Analyzing Different Trading Strategies in an Artificially Created Bitcoin Market

Jasper Krebbers, Lennart van de Guchte, Bram Wiggers

University of Groningen

Abstract

One of the most complex men designed constructs is economy with its financial markets. The cryptocurrency Bitcoin is a special case as it contains an Order Book mechanism and is very volatile, which makes profitable trading a complex task. In this research we create an agent-based model that simulates this Bitcoin market, as reported in a paper by Cocco et al. [4], to look at the differences between traders who have no experience (Random Trader), traders with standard idea's about economy (Chartists), a machine learning trader (Learning Trader) and a high-frequency trader (Fast Trader). The model consist of a heterogeneous Multi-Agent System to find which strategies are best to use in a Bitcoin market. In the model, different agents are simulated, together with the Bitcoin properties such as the Order Book mechanism, the mining of bitcoins and the coming and leaving of traders. The results show that the Learning Trader learns to always buy, due to the fat tail distribution of the market, which leads to a loss in wealth. Both Random and Chartist Traders make a little profit on average but the Fast Trader outperforms all strategies by far and makes a lot of profit as time progresses. However, as some factors are led out or simplified in our model such as paying fees to exchange bureaus and the timescale for updating the bitcoin price we can not say much about the results of applying these strategies on real markets.

1 Introduction

1.1 Problem

Over the last few years, cryptocurrency has entered the market[3]. Bitcoin (in this paper we will use capital-B for the Bitcoin system and lower-case-b for the unit of amount) is one example of such a virtual currency. The network was created by Satoshi Nakamoto [7], and has gotten a lot of attention lately. Important reasons for this are (1) privacy, (2) absence of financial government and (3) the amounts of money that people have earned and lost.

The Bitcoin economy is, as every form of money, based on scarcity. In regular financial situations, this scarcity is controlled by institutions, and when there are discrepancies, actions are taken to bring back balance. As Bitcoin lacks this centralized authority, there is less certainty about happens to this currency. The book keeping of this network of buyers and sellers is not centralized anymore, but distributed over all the users. Users who help with this book keeping, are rewarded with bitcoin. The technological properties of this "block chain" mechanism will not be elaborated on in this paper, but can be found in the 2015 paper by Bohme [3].

The creation of Bitcoins works by mining. The computational power of computers is used to find a certain block in the blockchain, which is paid for with Bitcoin. There is a finite amount of Bitcoin, and this amount is almost reached[3]. probability of generating Bitcoins gets lower and lower with each mined bitcoin. However, since the value of Bitcoin increases, the more interest it generates.

Several solutions to machine based stock prediction have been proposed. Allen (1993) [1] proposed a genetic algorithm but had no big advantages. There were good results, but after adding the transaction costs, the system was not able to make a profit. However, it was able to predict periods of volatility and positive and negative returns. It was found that when volatility is lower, the expected returns decrease and the upward revision is picked up by the rules.

An important question is: which strategy is best to trade with in the Bitcoin market. To find an answer to this question we will simulate a Multi Agent System cryptocurrency market in which a set of different agents can freely trade Bitcoins with each other. With a heterogeneous pool of agents, we will introduce four strategies for trading, to find out which strategy can make the most profit and approximates human behavior best. With the model we will simulate approximately two years of Bitcoin trading. By comparing the real data to the data of our simulation, we will analyze the external validity of our simulation. When the simulation approximates the real data we will analyze the different trading strategies.

1.2 Strategies

In this section different strategies are introduced to use in the Bitcoin market. The Random Trader and Chartist trader are based on the paper of Cocco[4], the learning and Fast Traders are new ideas introduced in this paper.

1.2.1 Random Trader

The Random Trader simulates new traders who enter the market unknowingly. They buy and sell randomly, without knowledge about the market. As the buying and selling is not influenced by the market, the Random Traders tend to have a stabilizing effect on the market[4].

1.2.2 Chartist Trader

The Chartist trader has knowledge of the market, and looks x steps back to see whether the price is going up or down. If the price is going up, the Chartist starts buying, if it is going down the Chartist starts to sell. This way, the Chartists have a exponential effect on the market, because they drive the price lower when it is already going down and vice versa.

1.2.3 Learning Trader

The market of stock trading is being done more and more by artificial intelligence. One popular technique nowadays is combining reinforcement learning with artificial neural networks. Here reinforcement learning is used to compute an optimal action in an environment. Within stock trading this action will be buy, sell or pass and the environment is the market. This market is represented by a neural network because it can handle a large amount of possible states, who then are being represented by a set of features. The Learning Trader simulates one of the professional traders who use computers to estimate the price of Bitcoins.

