sub-prime prices to the slow traders.

In summary, much interesting work has been done in order to understand the role and impact of high frequency trading, and algorithmic trading in general. Some works try to reach conclusions by analyzing real stock-data, whereas other papers approach the problem differently by first building a model, and then perform experiments to see if realistic dynamics can be replicated by the model. The latter approach is typical for papers in the MAS field, and this paper is one such paper. However, to the limit of the author's knowledge, no MAS-model has yet attempted a real-time simulation approach

as is proposed in this paper. Since high the timing of trading is very important for high frequency trading, we believe that a model which simulates that timing is necessary.

Chapter 2

Model

2.1 Model

As explained in section **??**, perhaps the most distinguishable aspect of high frequency trading is the speed with which agents can react to new market information. It is therefore essential that a model should capture this aspect, if it is to be used to draw generalized conclusions about the influence of high frequency trading in the markets.

2.1.1 Overall architecture

The model consists of a market and agents. Agents and the market communicate by exchanging messages which all arrive one or several rounds after they are issued. A complete simulation consists of the several consecutive rounds. In each rounds, some agents submit orders, while others wait for new market information. Order messages arrive at the order books, and trades are executed when prices match. The following sections will describe the model in detail.

2.1.2 Modeling delays

Although each round corresponds to a period of real-time, it is not particularly important to specify how long that period is. Instead, what matters is that there is a difference in speed between the agents. In other words, the important thing is that some agents are much faster than other agents. If one thinks of each round as a millisecond of real-time,

one realizes that an agent simulating a human trader will require several thousands of simulation rounds to react to market news. On the other hand, fast algorithmic traders may only require a few rounds, making them several orders of magnitude faster than the slow traders.

This focus on the extremely

Another issue when simulating

FIGURE 2.1

2.2 Auction type

2.3 Model components

Sending/receiving orders, supply liquidity

2.3.1 Stocks

A stock is an asset which is traded on a market. A stock is only worth as much as people are willing to pay for it, and the price at which is it traded thus goes up and down

according to what beliefs people hold. In financial markets, every trader is supposed to have access to the same information. However, two traders might disagree on the meaning of some piece of information. The way in which a trader evaluates market information and reaches a conclusion on how to trade is called a strategy. While any function which takes some information relevant to the market as input and gives a decision of how to trade (or not trade at all) can be termed a strategy, it is useful to divide strategies into two broad categories. In the first category are strategies which are dubbed chartist strategies, which basically tries to extrapolate on the past price movements. In the other category are strategies which are based on some analysis of the true value of the stock, called the fundamental value. Not surprisingly, these are called fundamentalist strategies.

Whether or not one type of strategy is more accurate than the other, it is a fact that both types are employed by traders. Hence, a model of such an environment needs to simulate both a traded price and a fundamental price.

2.3.2 Messages

All communication between the market and agents is transmitted in messages and all messages take a non-zero number of rounds to arrive. This means that no information is transmitted instantly between agent and market. The latency between any agent and the market is constant throughout the simulation, but different for each agent. The delay between market maker MMi and the market is λ_m^i, and the distance between chartist SCj and the market is λ_j^c. All latencies between agents and markets are non-negative integers. A message created in round tT by an agent with a latency of λ rounds to the market with arrive at the market in round $t = tT + \lambda$. Several message types were implemented in order to accommodate the various types of communication.

2.3.2.1 Market information

One of the key points of simulating delays is that agents always trade on old information. Before an agent can evaluate its strategy, it has to request the most recent market information. In a model without delays, an agent would simply receive the state of the market in the current round, but when information is delayed the process is somewhat more cumbersome. First the agent sends off a request to obtain the information about the market state. When the request arrives at the market some rounds later, the market serves the request and by sending back another message containing the information. The contents of this message depends on the agent strategy, as the various agent strategies require different information. In the case of a single market, it is reasonable to simplify

FIGURE 2.2

the model such that the market serves the request instantaneously, since any delay inherent in the market is common for all agents. After a further delay, the message containing the market state finally arrives at the agent, and the agent can then start evaluating its strategy. Figure 2.3 summarizes the procedure.

This is analogous to how

2.3.2.2 Orders

An order is a message which is sent from an agent to a market when the agent has decided to trade. An order is an offer to buy or sell a specified number of shares at a certain price at a certain market. Orders can either be limit orders or market orders. A limit order will only result in a trade to be executed if there is a matching order when it arrives to the market. A market order will stay in the order book until a matching order arrives, or until it expires after a number of rounds set by the submitting agent.

When an agent creates an order, the agent first of all decides on whether the order is a sell order or a buy order, a limit order or a market order. The agent also specifies the price and the volume of the order. Details on how each type of agent does this can be found in section 2.4. Once the price has been decided upon, it stays fixed for the

FIGURE 2.3: When an agent wants to submit an order it has to go through several steps of interaction with the market. The process is comparable to how real traders communicate with markets via a network, such as the Internet.

duration of the order lifetime. On the other hand, the volume does not remain constant. If the agent submits a market order, the order sits in the order book for a number of rounds before it expires. If during that time there is an order on the other side of the book with the same price, a trade is executed. If the two orders involved in the trade do not have identical orders, a partial match occurs (see section 2.3.3.2), and the volume of the largest order is updated. When this happens, the market instantly sends out a transaction receipt to the agent. However, since it takes time for the receipt to reach the agent, there is a period of time during which the agent is unaware that the order that it places has been involved in a trade. Hence, every order has two simultaneous volumes. The market-side volume specifies the actual remaining volume of the order as it is in the order book, while the agent-side volume reflect what the agent knows about the order. Whenever an order is involved in a trade, a discrepancy between the market-side volume and the agent-side volume occurs, and is resolves as the transaction receipt reaches the agent. This discrepancy between the market-side and the agent-side volumes can cause some conflicts.

XXXWHICH CONFLICTS?XXX For instance, an agent might observe a change in the best price at a point in time prior to the trade execution, and decide to cancel its current order and submit another order at the new best price. If the agent decides to do so at

any point of time during which the transaction receipt is travelling towards the agent, the new order submitted by the While the market-side order can be thought of as the "true" volume, the agent does not know this, and therefore bases its trade decision on the agent-side volume.

2.3.2.3 Transaction receipts

When two orders match, a receipt is sent to each of the two agents involved in the trade. The seller receives a receipt specifying the number of shares that it has to deliver, and the buyer gets a receipt for the amount of cash to be paid. Because of the transmission delay, the agents to not update their portfolios when the trade actually happens, but when they receive the receipt. In the case that an agent does not have enough shares or cash in its portfolio, the agent is allowed to borrow the necessary assets, thus bringing its portfolio into negative. An agent cannot submit new sell orders while holding a negative number of shares. Similarly, an agent cannot submit any new buy orders while having a negative amount of cash. In the case that the agent has neither cash nor shares, it simply becomes inactive.

2.3.2.4 Order cancellations

It can happen that an agent wants to change a previously submitted order, or cancel it entirely. In fact, this is what the market maker agent does frequently, as described in section 2.4.2. In this case, the agent issues a message to the market requesting that the order should be removed. Due to the presence of delays, the agent's order might be filled before the cancellation reaches the market, in which case the market will ignore the request to cancel.

2.3.3 Order book

The order book is a record of all unmatched orders for a single stock. Since any buy- and sell orders submitted at the same price will cause a trade to be executed, and the matched orders to subsequently be removed, there must at any point of time during the simulation be a non-negative price difference between the sell order with he lowest price and the buy order with the highest price. This difference is called the *spread*, and is denoted as follows

$$s_t = p_t^a - p_t^b \tag{2.1}$$

where p_an is the best (that is, lowest) sell price and p_bn is the bets (highest) buy price, both at round tn.

ASK-volume	Price	BID-volume

TABLE 2.1: TABLE ILLUSTRATING ORDER BOOK

2.3.3.1 Price updating

Each time an order is added or removed, the order book has to update the best bid and ask prices. Since it often happens that several orders arrive in the same round, the order book spread can fluctuate within a single round. However, since one round is considered the minimum quantum of time, these within-round fluctuations are not recorded in the order book history. Instead, after all orders have been processed, the resulting best bid and ask prices are registered as the best prices for that round. Agents that look at the market will therefore only be able to see the state of the order book after the book has finished processing all price changes due to the arrival or removal of orders. The subscript denoting time in equation 2.1 therefore refers to the prices at the end of that round. This difference between the traded prices and the best prices is shown on figure 2.4

FIGURE 2.4

When no orders arrive or are removed from the order book, the prices are updated as $p^a_{t+1} = p^a_t$ and $p^b_{t+1} = p^b_t$. Since orders can be removed due to cancellations or because they expire, the order in which incoming messages is processed matters to the outcome

of p_t^a and p_t^b. Messages are therefore processed in random orders, so that no agent is favored.

2.3.3.2 Order matching

When a trade is executed between orders o_1 and o_2, the traded volume is

$$\Delta v = \min(v_{o_1}, v_{o_1})$$

If $v_{o_1} = v_{o_2}$ the orders are *fully matched*, and both are removed from the order book. In the case of a *partial match*, that is, if the volume of one order is larger than the volume of the other, then the order with the smaller volume is removed, and the volume of the other order is subtracted by Δv. The price of the transaction is the price of the market order which was already in the book.

Each agent knows the volume of every order that it submitted, when the order was dispatched. However, when an order is partially filled by a matching but smaller order, the volume of the order at the market changes. Since it takes time for the order to be transmitted for the agent to the market, the agent cannot immediately update its knowledge of the order volume. In this case the order has one volume at the agent side and another in the market side. The momentary disparity of agent market side and agent side volumes can have several consequences, such as agents short-selling without, agents submitting cancellations for orders which have already been filled. Rules that handle these situations are described in section **??**. Unlike volumes, the price of a standing market order does not change, and hence the situation of a disparity between market- and agent-side price knowledge does not occur.

2.3.4 Market

2.3.4.1 Short selling

Although some market do allow deliberate short selling, this practice is not allowed in the simulation. That is, an agent is not allowed to place a sell order for more stock than it has in its portfolio at the time it places the order. However, due to the presence of delays, it can happen that an agent is required to deliver on a sell order for more stocks than it holds when notified of the order. A sequence of events which causes this to happen is illustrated on figure 2.5. The agent who is short is required to deliver the stocks, and thus goes into negative on its portfolio, and has to buy back the stocks before it can place further sell orders. Although the sequence of events shown in figure

2.5 may seem unlikely, it did in fact occur frequently, making it necessary to implement handling of this special case.

FIGURE 2.5: The HFT agent submits a sell order for 100 stocks, and another agent submits a price-matching buy order which fills the sell order. Before the transaction receipt reaches the seller, the seller decides to cancel the order, and submit another order at a different price. When the transaction receipt reaches the seller, the agent promptly sends out a cancellation of its second sell order, as it knows it cannot fulfill the order. However, before the cancellation reaches the market, a third agent fills the sell order, and a receipt is send to the seller who ends up being short.

2.4 Agents

The model contains three types of agents, each employing a different strategy. Each strategy has several parameters which greatly impact the behavior of the agent.

The main purpose of this work is to model a market in which some agents are much faster than other agents. To this effect, agents are divided into two groups. The first one is the group of slow traders, which are meant to represent human traders, and algorithmic traders using long-term strategies. The other group is the strategies representing the high frequency traders. In this report, HFT agents will often be referred to as being "fast" or "slow" when talking about agents with small or large latencies to the market. While an agent's latency to the market is unrelated to how fast or slow an agent is

(other parameters, such as T_m^i, T_i^c is more related to an agent's speed), the agent's latency does determine the delay with which the agent received market information. By fast and slow agents is therefore meant agents that can or cannot respond quickly to new market events.

2.4.1 Slow traders

The stylized trader model used in this work is inspired by the model used in [1] and [6]. However, due to the fundamental differences in the way that the simulation works, the model has been modified significantly.

The idea of the model is that there are three basic trading strategies techniques that any human trader trader mixes to form his own personal strategy.

Fundamental analysis Traders subscribing to this way of thinking believe that they can know the true value of a stock by estimating the fundamental price (see section **??**). Furthermore, such traders believe that any deviation from the fundamental price is due to other traders misinterpreting the market, and that such deviations will eventually disappear. In other words, given enough time, the traded price will converge towards the fundamental price.

Technical analysis Traders using technical analysis do not care whether or not the stock is over valued. Instead, they believe that they can predict future price movements from past data. Traditional technical analyst approaches extrapolates on price movements by using simple mathematical models and a good deal of heuristics.

Noise trading Some traders are in possession of insider knowledge, which means that they think that they know something about the stock that others do not. They use this information to trade the stock, for better or for worse. Such traders were dubbed noise traders, since it is highly unlikely that any one individual would come in possession of information which actually gives that person an advantage in the market. Any such belief is therefore naive, and might as well inflict a loss on the agent than generate a profit.

In this model the fundamental and noise trader parts of the stylized trader strategy are preserved, but the chartist behavior is removes. Most chartist strategies operate on a timescale of days to weeks to months. However, the simulation proposed in this

work merely simulates a few minutes of real-time[1]. Any chartist strategy based on the accumulated history of price movement over a long period of time will simply be too sluggish to follow with the high frequency price fluctuations occurring within the simulation. In other words, to a slow trader that relies heavily on the chartists part of the stylized trader strategy and calculates the moving average of the traded price over a period of days or months, any price changes that take place during a few seconds will hardly register at all. Even to relatively fast stylized traders that rely of minute-by-minute price changes, the time span simulation by the model is so short that the price changes that take place during the simulation will be smoothed out by trader's moving average calculation.

This does not mean that the model neglects to recognize the fact that slow trader chartists do exist, but since these traders base their price estimates on price data from the past days, weeks or even months, it is not possible to simulate these strategies directly as it would require an exorbitant number of simulation rounds.

Secondly, this model does not concern itself with long term market dynamics. The model is limited to simulating only periods which are of key interest, mainly when a shock to the fundamental occurs.

The stylized traders play the same role as the slow trades in [3]

Price determination

(2.2)

2.4.1.1 Order arrival

The orders submitted the stylized traders throughout the run of the simulation should be thought of as being submitted by a variety of different agents, all observing the same historical data, but interpreting it differently using different strategies.

Since the period of time which the model simulates is very short, it is assumed that the average number of orders per round is constant with a waiting time until the next order following an exponential distribution. The distribution of the number of orders per round is therefore given by a Poisson process, as in []

[1] This is not entirely accurate as each round can be interpreted as an arbitrary length of real-time. However, since we are interested in what phenomena occur when we have some agents which are several orders of magnitude faster than other agents, we need a high time resolution, effectively capping the length of real-time which can be simulated

$$possiondistribution \tag{2.3}$$

As for the timing of the order, the same argument goes. In the simulation, the average number of orders arriving at each round is constant, and sent to the buy- and sell side with equal probability. Thus no attempt it made to model any kind of herding phenomenon, since there is no time in which such a thing could occur. The number of orders arriving at each round is given by a Poisson dsitribution.

$$XXXPOISSONDISTRIBUTIONXXX \tag{2.4}$$

2.4.2 HFT Market makers

Market making is the term used for strategies in which the trader is simultaneously active on both sides of the order book. The trader attempts to make a profit bu submitting sell orders at price p_a and buy order at price p_b, making sure that $p_a > p_b$. The difference between the bid price and the ask price is the market maker's spread. Trading with a small spread makes the market maker competitive by attracting orders from other traders, but also makes the market maker incur risk by increasing the risk of trading at sub-optimal prices in the case of sudden price fluctuations.

2.4.2.1 Strategy outline

The market maker aggressively tries to maintain orders at the best buy and sell prices. As other agents submit orders, the buy and sell prices move up and down, and the difference between the best buy price and the best sell price, also known as the spread, increases and decreases accordingly. Since the agent tries to maximize its own profit, it chooses prices which maximizes the spread. Hence, when the spread increases, the agent is happy to let it do so, as it can then buy and sell with a larger profit on each unit of the traded asset. On the other hand, when the spread decreases, the agent also changes its prices in order to attract trades which would otherwise go to other agents. However, if the spread becomes too small, the profit on each unit of the traded asset becomes too small to justify the risk of holding stock which might suddenly drop in value. Therefore the strategy employed by the agent specifies a minimum accepted spread at which the agent is willing to trade. When the market maker receives a notification that one of its orders has been filled, it immediately uses the most recent locally known market information to submit a new order. For the sake of simplicity, each market maker agent

is allowed to have a single market order at each side of the order book. The agent can therefore not stack orders on either side of the order book.

2.4.2.2 Parameterization

The market maker behavior is controlled by the following parameters

Minimum spread The parameter θ_i controls the smallest spread at which the

Order volume Market maker MMi submits orders with volume V_m^j. Since the market maker tries to maintain standing orders at the orderbook for as much of the time as possible, V_m^i is usually chosen to be fairly large. λ_m^i is selected when the agent is created, and is constant throughout the simulation.

2.4.2.3 Price determination

When the agent receives the delayed trade prices from the market, the spread can either increase or decrease relative to the agent's previously known prices. In the former case, the agent always updates its prices to follow the widening gap between the buy side and the sell side. When the spread decreases, the strategy requires the agent to check that the minimum spread condition, that is whether or not $s_r < \theta_i$, where s_r is spread the calculated from the trade prices that the agent has just *received*. If the condition is not violated, then the agent can freely update prices to follow the current trade prices. If the condition is violated, then the agent determines the trade price by the following equations. p_n^b and p_n^s are the *new* buy and sell prices, p_r^b and p_r^s are the best buy and sell prices that the agent currently knows of, and Δp^b and Δp^s are the absolute differences of the prices: that is $\Delta p^b = |p_n^b - p_r^b|$ and $\Delta p^s = |p_n^s - p_r^s|$.

$$\left.\begin{aligned} p_n^b &= p_r^b \\ p_n^s &= p_r^b + \theta_i \end{aligned}\right\} \text{ when } p_n^b = p_r^b, p_r^s < p_r^s \tag{2.5}$$

$$\left.\begin{aligned} p_n^b &= p_r^s - s_m \\ p_n^s &= p_r^s \end{aligned}\right\} \text{ when } \begin{aligned} p_n^b &> p_r^b \\ p_n^s &= p_r^s \end{aligned} \quad p_n^b > p_r^b, p_n^s = p_r^s \tag{2.6}$$

$$\left.\begin{aligned} p_n^b &= p_r^s - (s_m - s_r)\frac{\Delta p^b}{\Delta p^b + \Delta p^s} \\ p_n^s &= p_r^b + (s_m - s_r)\frac{\Delta p^s}{\Delta p^b + \Delta p^s} \end{aligned}\right\} \text{ when } \begin{aligned} p_n^b &> p_r^b \\ p_n^s &< p_r^s \end{aligned} \tag{2.7}$$

The equations in 2.7 are used in the case where the both the buy and sell prices moves so that the spread becomes smaller than s_m. The equations are designed so that the agent will change its trade price in proportion to the change on either side of the book. For instance, if the buy price suddenly increases a lot while the sell side only decreases

a bit, such that $s_t < s < m$, the agent will increase his own buy side price by a lot and decrease his sell price side by a little, so that the spread of between the new orders are $s_n = p_n^s - p_n^b = s_m$.

2.4.2.4 Decision cycle

Any market maker MMi be thought of as having a trading cycle, the length of which depends on the parameters λ_m^i and T_m^i.

Acquire market information The agent requests market information at time $t = t_0$. The request reaches the market at $t = t_0 + \lambda_m^i$, and the market instantly sends out a reply.

Evaluate market information The agent receives the market information at time $t = t_0 + 2\lambda_m^i$, and evaluate its strategy.

Execute trade decision The agent reaches a conclusion on how to trade at time $t = t_0 + 2\lambda_m^i + T_m^i$, and sends off an order to the market. The order reaches the market at time $t = t_0 + 3\lambda_m^i + T_m^i$.

As soon as MMi has submitted the new order, it also submits a new request for market information. The agent receives this second batch of market information at round $t = t_0 + 3\lambda_m^i + T_m^i$, and the cycle restarts. The length of the cycle of MMi is therefore defined as

$$l_{\mathrm{MM},}i = 3\lambda_m^i + T_m^i \tag{2.8}$$

The length of the cycle is important for several reasons. First of all, since the market only requests new market information once during the cycle, the market maker observed the changes in the market with a frequency of $l_{\mathrm{MM},}i^{-1}$. Any market events that take place within a period smaller than $l_{\mathrm{MM},}$i can be only partially observed by the market maker.

Secondly, since the market maker can only have a single order on each side of the order book, the latency to the market decides how much volume the market maker is able to buy and sell over a period of time. The market maker i submits orders with a fixed volume V_m^i and the volume that the agent can supply the market with therefore depends on how much of the time the market maker manages to have standing orders at the market.

When an order submitted by the market maker is filled, the market sends out a transaction receipt which takes λ_m^i rounds to reach the market maker. When the market maker

receives the receipt, the agent restarts the cycle by submitting a request for market information in order to submit a new order. This means that a total number of $4\lambda_m^i + T_m^i$ will elapse from time time that the order was filled to the time when the market maker can have another order placed in the order book. During this time, the market maker does not have an order in the market on the side of the book that the filled order was at. When the market maker does not have a standing order on either side of the book, it means that the agent does not contribute to increasing the liquidity of the asset. Hence, in this simple model, fast market maker should do a better job of supplying liquidity than slow market makers.

2.4.3 HFT Chartists

The slow traders do not directly incorporate a chartist element into their strategy, as any changes in the traded price that occurs in the short time that the model simulates would be undetected by the slow traders. However, in order to add the modeling of chartist behavior, a fast trader agent using a chartist strategy was implemented. These agents will from now on be referred to as HFT chartists, or simply as chartists.

2.4.3.1 Strategy outline

The chartists use a simple strategy, in which they continually calculate the moving average of the traded prices over a time windows. If the current price drops below the moving average, the chartist believes that it has detected a downtrend. The chartists believes that the downtrend will continue for a while, and it will therefore try to get rid of any shares that it is currently holding. In order to accomplish this, the chartists starts to submit sell orders. The more urgently the agent is trying to sell its stocks, the lower it will set the price of the sell order, as a sell order at a lower price is more likely to be filled. On the other hand, if current price rises above the moving average, the agent believes that it has detected an up-trend, and that the stock will be worth more in the near future. The agents therefore starts to submit buy orders starts to submit buy orders at a higher price. The more strongly the agent believes in the stock price increasing, the higher it is willing to set the buy price. The agent only submit orders when it has detected a trend. As long as the chartist does not detect a trend, the agent is inactive and merely observes the market data. After submitting an order, the agent waits for a while, as a precaution against submitting too many orders which would make the agent incur great losses the trend detected by the agent turns out to be wrong. This mechanism is also a limiting assumption of the model instated in order to protect the market from a few market makers flooding the market with a hige number of orders. As

another step towards preventing the order books from filling up with chartist orders that might never be filled, and thus slow down the execution of the simulation is to make the chartists submit limit orders.

2.4.3.2 Parameterisation

The agent strategy is parameterized as follows. HFT chartist agent SCj has the following parameters:

Time horizon The parameter H_j^c determines determines the width of the window over which the agent calculates the moving average. Larger values of H_j^c makes the agent use trade price data from further in the past and makes the moving average less responsive to sudden change, while lower values makes the moving average react faster to sudden movements in the price.

Sensitivity The sensitivity of SCj is given by the parameter S_j^c and refers to the number of ticks that the moving average of the traded price must differ from the currently traded price before the chartist recognizes that it has detected a trend. The higher the value of S_j^c, the less often the chartist will detect a trend. In the case of $S_j^c = 0$, the agent will detect a trend whenever $M \neq p_c$, which will be true in the majority of rounds.

Aggressiveness One the chartist decides to trade, it needs to decide upon a price. The aggressiveness parameter A_j^c controls how far from $p_c$that the agent will set the price. This parameter reflects the agents confidence in its own belief about the trend, and the willingness of the agent to trade at sub-optimal prices in order to obtain or sell off shares.

Trading frequency Once the chartist has detected a trend, it will continue trading for as long as the trend persists. The parameter W_j^c limits the frequency with which the chartist can submitting orders by specifying the number of round that the agent waits before submitting another order.

2.4.3.3 Strategy evaluation

XXX WRITE ABOUT HOW THE CHARTIST CALCULATES MA XXX

2.5 Simulation rounds

Each round is composed by a number of steps, which can be divided into five parts as described below.

The round is initialized Time is incremented and the fundamental price is updated. In this thesis, the only change to the fundamental price is at time $t = t_\eta$, where the fundamental price changes from $F0$ to $F0 - \eta$.

Arriving messages are dispatched to recipients All messages with arrival time tT are delivered. Arriving transaction receipts delivered to the agents involved in the trades, and the portfolios of these agents are updated. Similarly, arriving orders and order cancellations are delivered to the market and placed in the order book message queue. Finally, all messages containing market information are received by the agents that requested the information.

Slow trader activity The number of slow trader orders that arrive in this round is sampled from equation 2.3 and the orders are created using equation 2.2. The orders are instantly placed in the order book message queue.

Fast trader activity All HFT agents finishing evaluation of their strategies wake up and execute their decisions by sending out new orders containing orders and order cancellations. Each market maker MMi Each chartist SCj starts its hibernation period of W_j^c.

Order book is updated All orders expiring this round are removed from the order book. Next, messages waiting in the message queue are processed in random order. Orders targeted by order cancellations are removed from the book, and transaction receipts are created for each pair of buy and sell orders with matching prices. After all messages have been processed, the best bid and ask prices, B_T and A_T, are set equal to the prices of the two most competitive remaining orders. In the case that no changed were made to the order book in round tT, $B_T = B_{T-1}$ and $A_T = A_{T-1}$.

Chapter 3

Searching model behavior

The model has several parameters which must be selected carefully before the simulation can be used to infer knowledge about market behavior.

The parameter tuning turned out to consume a significant amount of time, and simple using a genetic algorithm to optimize over the entire space of parameters was not enough. Instead, the process was a slow and iterative one of running the genetic algorithm to create a data set, analyze the data set to find out what was discovered in the search, and then run the genetic algorithm again with different parameters. Thus several data sets were created, each with the purpose of examining some aspect of the simulation, or of the parameter tuning method itself.

This chapter will cover the instruments used in the optimization of the model parameters, and also mention the machine learning tools used in the analysis of the data sets.

The parameter tuning has two overall goals, which are covered in the following section.

3.1 Motivation and overall procedure

First of all, the model must be calibrated such that it mimics the behavior of real markets. Since virtually every aspect of the simulation behavior depends on the values on the various parameters, these must be chosen carefully in order for the simulation to produce realistic behavior. An example of a simulation untuned parameters causing unrealistic behavior is given in figure 3.1a. Selecting realistic parameters is by far a simple task. First of all, it requires a way of quantifying the quality of each simulation. The choice of such a quantification is discussed in section 3.2.2. Secondly, there might be several different parameter configurations which produce seemingly realistic behavior,

(A) Parameters causing unrealistic dynamics

(B) Unrealistic parameters causing realistic dynamics

FIGURE 3.1: **Motivatoin for tuning:** the two

but do not correspond to a realistic market setting. An example for this is given in figure 3.1b, and section 3.2.5 briefly discusses this point.

The second goal of the parameter tuning is to find parameters which promotes certain desirable behaviors. For instance, we might be interested in determining which parameters causes the traded price to stabilize faster after a shock to the fundamental price. Metrics for doing this is discussed in section 3.2.2

The selection of parameters is a fairly complicated process because of the large parameter space, and because it takes a significant time to evaluate the fitness of a given set of parameters[1]. amount of time to execute a simulation. Because of this, the following three-step parameter selection procedure was used.

1. Fix some of the model parameters in order to reduce the search space for the optimization algorithm. This requires us to consider which parameters can be fixed without losing opportunity to gain insight into market behavior. Essentially this step is a question of prioritizing the optimization of some parameters over the optimization of others.

2. Use an optimization algorithm to find sets of parameters which yield realistic model behavior. A genetic algorithm was chosen for this purpose, and the details are explained in section 3.2.

3. From the set of parameter combinations found by the optimization algorithm, remove the parameter combinations which obviously do not correspond to a realistic setting.

[1]The calculation time depends largely on the parameters, such as the number of agents and how active these are. Typically one to several minutes are required to evaluate a single set of parameters.

3.1.1 Selecting fixed parameters

The main parameters of interest are the ones that control the latency and speed of the agents. The agent strategy parameters are less important, since

Number of rounds Due to the computational cost of running the simulation for a large number of rounds, the the number of rounds is fixed at 10^5 for all experiments.

Order volumes

Slow trader parameters

Agent start portfolio

Order book initialization

The remaining model parameters will either be fixed for each experiment, or varied by the genetic algorithm.

3.2 Inverse simulation with a genetic algorithm

Inverse simulation refers to the technique of specifying metrics measuring model behavior, and then using an optimization algorithm to search for parameters resulting in desirable (or undesirable) behavior.

In this work, a genetic algorithm was used to search the parameter space. The algorithm proceeds as explained below.

1. Generate a population of healthy individuals, e.g., individuals with valid parameters.
2. Evaluate fitness for every individual in the population.
3. Repeat n_{gen} times
   (a) Generate offspring by crossing existing individuals.
   (b) Apply mutation to with a certain probability to each individual (parents as well as children)
   (c) Evaluate fitness of children and mutated parents.

Mutation and crossover are the operators responsible for generating variation in the population, while the selection is responsible for propagating promising individuals to future generation where they might be improved. Several possible methods of performing each of these three steps exist in the literature (see[**?**], [**?**]), and section 3.2.3 briefly covers the method and parameters of the genetic algorithm.

3.2.1 Representing parameters as genes

Since the choice of mutation and crossover operators depends on the nature of the genes, the first step towards utilizing to search the model parameters is to decide on how to encode the parameters as individuals. A set of parameters is represented by an individual, i, consisting gene for each parameter, represented by a floating point $g_{i,j}$, where j denotes the index of the parameter. When the population is initialized, each $g_{i,j}$ is drawn from a uniform distributed in the range $g_j \in [0; 1]$:

$$g_{i,j} \sim \mathcal{U}(0, 1) \tag{3.1}$$

Some of the model parameters are integers, such as N_m and N_c, and these are rounded after being scaled, and before they are passed to the simulation.

3.2.2 Model fitness

XXX NOT FINISHED YET. In order to use inverse simulation, it is necessary to decide on how to measure the quality of an instance of the simulation. In this work, the overall goal is to examine which parameter values cause the market to be stable, and which cause it to be unstable.

Another interesting point is the speed with which the market responds to the shock to the fundamental price, and which parameters influence this property.

3.2.2.1 Market overshoot

XXX

3.2.2.2 Trade price stability

XXX

3.2.2.3 Market response time

XXXX

3.2.2.4 Market tendency to stay new the new fundamental

XXX

- 'fit
- Are there certain parameter combinations which cause the market to behave in certain ways. S

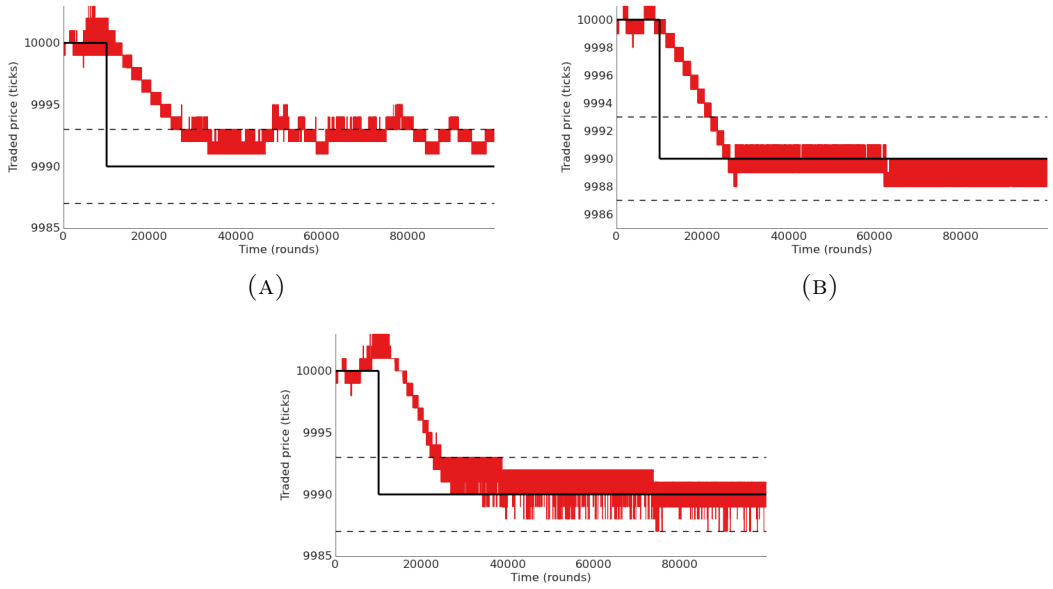
In particular the parameters controlling various time delays are of interest. The search space of the parameters is very large, which makes an exhaustive search impossible. To this end, four fitness measures were defined.

Several parameters influence the number of orders submitted by the high frequency traders.

$f_{\mathbf{s}} < f_{\mathbf{t}}$ This happens when the traded price never leaves the stability margin after reaching the new fundamental price. Note however that this case does not necessarily mean that the prices do not flicker.

$f_{\mathbf{s}} > f_{\mathbf{t}}$ This happens when the traded price leaves the stability margin once or more after reaching the new fundamental. The traded price can be close to the fundamental, but flickers in and out of the stability margin as on Figure 3.2a shows an example where the trade price fairly stable and with no overshoot, leading to good (low) $f_{\sigma}$and f_{o}fitness values to be assigned to the parameters. However, even though the traded prices are mostly within the stability margin, occasional flickers out of the margin causes the simulation to score a bad (high) f_{s}fitness. Note also that $f_{\mathrm{t}}$is undefined in this case.

$f_{\mathbf{s}} = f_{\mathbf{t}}$ This happens if a trade is executed at price $m_{\text{stable}} - p_{\text{fas}} < p^m < m_{\text{stable}} + p_{\text{fas}}$, and another trade is executed at price $p^m = p_{\text{fas}}$ in the same round.



(A) (B)

(C)

Parameter	Assignment
Number of individuals	200
Cross-over points	2
Tournament size	3
Mutation probability	0.1
Mutation distribution	$\mathcal{N}(\mu = 0, \sigma = 0.1)$

TABLE 3.1: Overview of parameters used in the genetic algorithm

3.2.3 Genetic algorithm parameters

Although this is a basic version of genetic algorithm, using it correctly is not necessarily easy, as was encountered. First of all, the parameters for the genetic algorithm itself must be established. The larger and more complex the search space, the more resources the search will require, since the evaluating the fitness function (i.e., the running the simulation) will have to be done a larger number of times.

Table 3.1 presents an overview of the parameters used in the genetic algorithm.

3.2.4 Optimizing towards desired behavior

XXX WRITE MORE. The four fitness measures defined in section 3.2.2 make it possible to specify the type of market behavior that is likely to be selected by the genetic algorithm.

First of all, we are interested in establishing which parameters cause the market to return to a stable state after the fundamental price has incurred a shock

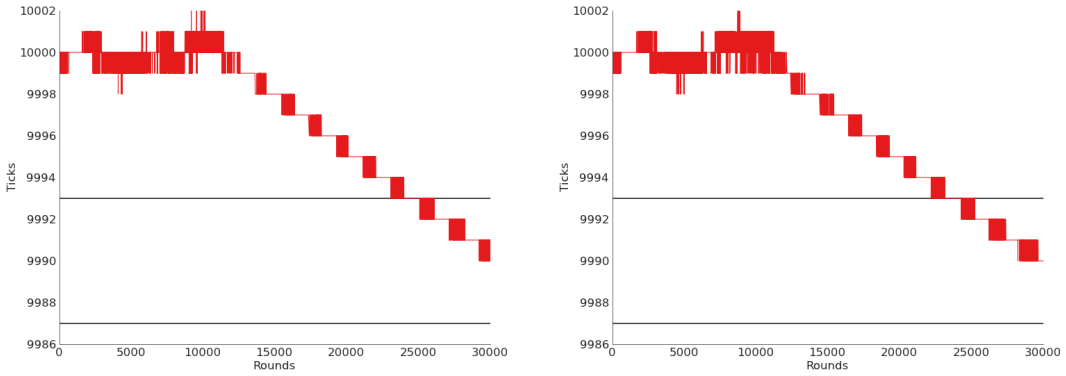
	f_o	f_s	f_σ	f_t
Overall stable market	min	min	min	min
Crash	max	-	-	-

TABLE 3.2: Optimization criteria for different types of market behavior

3.2.5 Filtering parameters

As mentioned earlier, it is not enough simply to define a fitness function which assigns high values to parameters causing realistic behavior. In addition, it is important to discard parameters which obviously do not correspond to a realistic setting. Imagine that the simulation scores high fitness values when executed without any market makers. Since it is known that real markets do in fact contain market makers, nothing can be inferred from such a result. Indeed this might be a consequence of poorly designed fitness measures, but since it is easier to use domain specific knowledge to filter out the unrealistic parameters

FIGURE 3.2: Example of a simulation which is assigned fairly good fitness values, but which was executed with clearly unrealistic parameters: $N_m = N_c = 0$. The simulation reaches the new fundamental price fairly quickly without any undershoot, and stays within the stability margin. The only point where it scores badly is the standard deviation which is slightly high due to the fluctuating trade price.



(A) $\lambda_{m,\mu}$=49, $\lambda_{m,\sigma}$=8, $\lambda_{c,\mu}$=32, $\lambda_{c,\sigma}$=19, f_o=1.0, f_t=29210.0, f_s=23970.0, f_σ=0.482

(B) $\lambda_{m,\mu}$=7, $\lambda_{m,\sigma}$=17, $\lambda_{c,\mu}$=21, $\lambda_{c,\sigma}$=17, f_o=1.0, f_t=28229.0, f_s=23146.0, f_σ=0.5

3.2.6 Handling failed simulations

Some parameters cause the simulation to act in strange ways, and even crash in some cases. For instance, the the order book becomes empty, the simulation throws an exception and terminates. Similarly, if the best bid/askprices drops to zero, the simulation exits. As such

The most common odd phenomenon was the market

3.3 Applying the genetic algorithm

Applying the genetic algorithm to produce markets with desirable behavior turned out to be more difficult that one could have hoped for. First of all, the high computational costs was a hurdle.

Secondly, many of the data sets (see section 3.3.3) produced did not contain any useful information.

The model parameters do influence the fitness values as they control model behavior, but they are not directly weighted into the fitness-values. This means that even after the parameters of the genetic algorithm was properly tuned in such a way that higher-fitness individuals were produced, these individuals often turned out to be of no interest. Such individuals were discarded according to the filtering criteria described in section 3.2.5 would have to be discarded. An example of such a case is discussed in section 3.28

3.3.1 Time complexity

The high time complexity stems from several factors

- Running a simulation for a given set of parameters required up to several minutes of computation on a single CPU core.
- Large number of model parameters increase the size of the search space. The more parameters Unfortunately there is no magic to the way the genetic algorithm works, and optimizing in a larger search means a larger time complexity.
- Large range of parameters. Some of the parameters are integers while some are real numbers. While most of the parameters have a lower bound, none of the parameters have upper bounds.
- Unstable fitness parameters. Since the same set of parameters can produce varying model behavior, the fitness-values may also vary. Therefore it is necessary to evaluate the simulation several times for each set of parameters.
- The model parameters also influence the time complexity. For instance, evaluating a simulation with many agents takes more time than evaluating a simulation with fewer agents.

Genetic algorithms are naturally suited for parallel computation. Two servers with a total of 40 cores and enough memory to evaluate as many simulations were utilized. With this equipment, evaluating a single strategy (see section 3.3.2)for generating data sets could take

reaching a point where the genetic algorithm began to produce useful output turned out to be somewhat of an iterative process.

First of all, the computational cost of running the genetic algorithm was high, which meant that the algorithm was not always able to find better individuals.

Det tager lang tid -¿ faerre params skod resultater -¿ nye experimenter

Second of all, even when the genetic algorithm did manage to improve the fitness of the population, this did not always result in useful data.

3.3.2 Experiments: Dividing the search into parts

As mentioned earlier, the number of parameters and the range of each parameter influences the complexity of the search. Because of this, it is desirable to keep the number of parameters that are included in each individual as small as possible. However, fixing parameters means that some interesting properties about the model might not be discovered. Furthermore, varying all parameters at the same time makes the analysis and

interpretation of the results more difficult. In an attempt to overcome this dilemma, several "experiments" were carried out [2]. Instead of trying to optimize all the model parameters at once, the search was split into several parts, each of which we call an experiment. Each of these experiments produce a data set, each of which were analyzed using the methods described in section 3.4. Some of the data sets produced interesting results, while others did not. Chapter **??** focuses on the analysis and presents the findings. A brief overview of the experiments is presented in **??**, but since the motivation for creating each data set is best understood in the context of the analysis of each data set, the details are deferred until chapter **??**. The next section will explain exactly what a data set is.

3.3.3 Gene pool as data set

The previous sections contain the details of each of the steps undertaken in order to produce data sets. To summarize, the list below enumerates the steps.

1. Initialize a population in the genetic algorithm with healthy individuals.
2. Evaluate the fitness for every individual several times and obtain fitness-values by calculating averages.
3. Stack all individuals that ever lived into a $N \times \mathcal{L}_i$ parameter data matrix $\boldsymbol{P}$, where N is the number of individuals, and $\mathcal{L}_i$ is the length of each individual. Likewise, stack the fitness values into a $N \times \mathcal{L}_f$ fitness-data matrix $\boldsymbol{F}$, where $\mathcal{L}_f$is the number of fitness values calculated.
4. Filter the data by removing rows in $\boldsymbol{P}$with parameters which can be deemed not to correspond to real markets, and by removing rows in $\boldsymbol{F}$with fitness values that are not realistic. Please refer to section 3.2.5 for details. Naturally, when a row is removed in $\boldsymbol{P}$, it is also removed in $\boldsymbol{F}$, and vice versa.
5. Likewise, data points which were generated by a simulation crashing before it could complete were removed.

Tables 3.3 and 3.4 contain the first rows of $\boldsymbol{P}$and $\boldsymbol{F}$for one of the data sets.

[2]The reason for the quotes is that the term experiment might be stretching the common understanding of what an experiment is a little.

	$\lambda_{c,\mu}$	$\lambda_{c,\sigma}$	N_c	$T_{c,\mu}$	$T_{c,\sigma}$	$H_{c,\mu}$	$H_{c,\sigma}$	$W_{c,\mu}$	$W_{c,\sigma}$	$\lambda_{m,\mu}$	$\lambda_{m,\sigma}$	N_m	$T_{m,\mu}$	$T_{m,\sigma}$
0	84	11	14	98	9	1071	445	38	17	3	2	48	8	1
1	23	21	74	49	24	529	554	45	13	9	0	8	5	3
2	51	13	53	47	13	3586	536	10	11	9	4	14	4	2
3	18	21	213	70	39	793	1179	33	15	7	2	43	6	3
4	94	41	144	10	25	2668	893	12	15	6	1	49	7	4
5	19	4	130	15	38	1085	1165	39	4	2	3	11	4	4
6	65	15	91	81	46	3867	1991	48	1	7	2	21	4	4
7	36	38	143	77	19	2805	1870	10	9	7	0	3	2	4
8	43	8	10	19	19	3384	1706	33	4	8	4	5	5	0
9	11	33	127	94	49	3597	723	12	2	7	1	33	5	4

TABLE 3.3: An example data matrix containing the parameters of ten individuals who lived sometime during the execution of the genetic algorithm. In this case, each individual contained parameters for the number of HFT agents, as well as the latency and thinking time parameters. Hence, the data matrix has a column for each parameter.

	f_o	f_s	f_σ	f_t
0	3	25359	0.382092	29838
1	7	99999	1.289659	23373
2	6	99999	1.253363	18748
3	7	99997	1.695150	22819
4	6	94343	1.329276	22703
5	16	99999	2.439084	31860
6	6	93378	1.287235	25645
7	10	99997	1.858166	19417
8	3	24039	0.935465	27381
9	19	99995	4.092439	24845

TABLE 3.4: This table contains the fitness values for each individual in table 3.3. Note that, in order to increase the reliability of the fitness measure of an individual, the recorded fitness-values are the average of the fitness-values obtained by evaluating each individual ten times

3.4 Data analysis

Using inverse simulation merely creates a lot of data. This data has to be analyzed before any

Data normalization

3.4.1 Data visualization

It is useful to be able to plot the data points

3.4.1.1 Color toned scatter plots

Scatter plots are useful for initial data analysis, as one can quickly detect problems such as outliers, and maybe even detect clusters of data points.

Scatter plots are probably among the most rudimentary of techniques for data analysis, yet they can be incredibly informative, especially when the data that is visualized is low-dimensional. The two plots in figure make two things clear. First of all, extreme values occur in f_σ. Even log scaling does not seem to fix this problem entirely. Second of all, XXXissue109XXX

3.4.2 Preprocessing

3.4.2.1 Handling outliers

The term outliers is often used as a label for data which is considered "invalid" in the sense that is not a product of the true data generation process, but due to various sources of noise. In this report, data that was caused by failing simulations corresponds to the usual understanding of outliers. However, as was already explained in section 3.2.6, data from such simulations are never included in $\boldsymbol{P}$ and $\boldsymbol{F}$ to begin with. Instead, outliers in this report refer to entries in $\boldsymbol{P}$ and $\boldsymbol{F}$ deviate significantly from the majority of the data points by having extreme values.

Such data points caused problems when applying data analysis techniques which rely on a fairly normal distribution of the data points, such as Principal Component Analysis (PCA). PCA looks for a rotation of the data space, such that the axes of the new basis are aligned with the directions of the largest variance in the original space. Since a few points with extreme values come to account for a large portion of the data set variance, the new basis computed by PCA will be aligned along these few data points. When PCA is used to extract lower dimensional features from the data set, outliers will degrade the quality of these features. In the case that PCA is used for data visualization, the scatter plots of the first few principal components will not be very informative, as they merely show the projection of the data onto the axes aligned with the outliers.

In all experiments, outliers were present in both the parameters space and in the fitness space. Outliers in the parameter space occur because of abnormally large mutations, and any dimension of the parameter space is susceptible to such an event. In the fitness space, only a few of the features suffered from the occurrence of outliers. The feature $f_\mathrm{s}$cannot contain outliers, because the shock to the fundamental occurs at round $t = 10^4$, and because the simulation is terminated at round $t = 10^5$, hence $f_\mathrm{s} \in [10^4, 10^4 + 1, \ldots 10^5]$.

The same is true for f_{t}. $f_{\sigma}$and f_{o}, on the other hand, are susceptible to outliers, because the two features have no upper bound.

1. Apply a monotonically increasing transformation $f(x)$ to some or all of the features for every data point in the data set. A common choice of f is $f(x) = \log x$, as it efficiently reduces the impact of data points with extremely high values. Figure ?? illustrates the effect of applying the log-transform. The log-scaling does a fairly good job of reducing the importance of the outliers in the f_{σ}feature, while the change is less dramatic in f_{t}. While the left scatter plot does not really reveal any structure of the data, the transformation makes it possible to spot to rough clusters when inspecting the right plot.

   Another simple method is to manually remove

2. Manually select one or more criteria for when a data point is to be considered an outlier. It is worth noticing that $f_{\mathrm{o}}$and f_{σ}

3. Use a one-class support vector machine to calculate a probability for each data point that said point is an inlier or an outlier. In this case, inliers

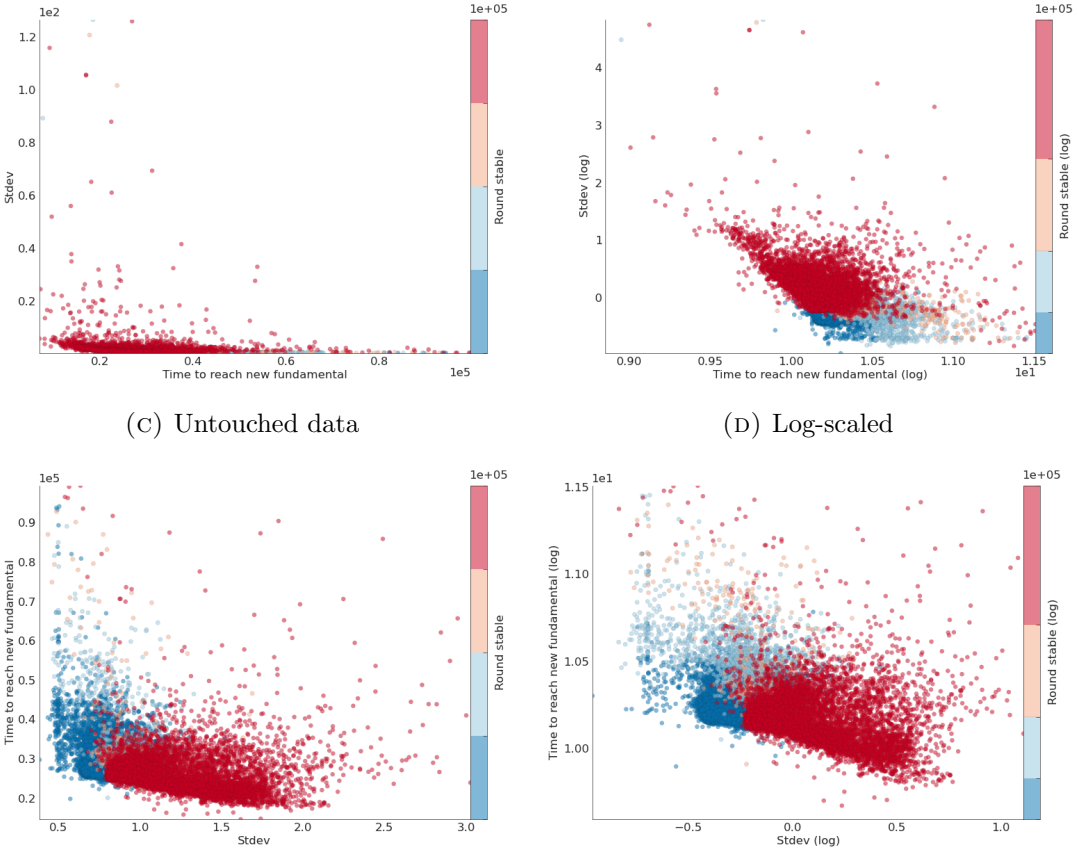
In the parameters data set, outliers can occur due to abnormally large mutations. Some parameters cause the simulation to act in strange ways, and even crash in some cases. For instance, the the order book becomes empty, the simulation throws an exception and terminates. Similarly, if the best bid/askprices drops to zero, the simulation exits. As such

The most common odd phenomenon was the market

3.4.3 Clustering algorithms

3.4.3.1 GMM

covariance type:full



(C) Untouched data

(D) Log-scaled

(E) Outliers manually removed

(F) Log-scaled and without outliers

FIGURE 3.3: Color toned scatter plots of f_{t}, $f_{\sigma}$and f_{s}taken from dataset d1after applying log-scaling and manually removing outliers

Experiments

As mentioned in the previous chapter, the process of finding was an iterative one of running an experiment, analyzing the generated data, draw conclusions and then repeat the steps with a new experiments designed to amend the mistakes of the previous experiment. This chapter will go through the results that were obtained from each of the data sets. A summary and discussion of the results is found in chapter 3.9.

3.5 Overview of experiments

The model has many parameters, and doing an exhaustive search over the entire parameter space is not possible. Instead, a genetic algorithm (GA) was used to do targeted searches of the parameters. For the details of the GA, please see chapter **??**. Depending on how the GA was set up, different areas of the search space was searched. Even when using a GA, some parameters had to be fixed. However, fixing parameters means that the effect of the fixed parameter on the behavior of the model remains unknown, since only a subspace of the entire parameter space is searched. For this reason the genetic algorithm was executed several times, each creating a data set containing fitness values for different parts of the parameter space. Each of these data set can be analyzed, providing information which can be corroborated in order to form an understanding of the overall market behavior.

Table 3.5 contains an overview of the different data sets, showing which parameters were fixed, and which were included as genes in the genetic algorithm.

In the following four experiments, the genetic algorithm was set to minimize all four fitness-measures.

d1: Varying the number of HFT agents, and all latency related parameters
This data set was generated by including all the model parameters concerning time latency as well as the number of agents into the individuals in the genetic algorithm. Due to the high number of variables, the data turned out to be difficult

to analyze, as too many factors pertaining to the simultaneous change of several parameters influenced the fitness values.

d9: Fixing the number of agents while varying latency parameters The analysis of d1 showed that when minimizing the four fitness-measures, the genetic algorithm tended to select model containing few or no HFT agents. The case of a market with no market makers and no chartists can safely be said to be trivial. Hence, in experiment d9, the number of HFT agents were fixed to $N_m = 30$ and $N_c = 100$.

d10: Fixing the number of HFT chartists Since d9kept $N_m$and $N_c$constant, the experiment did not reveal anything on how the market behavior changes when the number of agents changes. In order to investigate the impact of having many or few HFT market makers, $N_m$was varied in this experiment. Although it is also of interest how the market behavior depends on the number of HFT chartists, including $N_c$as a gene would yield results similar to those obtained in d1. For this reason the number of HFT chartists was fixed to $N_c = 150$.

d11: Fixing the number of HFT market makers This experiment was carried out in order to investigate the impact of the number of HFT chartists on the market behavior, and is supplementary to d10.

3.5.1 Correlation between fitness measures

A factor which influences the evolution of parameters is correlation between the fitness-measures. If two or more fitness measures have non-negative correlation coefficients, individuals will be statistically more likely to get good scores in the correlated fitness measures at the same time. Since all fitness measures are given equal weight in the selection process, individuals scoring well in the correlated fitness-measures will win over individuals which score well on another, statistically independent fitness measure. It is therefore important to compare the selection tendencies with the correlation between fitness-measures. Figure 3.4 shows a plot of the correlation matrix for d10. Since later generations will be affected by the biased selection and therefore contain more individuals which did well on the correlated fitness measures, the correlation coefficients in the figure were calculated over individuals in the first generation only.

For instance, the correlation between $f_o$and $f_\sigma$means that an individual which scores a good f_o-fitness will be statistically likely to also score a good f_σ-fitness. Since all four fitness measures are weighed evenly in the selection, models with behavior which

ID	Description	Fixed parameters	As genes
d1	All parameters varied	$V_{c,\mu} = 10, V_{c,\sigma} = 3, V_{m,\mu} = 50, V_{m,\sigma} = 20, S_{c,\mu} = 2, S_{c,\sigma} = 5, A_{c,\mu} = 3, A_{c,\sigma} = 2$	$\lambda_{c,\mu}$, $\lambda_{c,\sigma}$, N_c, $T_{c,\mu}$, $T_{c,\sigma}$, $H_{c,\mu}$, $H_{c,\sigma}$, $W_{c,\mu}$, $W_{c,\sigma}$, $\lambda_{m,\mu}$, $\lambda_{m,\sigma}$, N_m, $T_{m,\mu}$, $T_{m,\sigma}$
d9	Fixed number of HFT agents	$N_m = 30$, $N_c = 100, V_{c,\mu} = 10, V_{c,\sigma} = 3, V_{m,\mu} = 50, V_{m,\sigma} = 20, S_{c,\mu} = 2, S_{c,\sigma} = 5, A_{c,\mu} = 3$, $A_{c,\sigma} = 2$	$T_{c,\mu}$, $T_{c,\sigma}$, $H_{c,\mu}$, $H_{c,\sigma}$, $W_{c,\mu}$, $W_{c,\sigma}$, $\lambda_{m,\mu}$, $\lambda_{m,\sigma}$, N_m, $T_{m,\mu}$, $T_{m,\sigma}$
d10	Fixed number of HFT chartists and fixed strategy parameters	$N_c = 150$, $T_{m,\mu} = T_{c,\mu} = 50$, $T_{m,\sigma} = T_{c,\sigma} = 20$, $H_{c,\mu} = 5000$, $H_{c,\sigma} = 2000$, $W_{c,\mu} = 50$, $W_{c,\sigma} = 20, V_{c,\mu} = 10, V_{c,\sigma} = 3, V_{m,\mu} = 50, V_{m,\sigma} = 20, S_{c,\mu} = 2, S_{c,\sigma} = 5$, $A_{c,\mu} = 3, A_{c,\sigma} = 2$	N_m, $\lambda_{c,\mu}$, $\lambda_{c,\sigma}$, $\lambda_{m,\mu}$, $\lambda_{m,\sigma}$
d11	Fixed number of HFT market makers and fixed strategy parameters	$N_m = 52$, $T_{m,\mu} = T_{c,\mu} = 50$, $T_{m,\sigma} = T_{c,\sigma} = 20$, $H_{c,\mu} = 5000$, $H_{c,\sigma} = 2000$, $W_{c,\mu} = 50$, $W_{c,\sigma} = 20, V_{c,\mu} = 10, V_{c,\sigma} = 3, V_{m,\mu} = 50, V_{m,\sigma} = 20, S_{c,\mu} = 2, S_{c,\sigma} = 5$, $A_{c,\mu} = 3, A_{c,\sigma} = 2$	N_m, $\lambda_{c,\mu}$, $\lambda_{c,\sigma}$, $\lambda_{m,\mu}$, $\lambda_{m,\sigma}$

TABLE 3.5: Overview of datasets

FIGURE 3.4: Correlation matrix of the four fitness measures in the first generation of dataset d10

is assigned good values for $f_o$and $f_\sigma$will score a better overall fitness than a simulation with a good fitness

In other words, the correlation between $f_\sigma$and $f_o$means that stable individuals will outlive fast individuals as they are selected for breeding more often. This is not a property of a model itself, but rather a problem with the definition of the fitness measures. This problem can be circumvented by not using of the correlated fitness values. XXX

Both $f_o$and $f_\sigma$were used for the GA selection, and although these two fitness measures do reflect different properties of the simulations, they were found to be somewhat correlated. That is, a simulation which tends to have a small overshoot also tends to have stable traded prices.

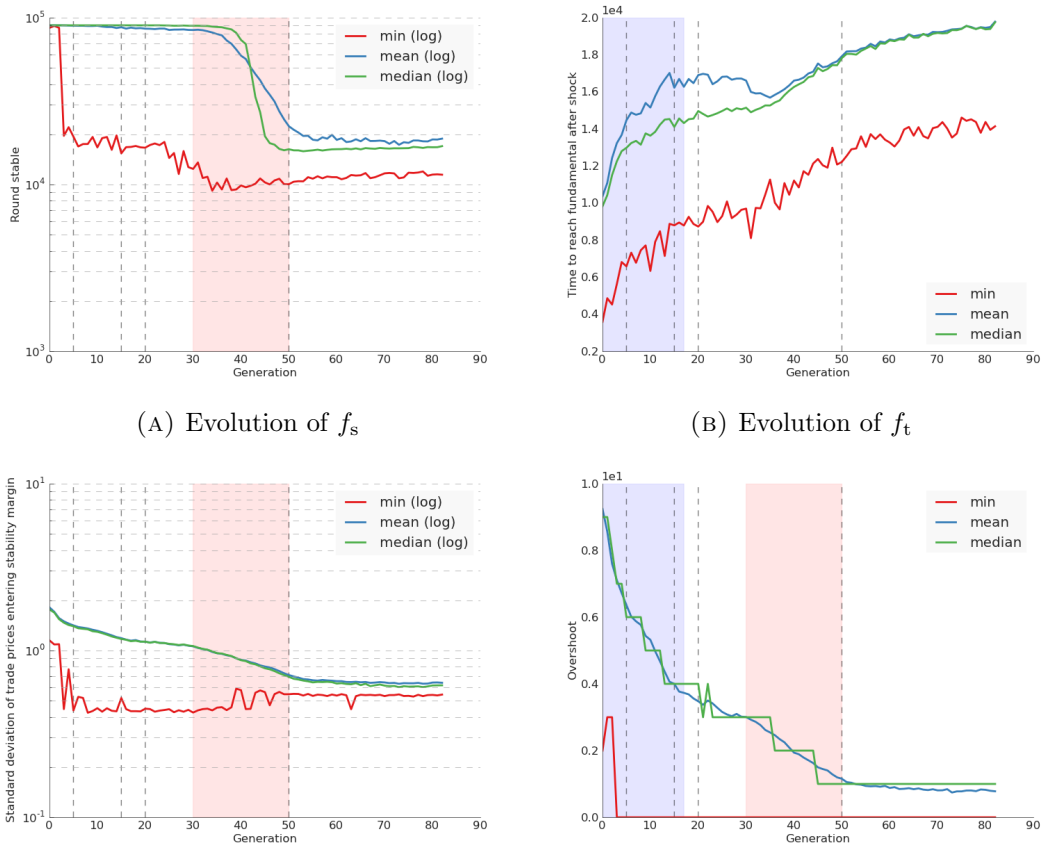
(simulations with a small overshoot also tend to have more stable trade prices), and there work together towards selecting the same type of simulations. $f_t$and $f_s$both The same is not the case for $f_t$and f_s, as it possible that a simulation responds quickly to the shock, but does not stay within the stability margin.

3.6 Fitness and parameter evolution

3.6.1 Variable number of market makers

Figure 3.5 shows the evolution of the four fitness measures. The population wide mean is plotted along the median and minimum statistics. Since all four fitness measures were minimized, the curve for the minimum value shows the best individual alive during each generation, with respect to each fitness measure. While the mean reflects how the overall population is evolving, the median is useful as it gives an insight into how skewed the population wide distribution of parameters is.

Model stability Figure 3.5a: shows that the GA quickly manages to find some parameters which cause the simulation to stabilize quickly. However, these individuals do not manage to dominate the population evident by the mean and median curves remaining almost the same until generation 30 or so. In the next 20 generations the population undergoes a rapid change, as the population wide average of $f_s$drop from close to 10^5 to around $2 \cdot 10^4$ rounds on average. The disparity between the mean and the median indicates that the population undergoes a rapid change in the same period, from mostly containing unstable individuals to mostly containing stable individuals. In generation 42, the median curve crosses the mean curve,



(A) Evolution of f_s

(B) Evolution of f_t

(C) Evolution of f_σ

(D) Evolution of f_o

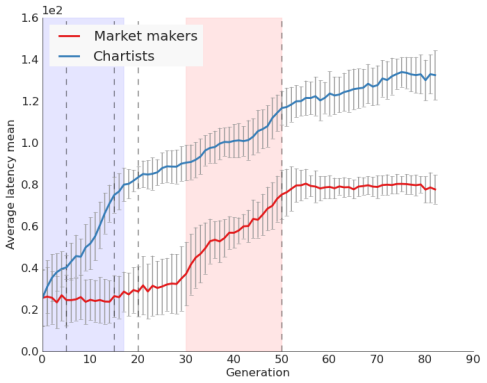
FIGURE 3.5: Evolution of the four fitness measures in experiment d10

which means that the the population contain as many stable simulations as it contain unstable simulations. From that point on the unstable simulations are quickly replaced by stable individuals.

Price fluctuations and overshoot During the same period, the population average $f_\sigma$also decreases fairly rapidly, but the drop is less pronounced than the drop in f_σ. As figures 3.6a and 3.6b show, the number of market makers rapidly increased during this period, as did the average latency of the market makers. Since the mean and median are close in both figure[3], the mean is representative of the evolution of the entire population.

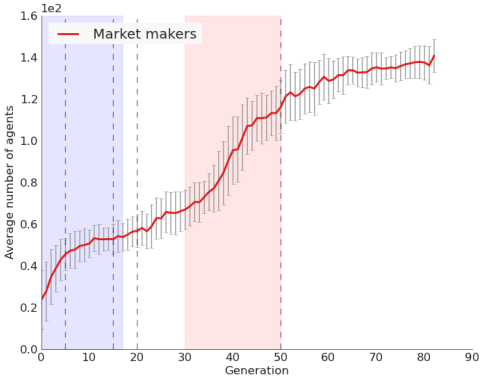
Responsiveness $f_t$measures the time it takes for the model to react to the shock in the fundamental, and the evolution of the population wide statistics is shown on figure 3.5b. Although the GA is instructed to minimize $f_t$in order to look for more faster models, it clearly fails to do this. Indeed, the most responsive simulation took only about 4000 rounds to reach the new fundamental, but this individual

[3]Since $f_o$is discrete, the median and min statistics are also discrete



(A) Evolution of $\lambda_{m,\mu}$ and $\lambda_{c,\mu}$

(B) Evolution of $\lambda_{m,\sigma}$ and $\lambda_{c,\sigma}$



(C) Evolution of N_m

FIGURE 3.6: Evolution of the model parameters in experiment d10

died out in favor of slower individuals. In the last generation the most responsive simulation took around 14000 rounds to reach the new fundamental. The reasons for this failure to locate responsive models is discussed in section 3.5.1. In the A large change of the average of f_{t}happens in the rounds five to 15. In this period, the median is lower than the mean, which means that the growth in the mean can be attributed to a minority of individuals.

On figures 3.5 and 3.6, the two areas shaded in a light blue and light red respectively show the two periods during which there was a drastic change in parameters and fitness-values. By comparing the time at which parameters and fitness-values change, it is possible to get an idea of how parameters influence the fitness-values. To that end, figure 3.6 shows the evolution of each of the parameters that were varied by the GA[4].

The two periods indicated by the shaded squares seem to reflect some sudden changes in the parameters.

[4]Since the median was found to follow the mean nicely for all the parameters, the medians are not displayed. Also, the gray error bars show the population wide variance

Average agent latency As is shown on figure 3.6a, individuals containing large latency parameters are selected for both HFT market makers and HFT chartists. $\mathrm{E_p}[\lambda_{c,\mu}]$ grows grows quickly during the first 20 rounds (blue shade). Referring back to figure 3.5, it is seen that $\mathrm{E_p}[f_\mathrm{t}]$ and $\mathrm{E_p}[f_\mathrm{o}]$ grows and shrinks respectively. As for $\mathrm{E_p}[\lambda_{m,\mu}]$, it grows from rounds 20 through 50 (red shade), and this seems to be strongly reflected in the growth of $\mathrm{E_p}[f_\mathrm{s}]$, and to a lesser degree a decline in $\mathrm{E_p}[f_\mathrm{o}]$ and $\mathrm{E_p}[f_\sigma]$. Furthermore, the small size of the error bars on both curves show that the population consistently moves towards containing more individuals with larger latency parameters for both HFT agent types. While initially $\mathrm{E_p}[\lambda_{m,\mu}] \approx \mathrm{E_p}[\lambda_{c,\mu}]$, the population wide mean $\mathrm{E_p}[\lambda_{c,\mu}]$ ends up being roughly 1.5 times larger than $\mathrm{E_p}[\lambda_{c,\mu}]$. Finally, note also that the growth of $\mathrm{E_p}[\lambda_{c,\mu}]$ and $\mathrm{E_p}[\lambda_{m,\mu}]$ seem to be somewhat independent, as they sometimes grow together, sometimes not.

Number of market makers The number of market makers increases almost every generation, but grows especially quickly through rounds 20 to 50 (red shade)

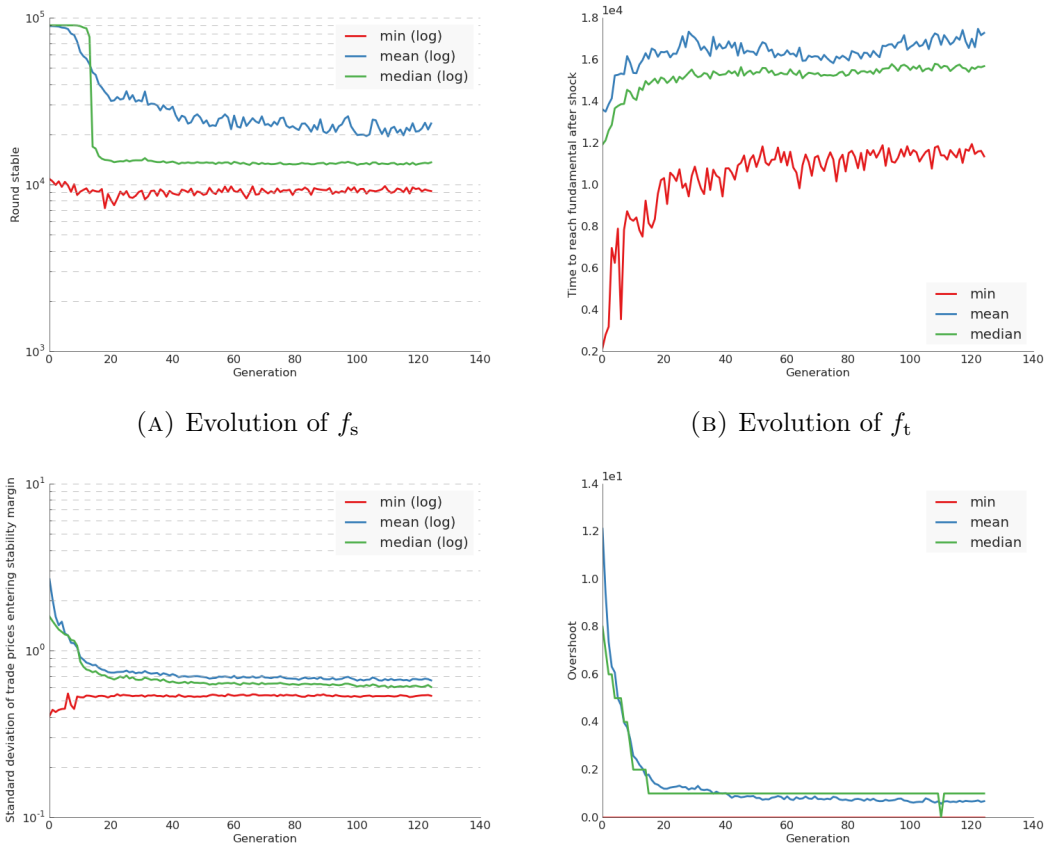
Agent latency variance Figure 3.6b: The trends for $\mathrm{E_p}[\lambda_{c,\sigma}]$ and $\mathrm{E_p}[\lambda_{m,\sigma}]$ are less clear, as the population-wide variances $\mathrm{Var_p}[\lambda_{c,\mu}]$ and $\lambda_{m,\mu}$ illustrated by the large error bars are high compared to the change in $\mathrm{E_p}[\lambda_{c,\mu}]$ and $\mathrm{E_p}[\lambda_{m,\mu}]$. While this could mean that the simulation behaves more nicely when the difference between the latency parameters of the trading agents is smaller, further experiments would have to be carried out to confirm this fact. XXX

In summary, the genetic algorithm prefers simulations with many, but relatively slow market makers. Apparently simulations with slow chartists also outperformed those with fast chartists, but since the number of HFT chartists was fixed at $N_c =$, this experiment does not reveal how the simulation would perform with more (or less) chartists. It is possible to imagine that the market would perform just as well with a few and fast chartists. Section **??** contains the analysis of an experiment in which the number of chartists were varied. The discussion above can be summarized as follows:

1. The responsiveness of the market is influenced by latency of the chartists. Slower chartists made the market require more time to respond to the fundamental shock.

2. The time it takes for the market to become in influenced by the number of market makers and on the latency of the market makers. More but slower market makers seems to make the market settle within the stability margin faster.

3. The overshoot, as well as the average size of the price fluctuations of the market, are both influenced by the latency of both agent types, as well as the number of market makers.



(A) Evolution of f_{s}

(B) Evolution of f_{t}

(C) Evolution of f_{σ}

(D) Evolution of f_{o}

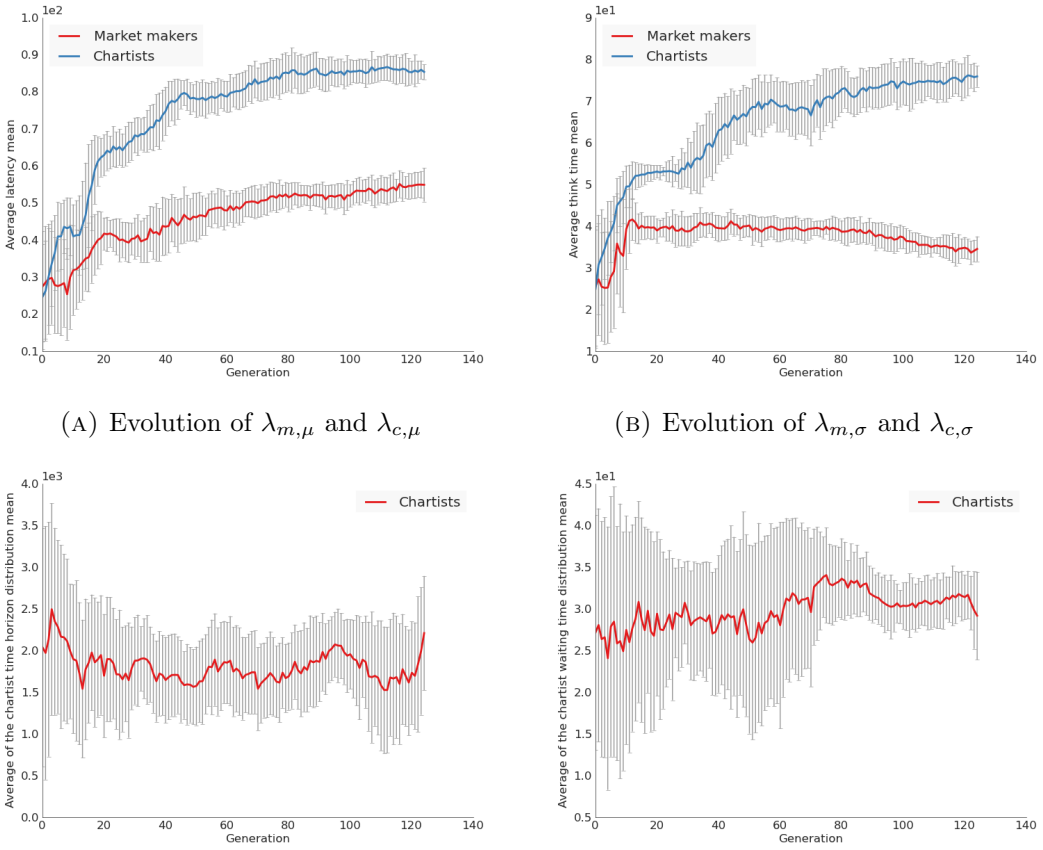
FIGURE 3.7: Evolution of the four fitness measures in experiment d9

4. The market was more stable but reacted slowly when the chartists were slower than the market makers.

The accuracy of the above analysis is limited as it only looks at population wide statistics at a given point in the duration of the GA. The following section contain an analysis in which the generation to which each data point belongs is considered irrelevant. The analysis will try to confirm each of the four statements above.

3.6.2 Fixed number of chartists and market makers

When the GA cannot change the number of chartists and market makers, it has to find better fitness values by selecting the right latency parameters. As shown on figure 3.7, the GA managed to find models with little or no overshoot, non-flickering prices, and which become stable. The price of having these nice qualities seems to be a slower response time to the shock. The GA find these well-behaving models by selecting latency parameters such that the chartists are slower than the market makers. $\mathrm{E_p}[H_{c,\mu}]$ and



(A) Evolution of $\lambda_{m,\mu}$ and $\lambda_{c,\mu}$

(B) Evolution of $\lambda_{m,\sigma}$ and $\lambda_{c,\sigma}$

(C) Evolution of N_m

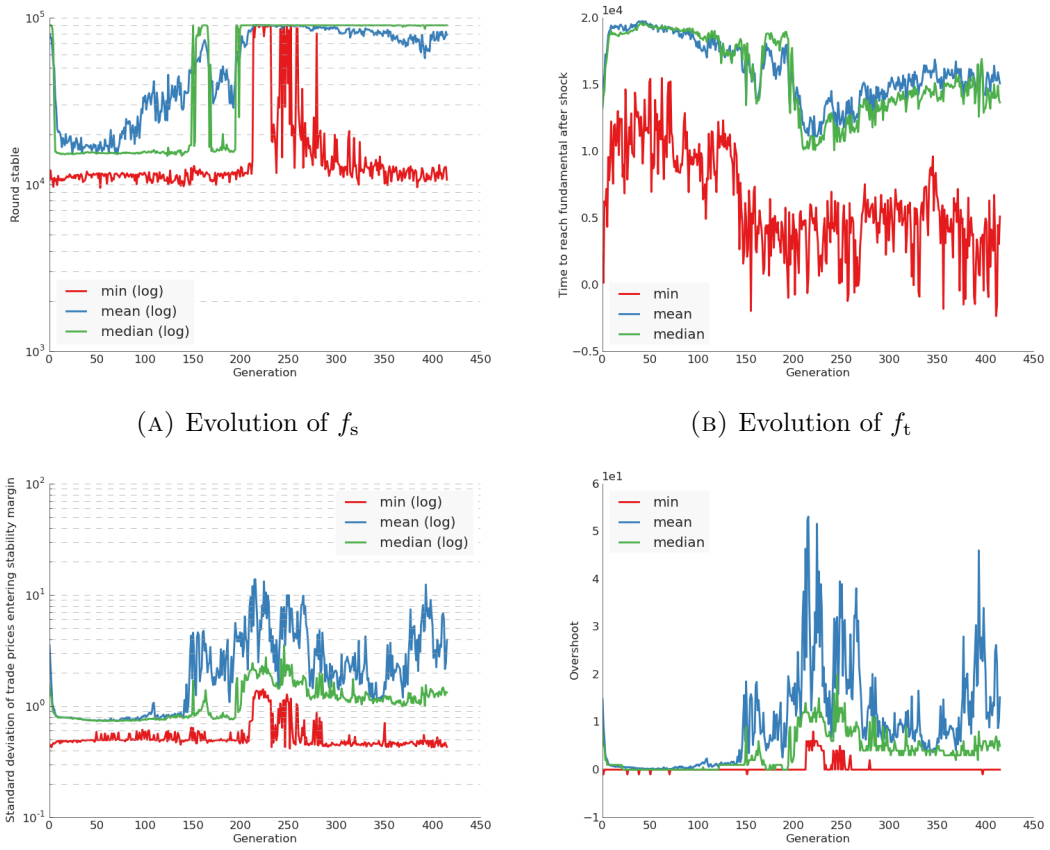
(D) Evolution of N_m

FIGURE 3.8: Evolution of the model parameters in experiment d10

$\mathrm{E_p}[W_{c,\mu}]$ change little over the are more or less unchanged, which seems to indicate that they have little effect of the fitness values, at least compared to other time related parameters such as $\lambda_{c,\mu}$, $\lambda_{m,\mu}$, $T_{c,\mu}$and $T_{m,\mu}$. This can either mean that these parameters does

3.6.3 Variable number of chartists

The evolution of the fitness values in d11, shown in figure 3.9 fluctuate significantly more than in d9 and d10. In contrary two the two other experiments, the GA here manages to decrease f_t, and $\mathrm{E_p}[f_\mathrm{t}]$ drops almost 10000 rounds around generation 200. At the same time $\mathrm{E_p}[f_\mathrm{o}]$, $\mathrm{E_p}[f_\sigma]$ and $\mathrm{E_p}[f_\mathrm{s}]$ all rise, again indicating the there exists a trade-off between speed and stability in the model. At the point in the evolution where $\mathrm{E_p}[f_\mathrm{t}]$ drops, several interesting things happen with the model parameters that live in the population. First of all the number of chartists increase dramatically, again pointing towards more chartists making the markets fast and unstable. Secondly, the market maker latency also drops to around the same level as the chartist latency. This



(A) Evolution of f_s

(B) Evolution of f_t

(C) Evolution of f_σ

(D) Evolution of f_o

FIGURE 3.9: Evolution of the four fitness measures in experiment d11

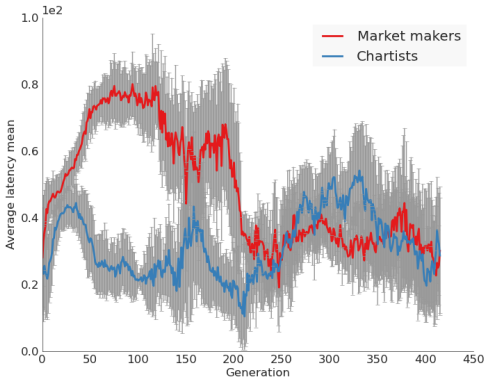
is interesting because it could mean that the faster market makers help drive the market towards a larger overshoot.

Market makers become slower and the chartists become faster. At the same time, the number of chartists rise rapidly

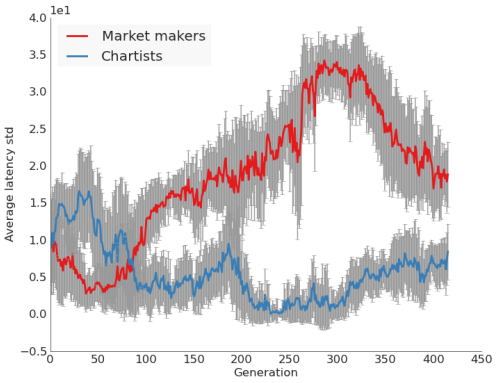
- A high number of market makers enable the market to respond quickly to the shock, but also cause the traded price to flicker more, and for the model to have a larger overshoot.
- Slower market makers also cause the market to respond faster to the shock

3.7 Parameter and fitness correlation

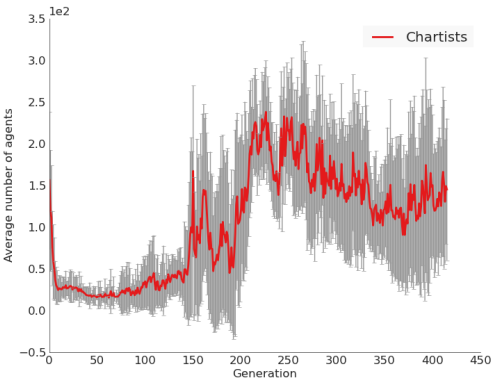
This section contains several figures which illustrate how the model fitness varies with the model parameters. In the figures showing the data from experiment d11, simulations with $f_o > 10$ are removed in order to make the figures easier to interpret.



(A) Evolution of $\lambda_{m,\mu}$ and $\lambda_{c,\mu}$



(B) Evolution of $\lambda_{m,\sigma}$ and $\lambda_{c,\sigma}$

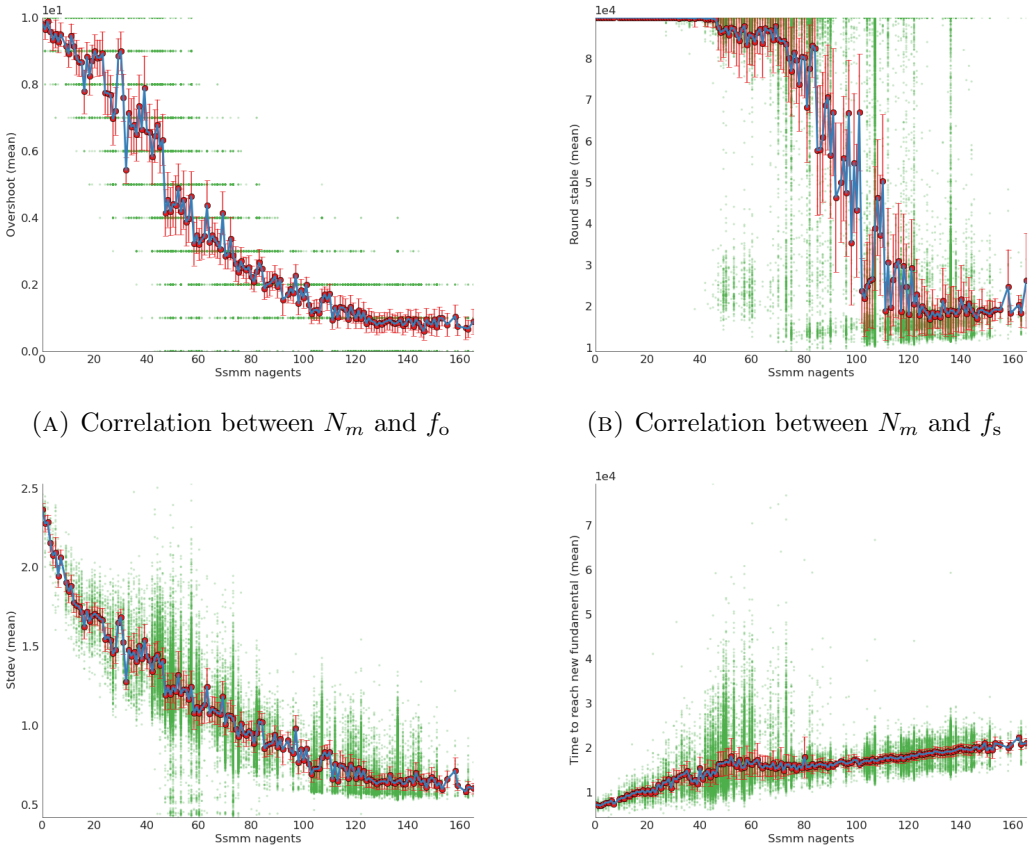


(C) Evolution of N_c

FIGURE 3.10: Evolution of the model parameters in experiment d11

3.7.1 Number of market makers

Figure 3.11 shows how the number of market makers correlates with the model fitness in experiment d10. It is clear that a large number of market makers reduces the overshoot, whereas the market virtually always have an overshoot when there are little or no market makers. The same is true for the trade prices flicker: few market makers always means flickering prices. A higher number of market makers makes the market less responsive, while fewer market makers makes the market more responsive. Finally, the number of market makers also influences how quickly the market settles within the stability margin, as markets with more market makers become stable faster than markets with few agents.



(A) Correlation between N_m and f_{o}

(B) Correlation between N_m and f_{s}

(C) Correlation between N_m and f_σ

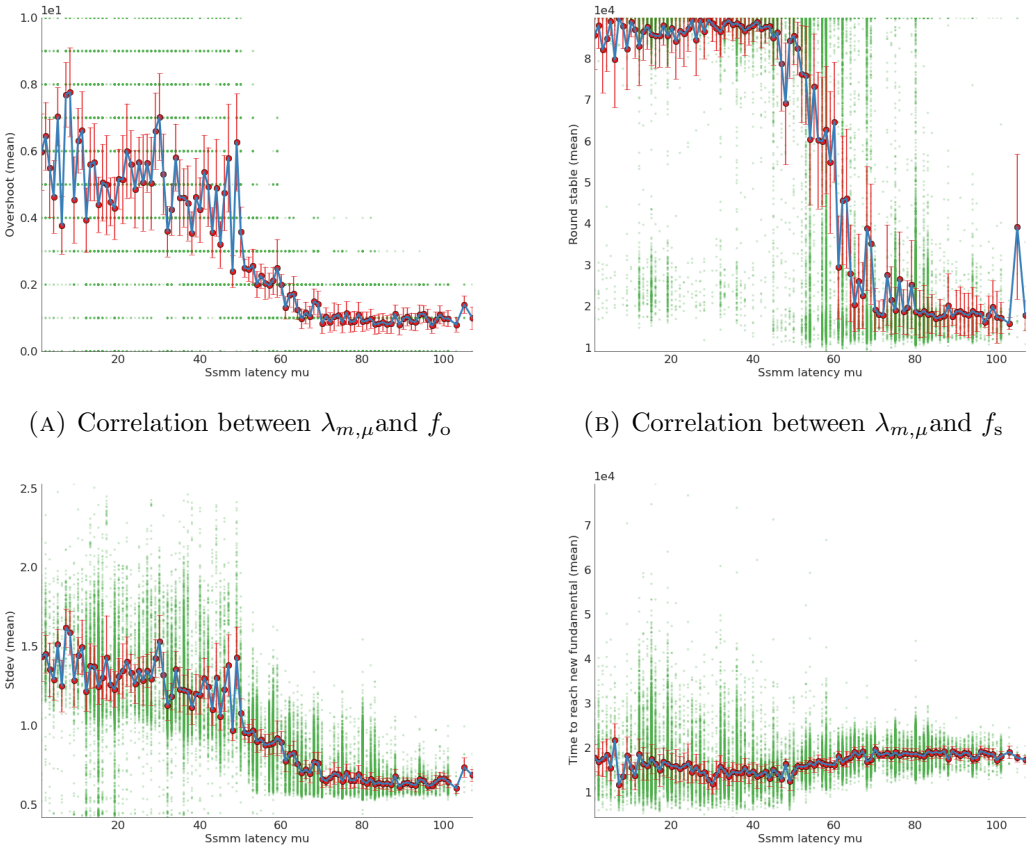
(D) Correlation between N_m and f_{t}

FIGURE 3.11: Correlation between $N_m$and the four fitness measures in experiment d10

3.7.2 Market maker latency

Fixing the number of chartists

In experiment d10, the number of chartists was kept fixed at $N_c = 150$, while $N_m$was varied by the GA. In this case, the $\lambda_{m,\mu}$is found to be somewhat correlated with the fitness measures as illustrated on figure 3.12. Especially for $\lambda_{m,\mu} > 50$, the data seems to be consistent, as the error bars showing the standard deviation of the data are small in this region. However, for $\lambda_{m,\mu} < 50$, the model behavior is no longer predictable by using $\lambda_{m,\mu}$alone. Figure 3.13 sheds light on the reason as to why this is, as it shows that even though the market maker latency does influence the market, the effect is secondary to that of the number of agents. For instance, when the market contains less than 25 market makers, all four fitness measures are more or less unchanged shown by the nearly flat red curves. As the number of market makers grow, so does the importance of how fast they are. The average overshoot and the average time to catch up to the new fundamental only change slighly, even with over 100 market makers (yellow line). On the other hand, the average price flickering and the average number of rounds it takes



(A) Correlation between $\lambda_{m,\mu}$and f_{o}

(B) Correlation between $\lambda_{m,\mu}$and f_{s}

(C) Correlation between $\lambda_{m,\mu}$and f_{σ}

(D) Correlation between $\lambda_{m,\mu}$and f_{t}

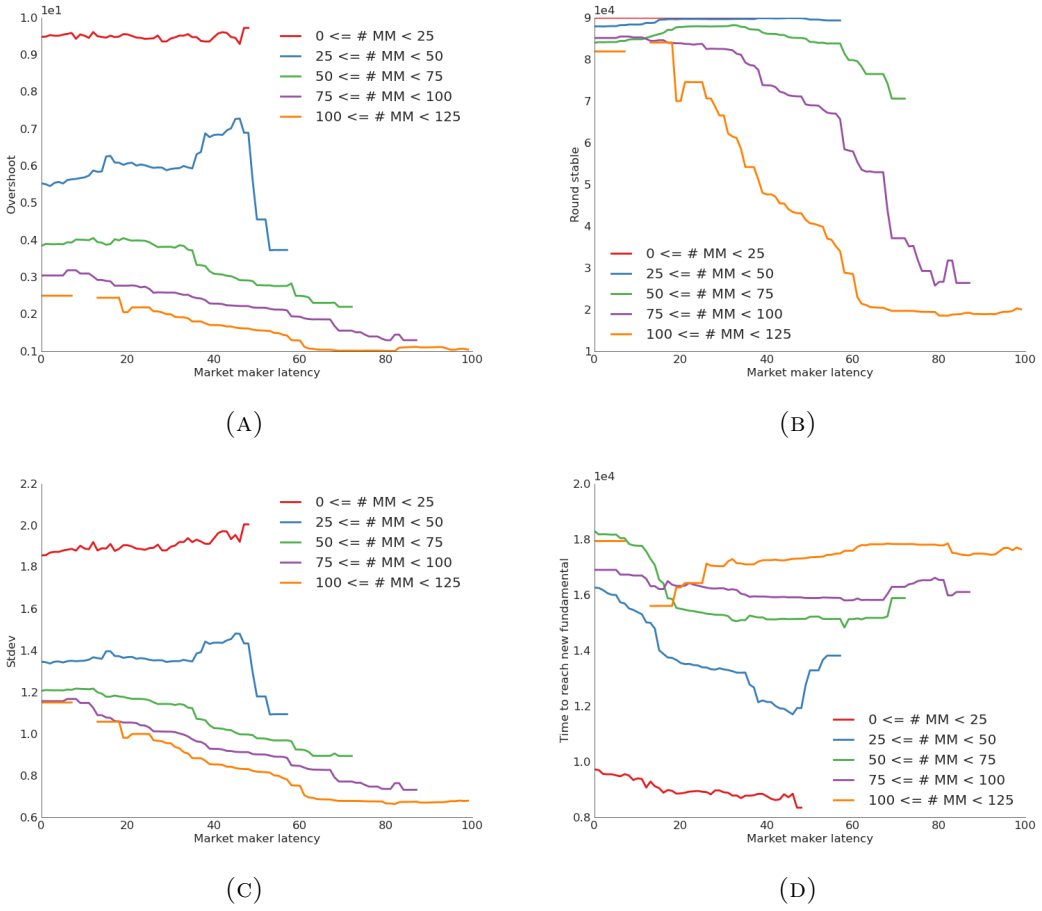
FIGURE 3.12: Correlation between $\lambda_{m,\mu}$ and fitness values (fixed N_c, variable N_m)

for the market to settle within the stability margin both change significantly with the market maker speed for $N_m > 50$.

In summary, figure 3.13 shows that in a market with only a few market makers, these agents have little influence no matter how fast they are. As the number of market makers grow, so does the collective force of all the market makers, and so does the importance of how slow or fast these agents are. The next section will examine how the market behaves with respect to how many chartists are active in the market, and with respect to the latency of the chartists.

3.7.2.1 Fixing the number of market makers

XXXNOT FINISHEDXXX In experiment d11, the number of chartists was varied, while the number of market makers were kept at a constant $N_m = 52$ agents. While it was fairly obvious that the market would not be impacted by changing $\lambda_{m,\mu}$when only a few market makers were active, it is less obvious that the same is true for the number of chartists. Yet figure



(A) (B)

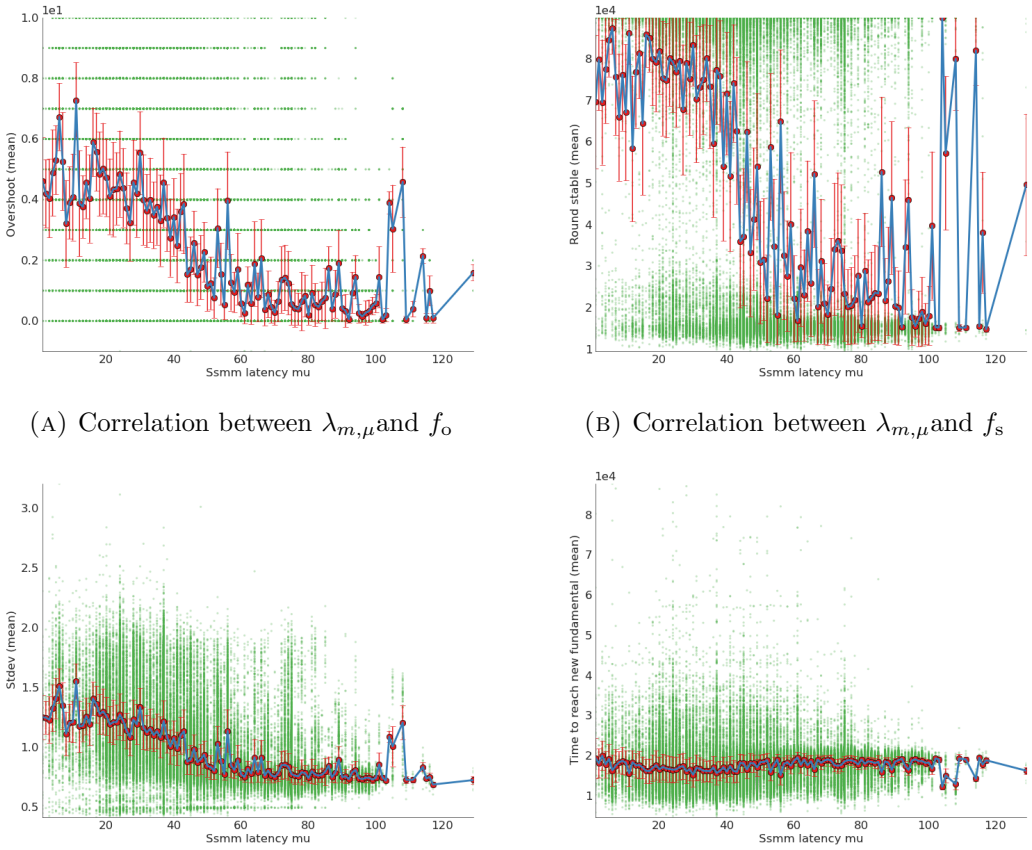
(C) (D)

FIGURE 3.13: Relation between N_m, $\lambda_{m,\mu}$, and the model fitness when the number of chartists was fixed to $N_c = 150 agents$. Due to missing data, some of the curves are not complete.

3.7.3 Number of chartists

Figure 3.16 shows the average population wide number of agents $\mathrm{E_p}[N_c]$ plotted against each of the four fitness measures, and the figures are summarized below.

- The more chartists a market has, the faster it responds to the fundamental. This is especially true when comparing markets with less than 100 chartists, and less pronounced when comparing markets with over 100 chartists.
- The model overshoot is also correlated with the number of chartists in such a way that markets with more chartists have a larger overshoot on average. Whereas the market only seemed to benefit from a decreased response time when the number of chartists were kept below 100, the overshoot continues to grow steadily larger even as the number of chartists is increased beyond 100 agents.
- $f_\sigma$is correlated with the number of chartists in the same way as f_o, such that more chartists make the traded prices flicker more.



(A) Correlation between $\lambda_{m,\mu}$and f_{o}

(B) Correlation between $\lambda_{m,\mu}$and f_{s}

(C) Correlation between $\lambda_{m,\mu}$and f_{σ}

(D) Correlation between $\lambda_{m,\mu}$and f_{t}

FIGURE 3.14: Correlation between $\lambda_{m,\mu}$ and fitness values (fixed N_m, variable N_c)

- Finally, the graph for f_{s}show that the market rarely becomes stable when it contains more than 50 chartists or so.

The large errorbars around the points with a large value of $N_c$is caused by data sparsity in this region. The GA was set to search for stable markets, and since markets with a large number of chartists tend to be unstable, such markets were rarely selected for creating offspring.

3.7.4 Chartist latency

Fixed number of chartists

Figure 3.17a shows that $\lambda_{c,\mu}$is negatively correlated with f_{o}, such that markets with faster chartists are more likely to have a larger overshoot. Next, figure 3.17 shows that $\lambda_{c,\mu}$is negatively correlated with f_{σ}, such that markets with faster chartists are more likely to have flickering trade prices.

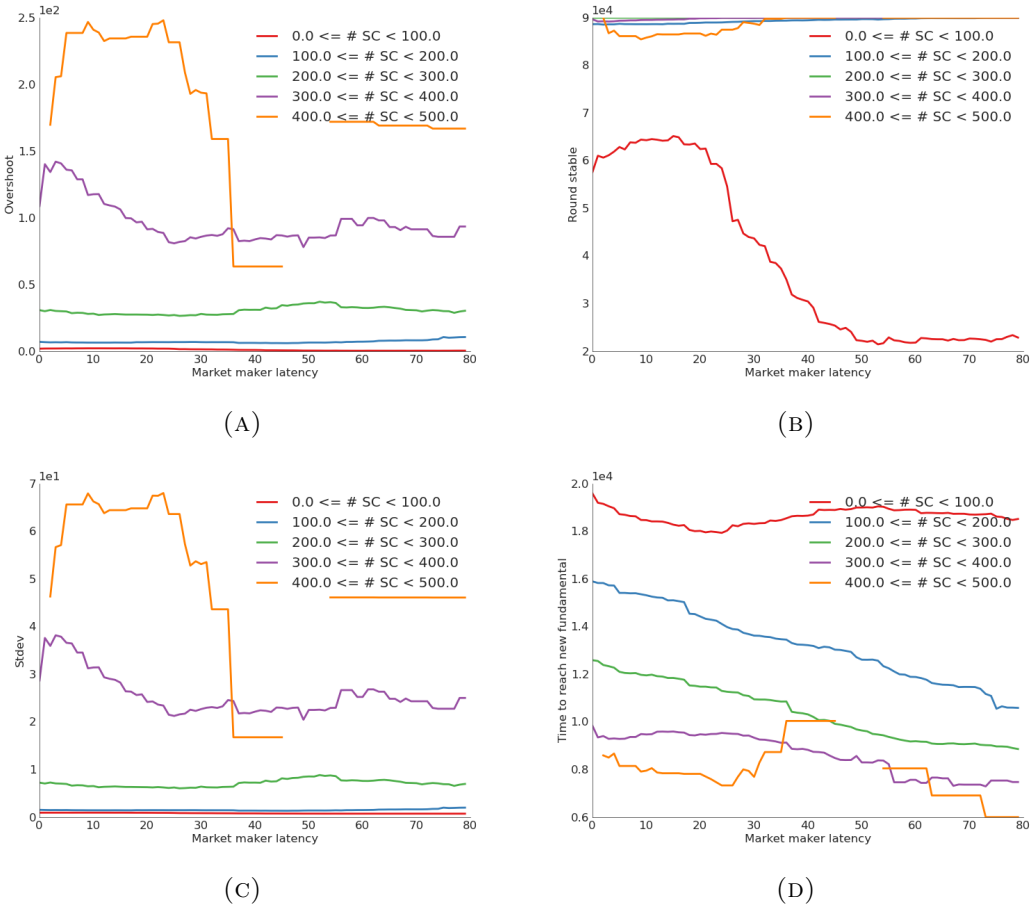
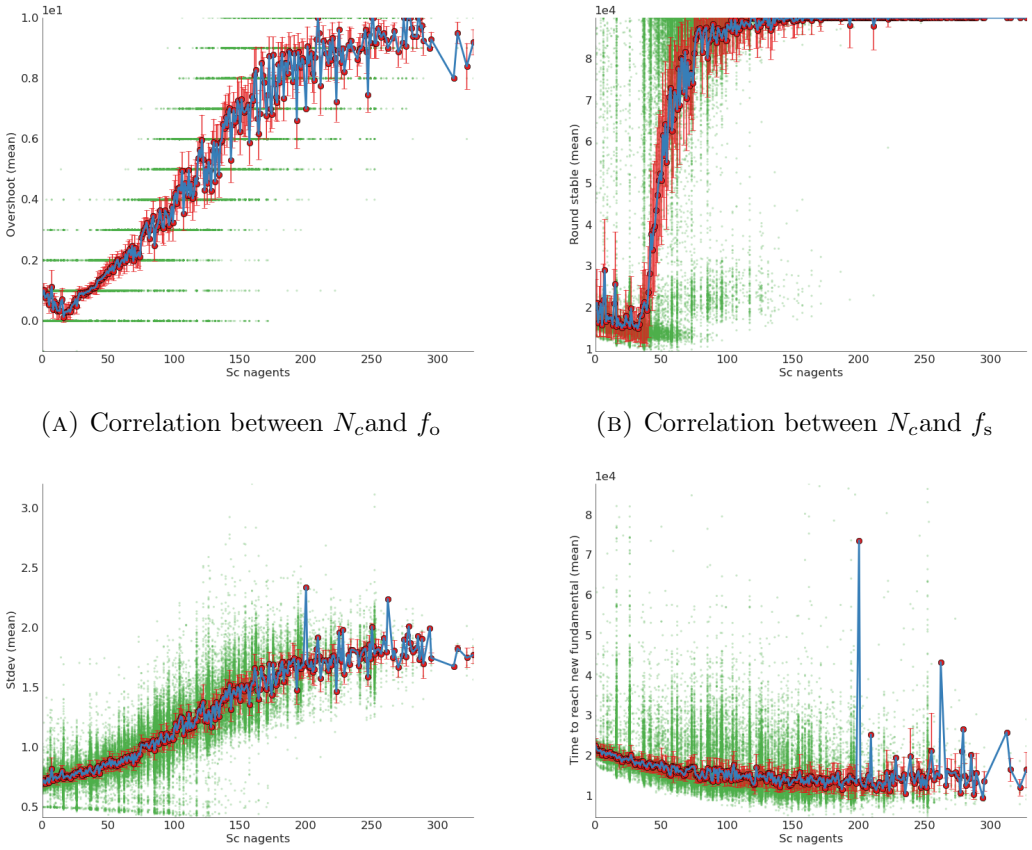


FIGURE 3.15: Relation between N_m, $\lambda_{m,\mu}$, and the model fitness when the number of market makers was fixed to $N_m = 52$ agents.

As for the market responsiveness, it is seen that $\lambda_{c,\mu}$is positively correlated with f_{t}, such that markets with faster agents is more likely to have a shorter response time to the market. Figure 3.17d confirms that markets with fast chartists did actually manage to reach the new fundamental price faster than those markets having slow chartists. The average response time of markets in which the chartists had a latency of less than 30 rounds was around 18000 rounds, whereas it was around 25000 rounds with chartists with more than 100 rounds of latency. The market response time is most sensitive in the range $20 < \lambda_{c,\mu} < 60$, and does not change much for larger latencies.

The plots of f_{o}, $f_{\sigma}$and f_{t}show that predicting the three fitness measures in markets with slow chartists would be more accurate than for markets with fast chartists, as the correlation of f_{o}, $f_{\sigma}$and $f_{\mathrm{t}}$with $\lambda_{c,\mu}$is stronger for larger values of $\lambda_{c,\mu}$.

Figure 3.17b shows that $\lambda_{c,\mu}$is positively correlated with f_{s}, but also that the relationship between $\lambda_{c,\mu}$and f_{s}seems highly non-linear. The figure illustrates the binary nature of the stability criteria, that is, that a simulation is either stable or not stable. This causes $f_{\mathrm{s}}$to have a high conditional variance of f_{s}given $\lambda_{c,\mu}$in the region $50 < \lambda_{c,\mu} < 120$,



(A) Correlation between $N_c$and f_o

(B) Correlation between $N_c$and f_s

(C) Correlation between $N_c$and f_σ

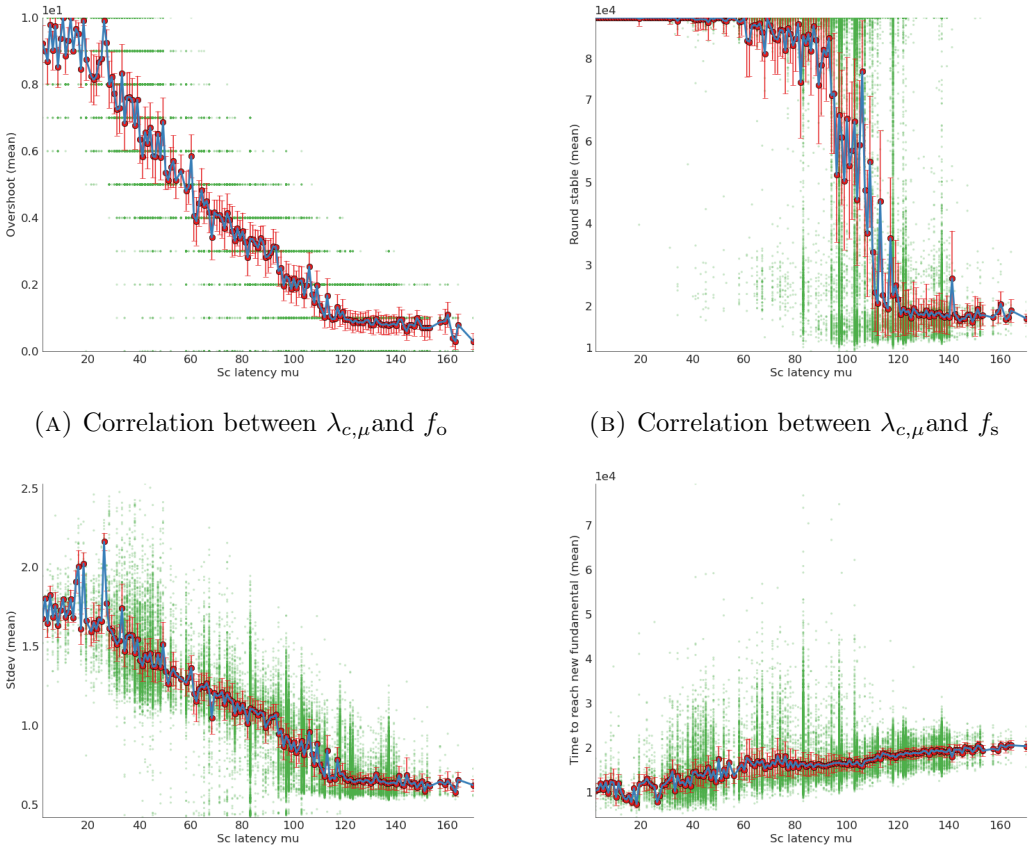
(D) Correlation between $N_c$and f_t

FIGURE 3.16: Correlation between $N_c$and the four fitness measures when $N_m = 52$ (experiment d11)

meaning that that prediction of $f_s$from $\lambda_{c,\mu}$in this region would not be very accurate. What this means is that the stability of a simulation is highly dependent on factors other than $\lambda_{c,\mu}$, when the parameter is within 50 to 120 rounds. When the chartists are faster than 50 rounds, the market is almost always unstable, and when the chartists are slower than 120 rounds the market is almost always stable.

Fixed number of market makers

Figures 3.19 seems to indicate that no correlations exist between the speed of the chartist agents, and the model fitness measures. However, since figure 3.17 does point towards the existence of such correlations, something else must be obscuring the scatter plots in 3.19. The reason is found to be that the number of chartists was not kept constant in experiment d11. It turns out that $\lambda_{c,\mu}$ is in fact correlated with f_o, $f_\sigma$and f_t, but that the correlation depends heavily on the number of chartists in the market.



(A) Correlation between $\lambda_{c,\mu}$and f_{o}

(B) Correlation between $\lambda_{c,\mu}$and f_{s}

(C) Correlation between $\lambda_{c,\mu}$and f_{σ}

(D) Correlation between $\lambda_{c,\mu}$and f_{t}

FIGURE 3.17: Correlation between chartist latency and fitness values (fixed N_c, variable N_m)

3.7.5 Chartist to market maker ratio

The above observations about how the number of agents influence the stability and speed of the market pointed out that more market makers made the market slow but stable, while more chartists made the market fast, but unstable. By merging $\boldsymbol{P}$d10 and $\boldsymbol{P}$d11, we can calculate the ratio, ρ_{A}, between the number of chartists and the number of market makers, and see how the fitness values correlate. Figure **??** shows the resulting scatter plots.

3.8 Grouping models by behavior

This section is be concerned with trying to tie various patterns of model behavior to different regions in the parameter space. The quickest way to get an idea of how the data generated by the simulations is distributed is to make scatter plots. The data of the model fitness is four-dimensional, requiring twelve plots to visualize all combinations.

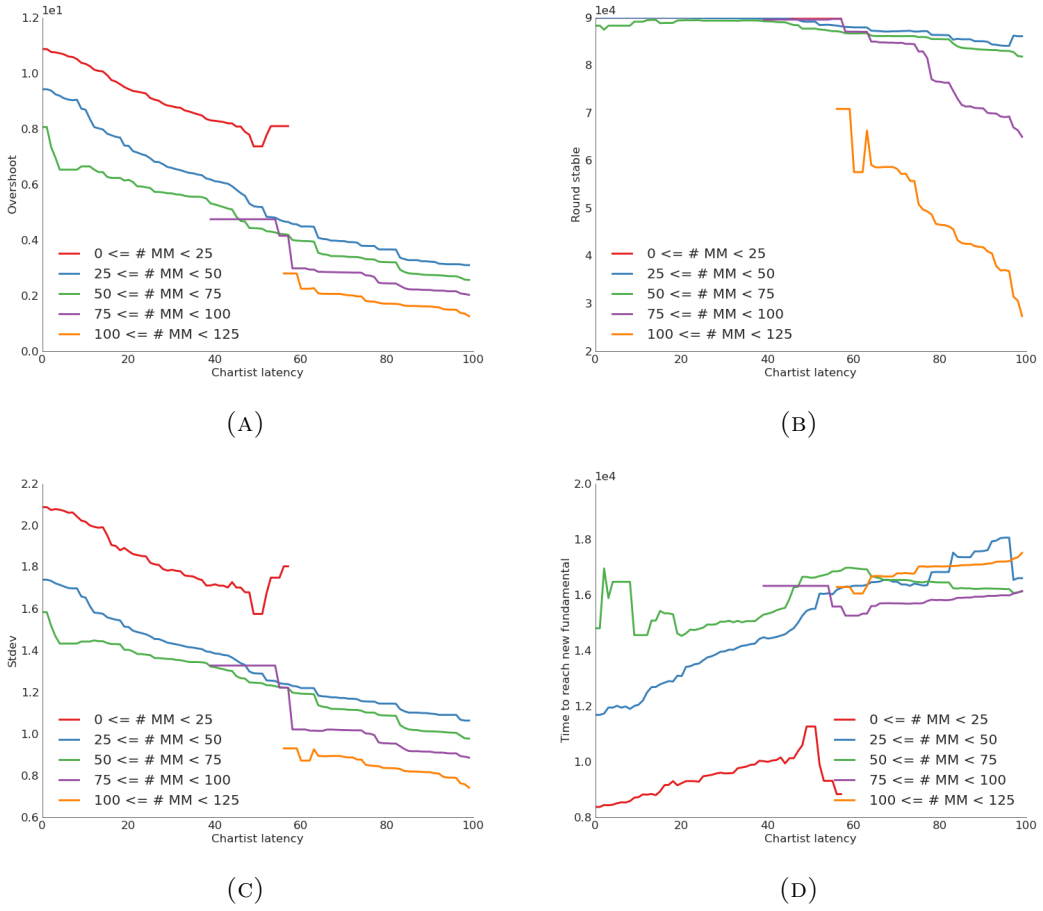
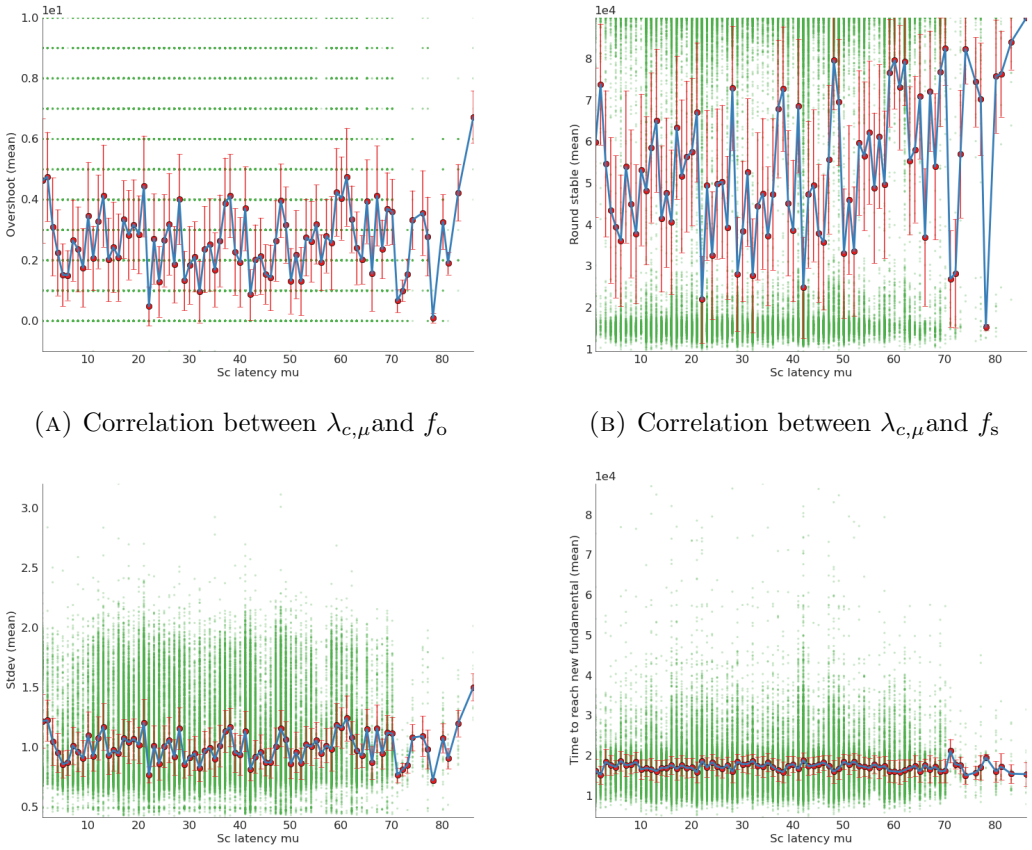


FIGURE 3.18: Relation between N_m, $\lambda_{c,\mu}$, and the model fitness when the number of chartists was fixed to $N_c = 150 agents$. Due to missing data, some of the curves are not complete.

However, since $f_{\mathrm{o}}$is discrete with a small range of values, it is not suitable for a scatter plot. Furthermore, some scatter plots are not useful for interpretation if they do not show any structure in the data. Figure 3.23 shows three scatter plots which were found to best illustrate the structure of the dataset from d9. Note also that coloring each point corresponding to its value in one dimension makes it possible to show how the data is distributed in three dimensions. The scatter plots do seem to reveal some structure, the presence of large values in the f_{σ}feature obscures the nature of this structure, in spite of the logarithmic scaling. The plot showing $\log f_{\sigma}$ vs. $\log f_{\mathrm{s}}$ is squeezed to the left, and the color grading on the scatter plot for $\log f_{\mathrm{o}}$ vs . $\log f_{\sigma}$ reveals no variety in the f_{σ}feature. In an attempt to get some more information out of the scatter plot, data points with an overshoot of over 100 % of the shock to the fundamental (corresponding to $f_{\mathrm{o}} > 10$) are removed. The resulting scatter plots for the reduced data set are shown on figures 3.24 and **??**.

First of all, it is seen that while the data is distributed similarly in the three data sets, there are some differences. The data from d9 seems to have many “lonely” data points,



(A) Correlation between $\lambda_{c,\mu}$and f_{o}

(B) Correlation between $\lambda_{c,\mu}$and f_{s}

(C) Correlation between $\lambda_{c,\mu}$and f_{σ}

(D) Correlation between $\lambda_{c,\mu}$and f_{t}

FIGURE 3.19: Correlation between chartist latency and fitness values (fixed N_m, variable N_c)

which are not part of any cluster, whereas the data from d10 somehow seems to be the cleanest of the three. In all three data sets, there are clusters of data. The clusters do not necessarily mean anything in themselves. They might simply be due to the way that data points are mutated and crossed by the GA. However, by considering which regions of the fitness space that each cluster covers, it is possible to add meaning to the clusters in terms of model behavior.

3.8.1 Manually grouping simulations by behavior

Table 3.6 contains an overview of the named criteria used for roughly grouping simulations into different types of behavior. The following text contains the reasoning for why each of these groups are interesting.

In figure 3.24, the black dashed lines at $f_{\mathrm{t}} = f_{\mathrm{s}}$ divide each plot into region A, (upper left triangle) and region B (lower right triangle). Region A contains the fitness-points of

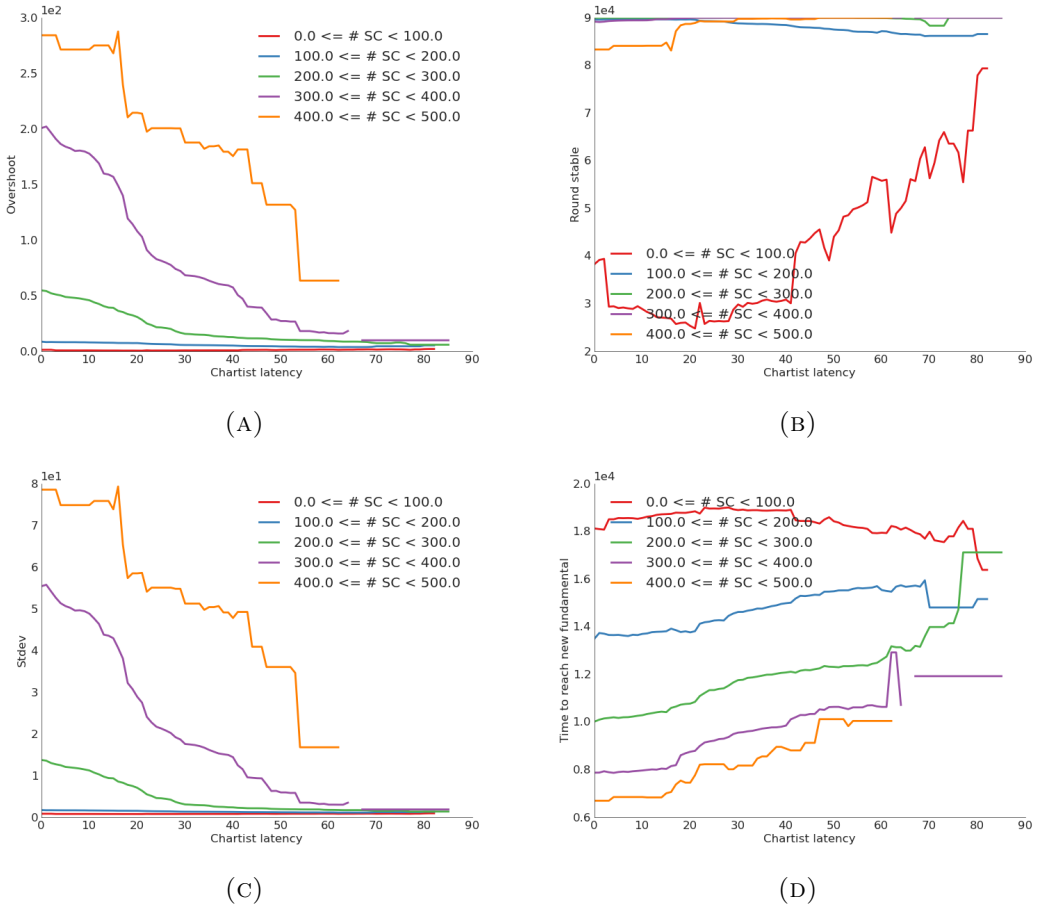


FIGURE 3.20: Relation between N_c, $\lambda_{c,\mu}$, and the model fitness when the number of market makers was fixed to $N_m = 52$ agents.

the simulations which are counted as stable *after* they reach the new fundamental, and region B contain those that become stable before.

Fast and stable simulations with flickering prices

The points in the lower left corner in are those which quickly reached the new fundamental price, and quickly because stable, leading those simulations to be assigned low $f_{\text{t}}$and f_{s}fitness-values. These points are extracted by using filter F1 (see table 3.6)

Slow or fast and stable simulations with non-flickering prices

All three data sets have data points which are close to the diagonal. However, only d9 has data points which are close to the diagonal in the upper right corner of the figure. These points are interesting because they belong to simulations which became stable as soon as they reached the new fundamental price. Hence, these simulations should have

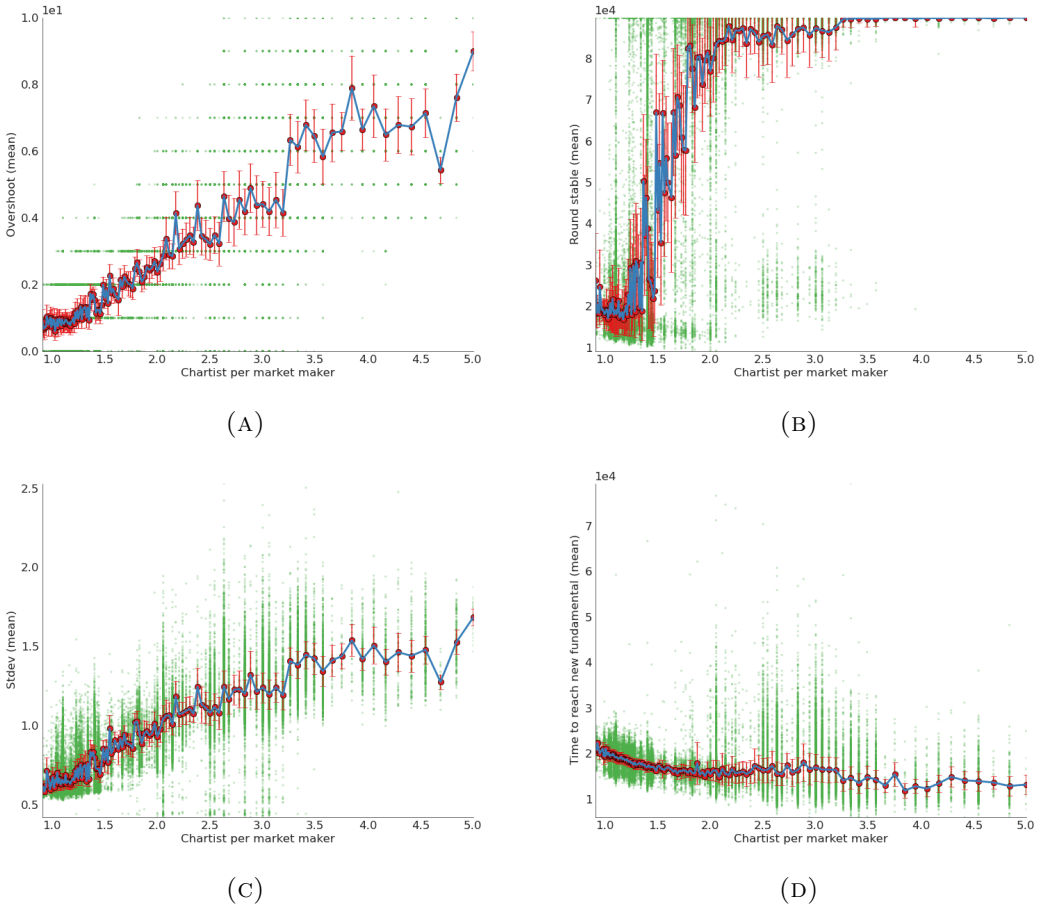


FIGURE 3.21: Correlations between $\rho_{\mathrm{A}}$and the fitness values when $N_c = 150$

prices that do not flicker, and therefore yield a small f_σ-fitness. This is confirmed by looking at the left scatter plot of d9, as all the points close to the dotted line has a green/blue color. The right plot of d9 shows that these simulations did not have any overshoot. It is interesting that both slow simulations which take a long time to reach the new fundamental, as well as simulations who manage to be fast, have no overshoot, and this observation begs the question of whether or not these simulations have some common parameters that make them behave in such a way. These points are extracted by applying filters F2 and F3 (see table 3.6) to the data matrix $\boldsymbol{F}$d9 which selects points that lie within a distance of 400 rounds of the diagonal.

Stable before reaching the new fundamental

Most of the simulations falling in region B, meaning that they became stable before reaching the new fundamental price, had no overshoot. However, when the model was allowed to have a large number of chartists, but in d11, a group of simulations did have

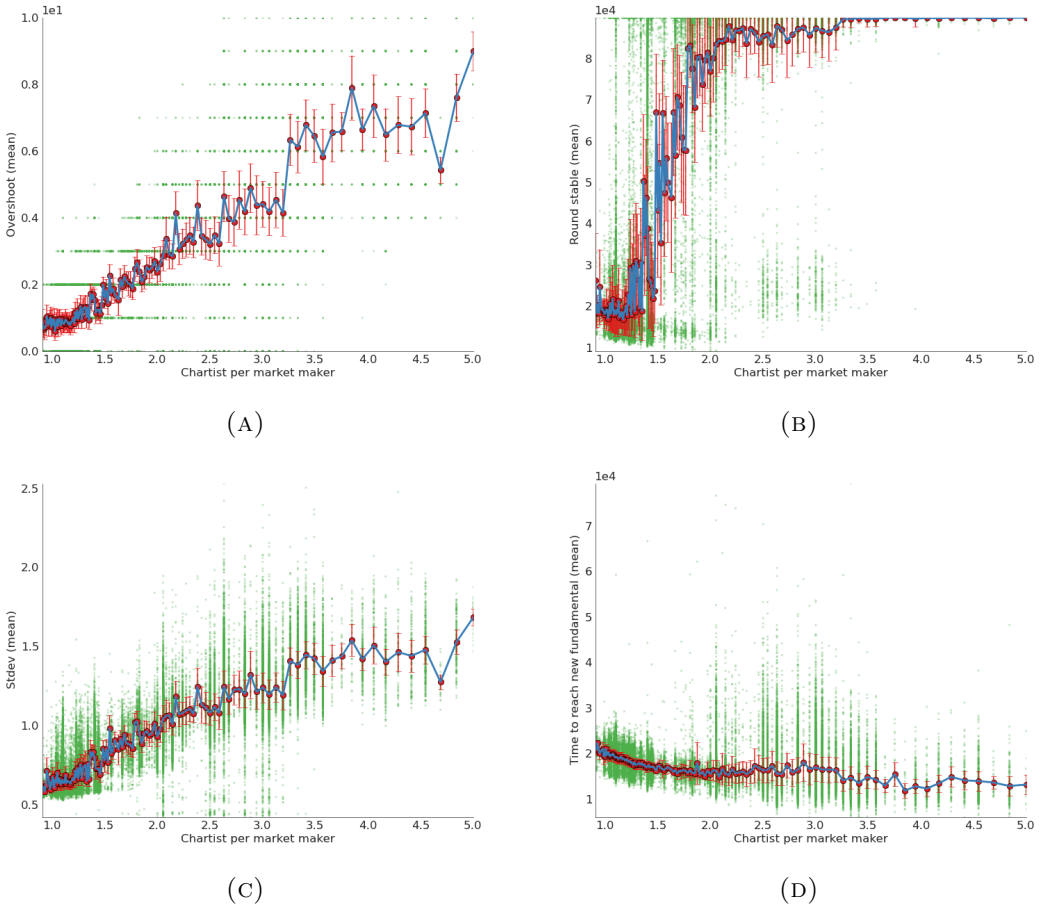


FIGURE 3.22: Correlations between $\rho_{\mathrm{A}}$and the fitness values when $N_m = 52$

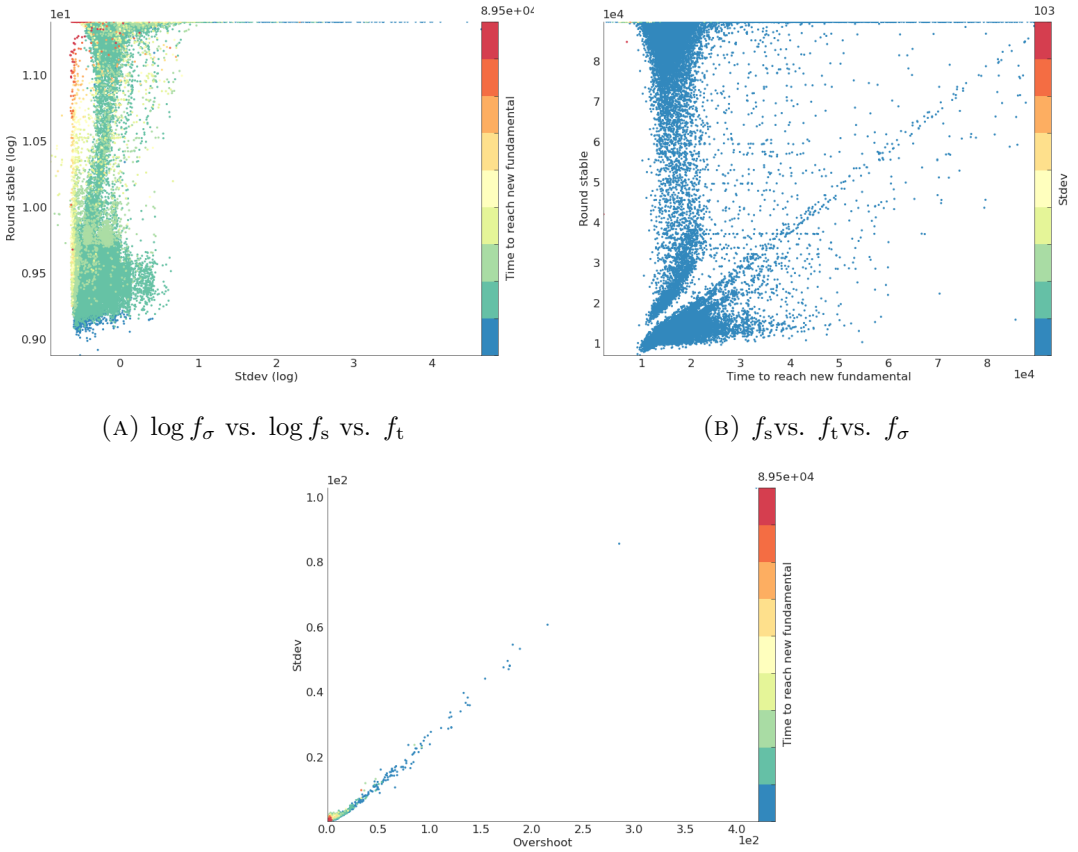
a small overshoot. These two groups of points were extracted by applying filters F4 and F5 (see table 3.6).

Simulations with overshoot

XXX NOT FINISHED

3.8.1.1 Fast simulations

All three experiments produces a group of simulations which had a quick response to the shock, but took longer to become stable. The simulations are in the column-shaped cluster in figure 3.24 and the all have relatively low f_{t}-fitness of less than 25000 rounds or so. These data points were extracted using filters F7 and F8 (see table 3.6).



(A) $\log f_\sigma$ vs. $\log f_\mathrm{s}$ vs. f_t

(B) $f_\mathrm{s}$vs. $f_\mathrm{t}$vs. f_σ

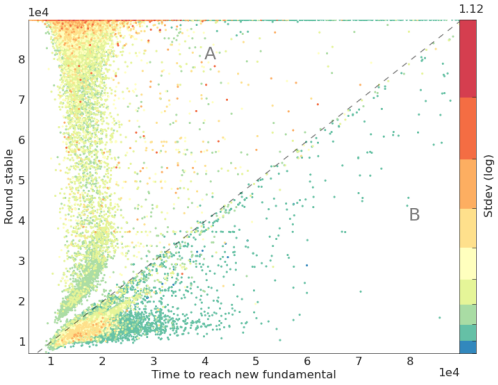
(C) $\log f_\mathrm{o}$ vs . $\log f_\sigma$ vs. f_t

FIGURE 3.23: Scatter plots of fitness measures in experiment d9.

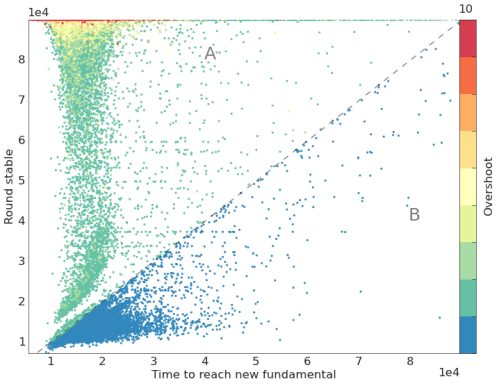
Unstable simulations with non-flickering prices

The final group of simulation that are singles out in this section are those that had very smooth price curves (that is, a small value for f_σ), yet did not manage to become stable. These simulations were selected by filter F9.

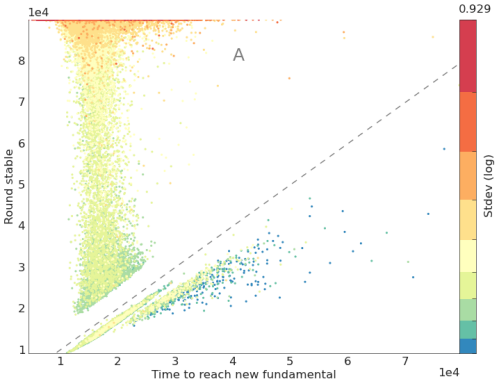
The criteria in table 3.6 do not prevent a simulation to be selected by different filters. The groups of points are therefore likely to have a non-empty intersection. Using the filters is an the attempt to separate the simulations by their behavior. Hence, the filters should in general not select the same data points. The Jaccard index $J(A, B)$ is calculated between sets A and B and used to determine the overlap between the sets. Figure 3.26 shows the Jaccard index between the sets created by applying the filters to each of the three data sets. Since the distance matrix is symmetrical, only the upper left part has been plotted. In case that the filter produced an empty set, the Jaccard index is undefined, resulting in white squares.



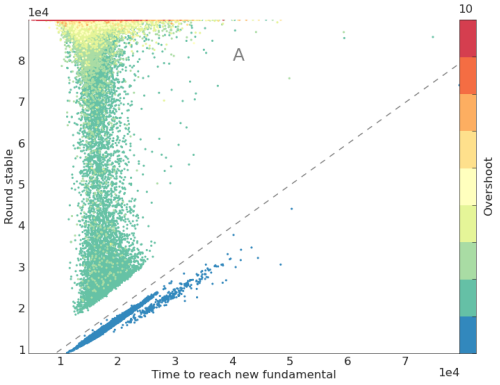
(A) Experiment d9



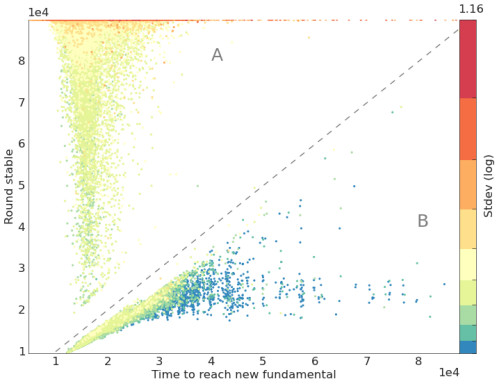
(B) Experiment d9



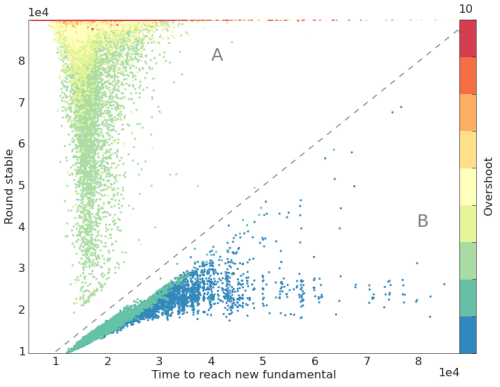
(C) Experiment d10



(D) Experiment d10

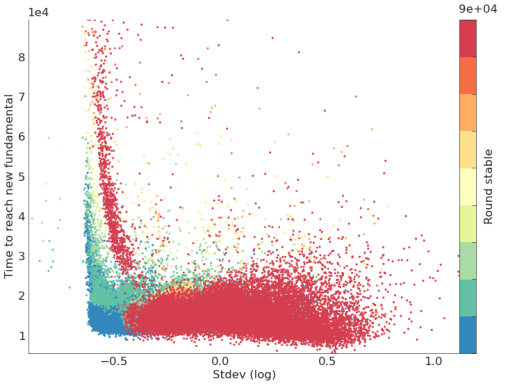


(E) Experiment d11

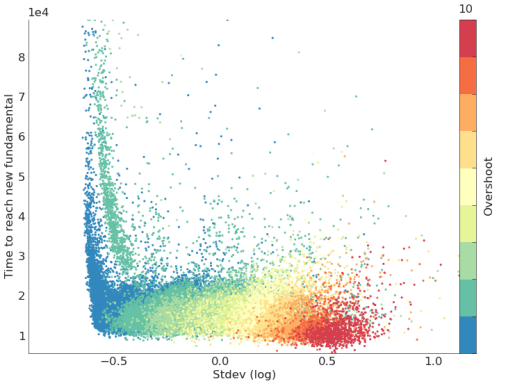


(F) Experiment d11

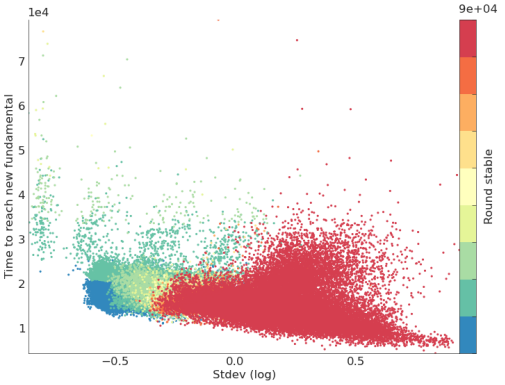
FIGURE 3.24: Scatter plot of f_{s}against $f_{\mathrm{t}}$with coloring showing $\log f_{\sigma}$ and f_{o}



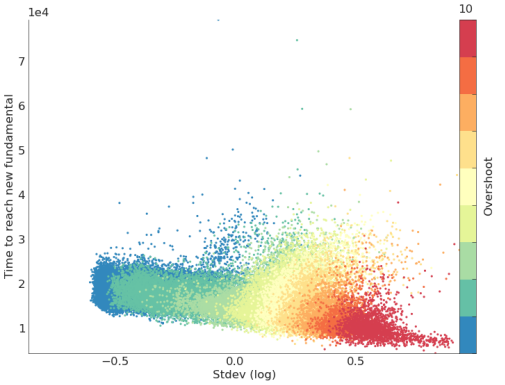
(A) Experiment d9



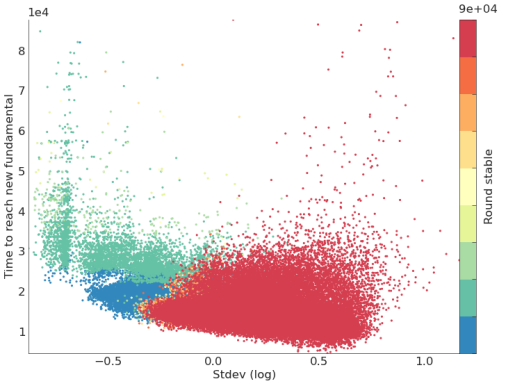
(B) Experiment d9



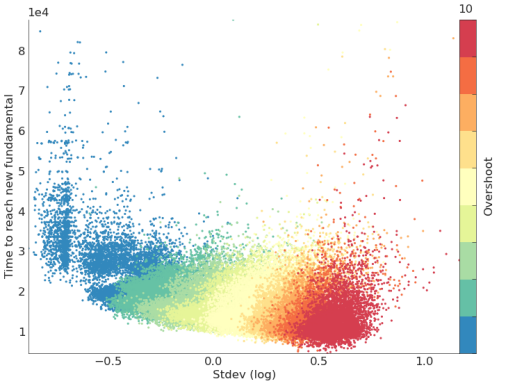
(C) Experiment d10



(D) Experiment d10

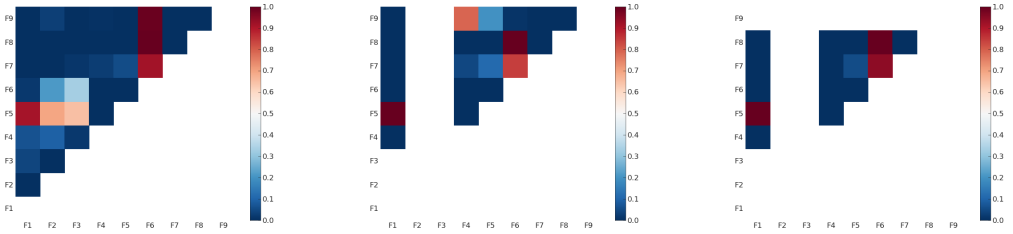


(E) Experiment d11



(F) Experiment d11

FIGURE 3.25: Scatter plot of $\log f_\sigma$ against $f_\mathrm{t}$with coloring showing f_s and f_o



(A) Experiment d9 (B) Experiment d9 (C) Experiment d10

FIGURE 3.26: Jaccard index between the sets of data points extracted by each filter

ID	Target simulations	Filter criteria
F1	Fast and stable (but maybe flickering)	$f_t < 12000$, $f_s < 12000$, $\log f_\sigma > 0$
F2	Slow, stable and not flickering (diagonal)	$\lvert f_t - f_s \rvert < 400$
F3	Fast and stable and not flickering (diagonal)	$\lvert f_t - f_s \rvert < 400$
F4	Stable before reaching fundamental, no overshoot	$f_s < f_t$, $f_o = 0$
F5	Stable before reaching fundamental, with overshoot	$f_s < f_t$, $f_o =>$
F6	Has overshoot	$f_s > f_t$, $f_o > 0$
F7	Fast response, quick to stabilize	$1000 < f_t < 25000$, $20000 < f_s < 40000$
F8	Fast response, slow to stabilize	$1000 < f_t < 25000$, $40000 < f_s < 75000$
F9	Smooth prices with a small overshoot, yet unstable	$e^{-0.5} - 0.1 < \log f_\sigma < e^{-0.5} + 0.1$

TABLE 3.6: Filter IDs and fitness-regions

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.2}$$

$$J(A,B) = \frac{|A \cap B|}{\min|A|,|B|} \tag{3.3}$$

Tables ??, ?? and ?? show the fitness and parameter arithmetic means for each of the nine groups selected by the filters

When the number of chartists were fixed as in experiment d10, the simulations with overshoot (picked out by filter F6) has an average overshot of $\mathrm{E}_{\mathrm{F}f_o}[6] = 4.1$ ticks. These markets had comparatively few, but fast market makers ($\mathrm{E}_{\mathrm{F}N_m}[6] = 67.1$ and $\mathrm{E}_{\mathrm{F}\lambda_{m,\mu}}[6]$), and fast chartists ($\mathrm{E}_{\mathrm{F}\lambda_{c,\mu}}[6] = 78.3$). Figure ?? shows that F7 and F8 mostly contain points that are also contained in F6. Both F7 and F8 a lower average overshoot, and

	F1	F2	F3	F4	F5	F6	F7	F8	F9
$\lambda_{c,\mu}$	70.1	65.5	49.5	81.5	76.7	58.5	75.8	75.6	76.0
$\lambda_{c,\sigma}$	5.4	5.5	7.8	4.5	4.9	6.8	5.0	4.9	5.0
$T_{c,\mu}$	59.7	58.8	51.8	70.1	65.8	53.5	65.7	64.8	65.9
$T_{c,\sigma}$	12.0	12.0	12.8	11.2	11.6	11.2	11.4	11.6	11.6
$H_{c,\mu}$	1744.0	1676.1	1758.1	1766.8	1773.6	1906.9	1795.3	1757.2	1777.9
$H_{c,\sigma}$	1483.2	1472.7	1573.4	1415.7	1444.2	1346.7	1437.3	1442.4	1465.4
$W_{c,\mu}$	29.5	27.4	29.3	30.5	30.0	28.6	30.0	29.7	29.9
$W_{c,\sigma}$	3.2	4.1	4.2	3.0	3.1	5.1	3.1	3.0	3.1
$\lambda_{m,\mu}$	45.2	38.5	37.6	49.8	47.3	39.2	46.1	46.3	44.5
$\lambda_{m,\sigma}$	4.2	4.9	4.6	4.3	4.4	5.4	4.1	4.2	5.1
$T_{m,\mu}$	39.8	38.2	40.1	37.9	38.6	35.5	39.0	39.1	35.5
$T_{m,\sigma}$	0.0	0.0	0.0	0.0	0.0	3.6	0.0	0.3	0.2
f_{o}	1.0	1.1	1.3	0.0	1.0	3.9	1.9	2.1	1.4
f_{s}	11712.3	44993.4	13454.5	13121.9	13468.1	80410.6	26840.4	61259.6	54143.9
f_{σ}	1.2	0.7	0.7	0.6	0.7	1.1	0.8	0.9	0.6
f_{t}	13767.0	45089.4	13579.4	15564.3	16743.6	16513.3	17502.8	16812.7	40564.2
Count	498	33	1031	33928	41982	30468	2016	1733	2160

Table 3.7: XXX

	F1	F2	F3	F4	F5	F6	F7	F8	F9
$\lambda_{c,\mu}$	N/A	N/A	N/A	123.8	121.5	78.3	116.9	104.2	95.0
$\lambda_{c,\sigma}$	N/A	N/A	N/A	6.5	6.5	9.0	7.1	8.7	8.7
$\lambda_{m,\mu}$	N/A	N/A	N/A	73.9	75.8	39.2	73.3	59.6	35.9
$\lambda_{m,\sigma}$	N/A	N/A	N/A	5.6	6.0	9.6	6.6	8.7	9.2
N_m	N/A	N/A	N/A	126.5	125.6	67.1	121.2	98.0	81.2
f_{o}	N/A	N/A	N/A	0.0	1.0	4.1	1.8	2.1	0.2
f_{s}	N/A	N/A	N/A	17025.9	15920.1	81539.2	27387.5	59444.5	24392.3
f_{σ}	N/A	N/A	N/A	0.6	0.7	1.2	0.7	0.9	0.6
f_{t}	N/A	N/A	N/A	20405.2	18724.9	15504.3	18725.1	16840.5	31017.6
Count	0	0	0	8486	25574	47493	5379	2528	399

Table 3.8: XXX

this is caused by the markets to have slower chartists, and fewer, slower market makers. When the number of chartists were varied as in experiment d11 (and with a constant of $N_m = 52$ market makers), the average latency of the chartists $\mathrm{E}_{\mathrm{F}\lambda_{c,\mu}}[6]$ did not differ from the other groups. Instead, the biggest difference was that markets with overshoot on average had a high number of chartists. As for the market maker latency, it was smaller than any of the other groups. On the the other hand, markets with no overshoot on average had a large number of marker makers when $N_c$is fixed to $N_c = 150$, although the market makers were not particularly fast. Markets with no overshoot also had the lowest average number of chartists among the nine groups.

	F1	F2	F3	F4	F5	F6	F7	F8	F9
$\lambda_{c,\mu}$	N/A	N/A	N/A	30.4	32.3	33.3	34.3	35.1	35.8
$\lambda_{c,\sigma}$	N/A	N/A	N/A	9.3	9.9	4.6	7.2	6.6	9.0
N_c	N/A	N/A	N/A	19.0	26.9	154.0	46.3	54.6	41.1
$\lambda_{m,\mu}$	N/A	N/A	N/A	67.2	55.0	38.9	50.0	47.3	46.0
$\lambda_{m,\sigma}$	N/A	N/A	N/A	8.4	11.6	22.9	17.5	19.5	14.7
f_{o}	N/A	N/A	N/A	0.0	1.0	14.6	2.0	2.6	0.0
f_{s}	N/A	N/A	N/A	15678.2	15075.3	87971.4	31954.9	63613.9	21851.4
f_σ	N/A	N/A	N/A	0.7	0.8	3.6	0.9	1.0	0.6
f_{t}	N/A	N/A	N/A	19965.3	18765.4	13685.0	17265.5	16449.5	31048.9
Count	0	0	0	71329	31377	84597	349	3014	1706

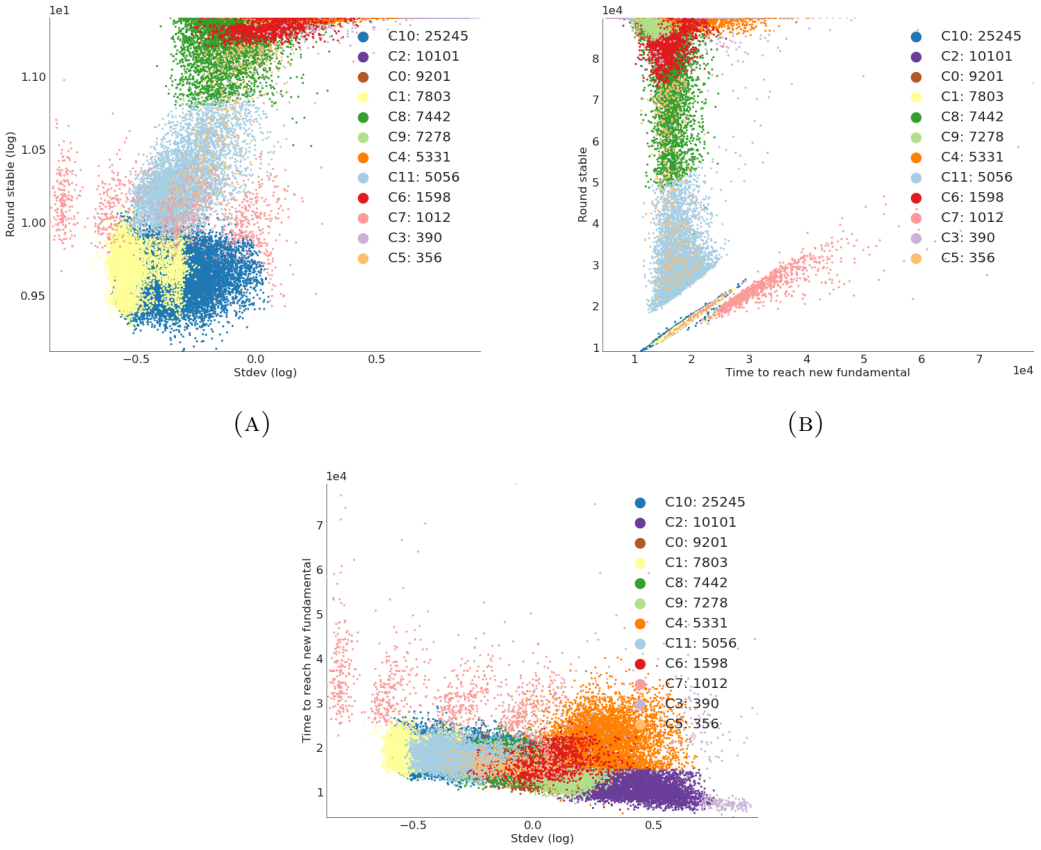
TABLE 3.9: XXX

3.8.2 Clustering with mixture of Gaussians

In this section, the focus is shifted from looking at population wide statistics to analysis sub-groups within each population. Whereas the previous sections showed that there do indeed exist statistical relationships between the latency of the agents and the behavior of the model, each discovered correlation was calculated over the entire population. Although the correlations reveal overall tendencies of the model behavior when changing a single parameter, little can be said about how the various parameters interact to determine model behavior. For instance, even though prediction of, say a negative correlation between $f_{\mathrm{t}}$was $\lambda_{c,\mu}$was found, there might be configurations of the model in which faster chartists were actually beneficial to the market.

One way to approach this problem is to see whether or not there exists relationships between partitions in the parameter space to partitions in the fitness space. As in section 3.2.5, the first step is to partition space, since each partition can be interpreted in terms of the model behavior. For instance, a partition covering the lower left half of the 2-dimensional space in figure 3.24 would encompass all the simulations which had a fast response time and became stable quickly (no matter if they had prices that flickered within the stability margin or not).

In order to investigate this, a Gaussian mixture model (GMM) was used to find clusters in the fitness space. All four fitness measures were used for the clustering. After discarding simulations with undefined fitness values and removing outliers, the data set contained 80813 data points. The large number of data points and the low number of dimensions made it possible to allow a each Gaussian component to have a full covariance matrix, giving the model a high level of flexibility. Figure 3.27 shows scatter plots of the data after it has been grouped. Tables **??** and **??** respectively show the mean and standard deviation calculated over each cluster. The tables are sorted by the average value of f_{o}.



(A) (B)

(C)

FIGURE 3.27: Scatter plots in fitness space showing the grouping of data points when using a GMM with 12 clusters.

XXX NOT FINISHED. WRITE ABOUT THE CLUSTERS THAT HAVE SOME INTERPRETABLE VALUE. IT DOES NOT HAVE TO BE ALL OF THEM. $\mathrm{Var}_{\mathrm{C8}}[f_{\mathrm{s}}]$ and $\mathrm{Var}_{\mathrm{C11}}[f_{\mathrm{s}}]$ and $\mathrm{Var}_{\mathrm{C5}}[f_{\mathrm{s}}]$ are large. The points in this cluster have parameters which

Chartist latency and market response time

As was noted earlier, the evolution of $\lambda_{c,\mu}$and f_{t}indicates that slow chartists made the market slow, and fast chartists made the market fast. C1 is the cluster with the $\mathrm{E}_{\mathrm{C1}}[f_{\mathrm{t}}]$

XXX CONSIDER MOVING SOME OF THE TABLES TO THE APPENDIX?

3.9 Summary of results

XXX NOT FINISHED

	f_o	f_s	f_σ	f_t	$\lambda_{c,\mu}$	$\lambda_{c,\sigma}$	$\lambda_{m,\mu}$	$\lambda_{m,\sigma}$	N_m	Count
C1	0.0	16301.2	0.6	19217.4	127.2	6.2	78.4	5.2	132.2	7803
C7	0.3	24383.1	0.7	32005.5	87.0	9.5	25.8	10.5	66.7	1012
C10	1.0	15832.8	0.7	18602.8	121.9	6.5	76.3	6.0	126.3	25245
C8	2.0	81599.7	0.9	16333.6	101.8	8.7	56.4	9.2	91.3	7442
C11	2.0	30515.0	0.7	17822.9	114.2	7.3	71.7	7.0	117.5	5056
C0	3.0	89015.0	1.1	14687.5	89.0	9.2	39.5	10.0	66.8	9201
C5	3.0	51306.4	0.9	16508.7	102.2	9.0	59.2	8.8	96.0	356
C6	3.0	83914.2	1.1	16820.9	92.3	9.2	40.6	10.0	72.4	1598
C9	4.0	89496.0	1.2	14459.8	77.7	9.6	29.9	10.4	56.7	7278
C4	5.0	89896.6	1.4	22034.2	62.4	9.4	21.5	10.3	51.6	5331
C3	6.7	86440.9	1.8	18692.7	54.9	9.2	26.2	10.5	38.0	390
C2	7.9	89996.1	1.6	11574.7	36.4	9.2	26.4	9.6	36.4	10101
Outliers	11.5	89998.8	2.2	8835.1	17.8	9.3	24.3	8.5	19.4	740

TABLE 3.10: Cluster means (d10)

	f_o	f_s	f_σ	f_t	$\lambda_{c,\mu}$	$\lambda_{c,\sigma}$	$\lambda_{m,\mu}$	$\lambda_{m,\sigma}$	N_m	Count
C1	0.0	2142.8	0.0	2220.9	10.9	4.7	7.1	3.3	11.3	7803
C7	0.5	4827.7	0.2	7424.7	18.4	3.8	18.7	5.4	23.2	1012
C10	0.0	2150.8	0.1	2261.1	12.7	4.7	9.3	4.1	14.7	25245
C8	0.0	10563.6	0.1	1926.7	10.3	4.0	12.2	5.3	20.0	7442
C11	0.0	6952.5	0.1	2276.1	13.8	4.7	11.6	4.7	18.3	5056
C0	0.0	1380.7	0.1	1479.2	9.7	3.4	14.7	5.0	12.7	9201
C5	0.0	17151.2	0.1	2087.1	11.2	4.2	12.5	5.2	21.4	356
C6	0.0	3961.3	0.1	2646.9	10.3	3.5	16.9	5.2	16.0	1598
C9	0.0	1048.5	0.1	2368.7	13.0	3.4	12.7	4.9	9.4	7278
C4	1.4	423.6	0.2	4246.5	21.3	4.3	11.6	5.3	10.4	5331
C2	1.6	211.8	0.2	1943.0	16.0	5.2	11.7	5.5	13.7	10101
C3	2.9	7387.0	0.4	11884.0	26.0	4.5	15.0	5.5	30.6	390
Outliers	0.8	0.6	0.3	2691.9	11.3	6.1	13.4	5.7	14.2	740

TABLE 3.11: Cluster standard deviations (d10)

	f_o	f_s	f_σ	f_t	$\lambda_{c,\mu}$	$\lambda_{c,\sigma}$	N_c	$\lambda_{m,\mu}$	$\lambda_{m,\sigma}$	Count
C5	0.0	15270.3	0.7	19209.4	30.0	9.3	17.5	68.7	7.9	66665
C11	0.0	21182	0.6	29334.9	35.5	8.9	40.2	46.6	14.5	4369
C0	1.0	14897	0.8	18523.0	32.2	10.0	25.9	55.6	11.3	30258
C4	1.0	19666	0.9	25087.1	35.3	7.3	55.5	39.0	18.9	1108
C9	1.0	30399	0.7	35789.7	34.4	7.8	45.6	43.6	17.9	591
C6	2.0	79198	0.9	15620.1	37.5	6.1	63.5	47.3	20.6	11141
C7	2.6	78771	1.0	20524.6	37.7	5.6	63.8	38.5	22.6	967
C1	3.0	88210	1.0	14225.8	37.9	4.6	88.0	44.2	23.3	10577
C8	4.0	89633	1.1	13452.7	33.6	4.4	103.9	44.4	22.7	8099
C3	5.7	89842	1.4	20087.3	34.6	4.5	136.1	29.6	23.9	9613
C10	7.0	89935	1.8	27587.6	32.8	4.2	165.8	25.9	24.3	1401
C2	7.0	89982	1.5	11691.4	32.3	4.4	154.4	37.6	22.7	19067
Outliers	40.5	89951	9.9	10434.7	29.2	4.0	255.5	36.4	23.5	23454

TABLE 3.12: Cluster means (d11)

	f_o	f_s	f_σ	f_t	$\lambda_{c,\mu}$	$\lambda_{c,\sigma}$	N_c	$\lambda_{m,\mu}$	$\lambda_{m,\sigma}$	Count
C0	0.0	1349.9	0.1	1678.1	13.7	6.5	12.1	16.3	8.1	30258
C1	0.0	2448	0.1	1383.3	16.8	4.6	15.5	18.6	10.1	10577
C4	0.0	3347	0.1	4304.5	16.8	5.7	33.3	17.4	10.8	1108
C5	0.0	1102	0.0	1155.7	12.0	6.8	4.2	13.0	6.9	66665
C6	0.0	11795	0.1	1640.9	17.7	5.1	18.1	19.4	10.1	11141
C8	0.0	893	0.1	1446.5	16.7	4.4	19.1	20.5	9.7	8099
C11	0.0	2801	0.1	4505.8	15.0	6.4	31.7	18.6	11.2	4369
C7	0.6	6266	0.2	4695.2	17.4	4.9	29.9	16.7	9.7	967
C9	1.1	9640	0.2	18435.6	16.5	5.8	33.1	18.6	10.2	591
C2	1.7	143	0.2	1566.7	16.3	4.7	39.8	17.6	8.9	19067
C3	2.2	522	0.3	3765.1	16.7	4.5	48.7	13.0	8.9	9613
C10	2.3	407	0.3	10643.3	16.8	4.4	52.4	12.4	8.8	1401
Outliers	53.1	1107	15.0	3790.6	15.5	4.5	50.7	15.9	8.3	23454

TABLE 3.13: Cluster standard deviations (d11)

Discussion

3.10 Benefits of fast traders

3.10.1 Fast market makers reduce price flickering

In

3.11 Agent strategies market crashes

It is somewhat intuitive that HFT chartists should be suspected of having an influence on the market such that the market become more likely to crash. The results did indeed confirm this, as it was shown that a negative correlation exists between the number of chartists active in the market, and the size of the overshoot (see figure **??**).

It is conceivable that the market makers also contribute to the market crashing, since the market makers also ignore the true fundamental price. Indeed, the results showed that the market does not crash from having fast chartists alone. The results also showed that the market will not crash from having only fast market makers either. Instead, both types of fast traders were required to make the market crash. The following text will provide an explanation as to why this is so.

It was found that markets containing no market makers will almost never crash. Without the presence of market makers, even a market saturated with chartists will eventually return to the fundamental price, as the force of the initial downtrend created by the fundamentalists dissipates.

Case with no fast traders

The shock to the fundamental creates a drive in the market for falling prices, due to the presence of the fundamentalists. The fundamentalists have a large delay, and the

downwards drive is therefore initially small, as most of the fundamentalists fail to observe that the shock has happened. As the fundamentalists begin to observe the shock, they start submitting sell orders at lower prices, as they believe that the stock is no longer worth the price at which they were previously willing to sell. Hence, the number of sell orders starts to increase.

As for the buy side of the order book, the number of new buy orders starts to fall, as the fundamentalists start to register the shock. The buy orders that were previously submitted by slow traders at prices slightly below the old fundamental are not canceled, as the model assumes that the fundamentalists are too slow to register the change in the fundamental. Furthermore, in order to simulate an order book with a long trade history, the order book was initialized with a large number of market orders with a normal price distribution centered around the initial fundamental. These buy orders provide matches for the increasing number of sell orders, and the traded price begins to drop. If the market has no fast traders, the traded price will eventually reach the new fundamental, and stay within the stability margin. Thus, in the rather simple case where the market only contains fundamentalists, crashes do not occur.

Case with chartists but no market makers

When adding chartists, the market starts to behave in a different manner. The chartists do not use any information about the true true fundamental price, but are instead only concerned with the actual traded price. After the shock, the fundamentalists start submitting bids to sell at lower prices, Depending on the parameters of the chartists, some chartists will interpret this as a downtrend, while others will not. The chartists that detect a downtrend will start submitting bids to sell at a lower price, as they believe the price will continue to drop. The chartists that did not detect a trend will remain inactive. The sell orders submitted by the chartists are matched by previously existing buy market orders at lower prices. Hence, the chartists add to the force that drives the traded price down by submitting sell orders at lower prices. However, since the only active traders in the market are fundamentalists and chartists, the supply of buy orders at prices lower than the new fundamental are limited. The chartists that detected a downtrend will exclusively place sell orders, and the fundamentalists will rarely submit buy orders at prices much lower than the fundamental. When the supply of buy orders at prices below the new fundamental dries out, the execution price will not drop further. The chartists that detected a downtrend will continue to submit sell orders for as long as they believe that the trend continues, but the only new buy orders are submitted by the fundamentalists. As some of the fundamentalist buy orders are placed a few ticks

below the fundamental price, the execution price will flicker, but always in a region close the true fundamental price.

Case with chartists and market makers

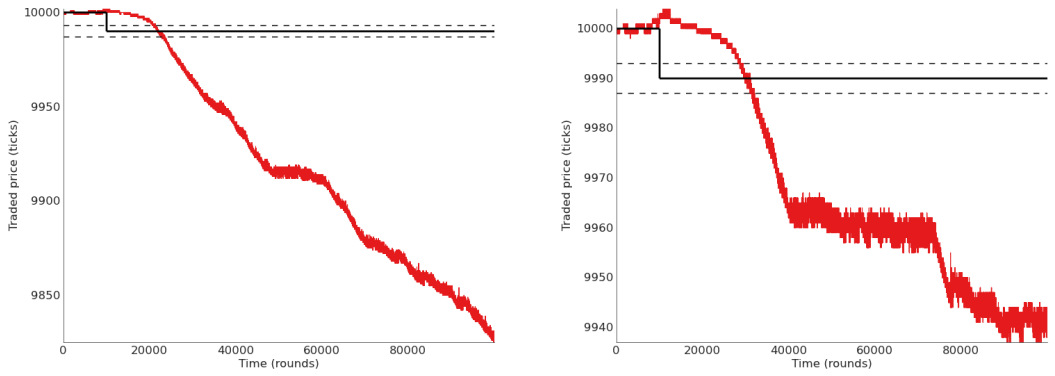
When the market also contains market makers, the situation is quite different. Like the chartists, the market makers ignore the fundamental price. Instead they submit buy and sell orders just above and below the best buy and sell prices existing in the order book at the time that the market maker requested the market information. The market maker strategy is such that it will always try to follow a narrowing spread, in order to stay competitive. On the other hand, if the market maker discovers that the spread is widening, the agent will attempt to avoid buying/selling at a higher/lower price than necessary. The agent therefore tries to follow the widening spread by submitting buy/sell orders at lower/higher prices.

When the sell price starts to drop after the shock due to the activity of the fundamentalists and the chartists, the market maker will try to stay competitive on the sell side be decreasing its own sell price. If the decrease in the sell price is large enough to make the spread smaller than what the agent is prepared to risk, the market maker submits a new sell order with as low a price than its strategy allows.

On the other side of the order book, the best buy price starts to drop as the sell orders submitted by the chartists start to eat away at the existing buy orders. If the market maker orders are among the orders that match the chartist orders, the market makers request the latest market information and use it to submit new buy orders. If the market maker orders were not matched by chartist orders the market makers will cancel their existing buy order and submit a new one at a lower price in order to stay a competitive buyer. In any case, the market maker will eventually start submitting buy orders at a lower price than before. Hence, the market makers provide the market with a new supply of buy orders, the prices of which can be arbitrarily low. As these buy orders are filled by sell orders, initially by both fundamentalists and chartists but eventually solely by chartists, the traded price will drop, and the chartists will continue detecting a trend and continue to drive the market down into a crash.

3.11.1 Frequency of crashes

Crashes did not occur often particularly. Even during experiment d11, in which the genetic algorithm ended up prioritizing the market response times, and generate a large number of genes with many chartists and very fast market makers, the market had an



(A) (B)

FIGURE 3.28: Two examples of market crashes

overshoot of over 25 tick in just less than 0.2% of the cases[5]. In experiment d10, not a single case of markets with an overshoot of over 17 ticks was generated.

XXX ADD A SMALL DISCUSSION OF HOW OFTEN CRASHES OCCUR IN REAL MARKETS

3.11.2 Agent speed and market crashes

XXX INSERT DATA THAT SHOWS THAT CRASHING MARKETS HAD FASTER AGENTS THAN NON CRASHING MARKETS

3.12 Market makers causing the stock to be over-evaluated

XXX NOT FINISHED. COLLECT EVIDENCE

3.13 title

XXX discuss the variability of market behavior for markets simulated with the same parameters. is it reasonable that the same set of parameters can cause various types of behavior? XXX

[5]The simulation was run around $4 \cdot 10^5$ times, and 7989 of these had $f_o > 25$

3.14 Co-location

Stable markets had an average market maker latency of $\lambda_{m,\mu} = 60ms$, while crashing markets had an average of $\lambda_{m,\mu} = 30ms$. Is it realistic that just a factor of two can have such a dramatic effect on the markets?

The recent construction of the transatlantic fiber line reduced the latency from European markets to American markets

Note also that latency is nt just a result of physical distance in the market. Network crowding can cause an increase in the latency as well.

3.15 Strategy crowding

Strategy crowding is a commonly observed phenomenon in the field of multi-agent systems. XXX FIND REFERENCE XXX

3.16 Future work

A commonly used strategy among high frequency traders is that of arbitrage,

Additional tables

.1 Dataset 1

Write your Appendix content here.

Third party software

Deap scoop sklearn matplotlib numpy, scipy and pandas geometric brownian walk

Additional figures

Bibliography

[1] C. J. Hawthorn, K. P. Weber, and R. E. Scholten. Littrow configuration tunable external cavity diode laser with fixed direction output beam. *Review of Scientific Instruments*, 72(12):4477–4479, December 2001. URL http://link.aip.org/link/?RSI/72/4477/1.

Bibliography

[1] Chi Wang, Kiyoshi Izumi, Takanobu Mizuta and Shinobu Yoshimura: "Investigating the Impact of Trading Frequencies of Market Makers: a Multi-agent Simulation Approach" *SICE Journal of Control Measurement, and System Integration, Vol. 4, No. 1, pp. 001-005, January 2011.*

[2] Michael J. McGowan: "The Rise of Computerized High Frequency Trading: Use and Controversies", *2010 Duke L. & Tech. Rev., 2010.*

[3] Thomas McInish and James Upson: "Strategic Liquidity Supply in a Market with Fast and Slow Traders", *Available at SSRN 1924991 (2012).*

[4] Niel Johnson, Guannan Zhao, Eric Hunsader, Jing Meng, Amith Ravindar, Spencer Carran and Brian Tivnan: "Financial Black Swans Driven by Ultrafast Machine Ecology", *Available at SSRN (2012).*

[5] Kiyoshii Izumi, Kiyoshi, Fujio Toriumi, and Hiroki Matsui: "Evaluation of automated-trading strategies using an artificial market", *Neurocomputing 72.16 (2009): 3469-3476.*

[6] Chiarella, Carl, Giulia Iori, and Josep Perelló: "The impact of heterogeneous trading rules on the limit order book and order flows" *Journal of Economic Dynamics and Control 33.3 (2009): 525-537.*

[7] Peter Gomber, Björn Arndt, Marco Lutat, Tim Uhle: "High-frequency trading." *Available at SSRN 1858626 (2011).*

[8] J. Doyne Farmer and Spyros Skouras: "An ecological perspective on the future of computer trading" *Quantitative Finance 13.3 (2013): 325-346.*