1.2.4 Fast Trader

The Fast Trader puts focus in being the fastest to respond to changes in the market and speculate on falling or exploding prices. It represents people and algorithms who are full time trading and constantly aware of changes. The Fast Trader buys when he detected a local minimum and sells when he has has detected a local maximum. The Fast Trader bets that a large fluctuation in price is on the way, therefore he tries to sell before the perceived fluctuation. The traders can trade every time step to simulate flashtrading algorithms.

1.3 State of Art

In 2016, L. Cocco et al. published their findings on Bitcoin mining[5]. In their paper, they introduced an agent-based artifical market model of the Bitcoin mining process. In the model the advancement on technological level were modelled, together with the price of Bitcoin. The core aspects of mining were found, as well as the unit root property, the fat tail phenomenon and the volatility clustering of Bitcoin price series. A year later, another article[4] was published, this time focusing on the trading of Bitcoins. In this article, two types of agents were able to trade Bitcoins with each other, via an Order Book principle which is similar to the one of the Bitcoin market. The trading was done by using fiat cash (*dollars*) and Bitcoin. A simplification of the mining mechanism and the the migration of traders was added to this model. Contrary to the 2016 paper, this paper found statistical properties of the price observed in the real Bitcoin market, such as the unit root property, fat tail phenomenon and volatility clustering. We will elaborate more on this model in Section 2.

1.4 New Idea

After the previous research that has done by Cocco, we will analyze more traders' behaviors, which was suggested by Cocco. We will create four different strategies for traders, and analyze the behavior of the simulation with different kinds of traders. Since Cocco succesfully researched the price of Bitcoin, and made a simulation which could estimate the real price of Bitcoin, we focus on the behavior of the agents more, especially the Learning Trader and Fast Trader which are added. The comparison between the Random, Chartist, Fast and Learning Traders will be done by looking at their generated profit and relative wealth over time. We will than check the external validity by comparing the price of the Bitcoin to the real data and the data of Cocco.

The remainder of this paper is set up as follows. In Section 2 the model will be described. After this we will discuss the results in Section 3. Lastly, we will discuss what conclusions we can derive from these results in Section 4

2 Methods

2.1 Simulation model

The model by Cocco was made in the Smalltalk language. It contained two types of agents, the Order Book and a simple approximation of the bitcoin mining. In this project we will recreate this model in Python, together with the Mesa, TA-Lib and Keras package. The Mesa package is used for the creation of Multi Agent Systems. The TA-Lib package is used to handle the real-world data we use for number of traders. The Keras package is needed for the Learning Trader for deep learning.

2.1.1 Agents

In our multi-agent system (MAS) the agents represent traders at the Bitcoin market. Those traders hold an amount of fiat currency (dollars) and an amount of Bitcoins. Therefore they can buy Bitcoins using the dollars and earn dollars by selling the Bitcoins. New traders can enter the market in which they start with only an amount of fiat currency. This wealth distribution over all agents is determined by the pareto law, to estimate real world data, since financial power of traders hugely varies. To determine when traders will buy or sell their Bitcoins we will implement trading strategies. For this we will use two strategies from the paper of Cocco[4] and one we came up by our self. The buy and sell prices are also determined by functions derived from Cocco[4] and go as follows:

$$lp(t) = G(a, sd) * p(t) \tag{1}$$

Where lp is the limit price and G is a random draw from a Gaussian distribution with average a and standard deviation sd. For buying a = 1.02 and for selling a = 1.00 while for both the sd = 0.05. A description of all used strategies is given below:

- Random Traders represent novice traders which enter the market for reasons as personal interest, entertainment or curiosity, not for professional reasons. They issue buy and sell orders because they want to have bitcoins or need cash. In the model, we represent this by the random trading of these traders. Buy and sell orders are issued in a random fashion. The Random Traders give stability to the model by essentially adding noise.
- Chartists represent the speculative traders, who focus on making a profit on the Bitcoin market. Their rule for speculating is relatively easily, when the price is rising it will keep rising and when it is falling it will keep falling. Therefore, the Chartists issue buy orders when the price is climbing and sell orders when the price is falling. The Chartists produce high volatility in the simulation, as the effect is reinforcing itself.

Algorithm 1 Trading behavior

let B be an ordered list of buy orders where B(b) is the first order and has the highest buyprice
let S be an ordered list of sell orders where S(s) is the first order and has the lowest sellprice
while S(s) smaller than B(b) **do**:
    resolve trade with orders S(s) and B(b)
    if S(s) quantity equals 0 **then**:
        remove S(s) from S
        new S(s) = S.next
    end if
    if B(b) quantity equals 0 **then**:
        remove B(b) from B
        new B(b) = B.next
    end if
end while

- Learning Traders combine the reinforcement learning technique Q-learning [8] and the type of artificial neural network: Long Short-Term Mermory (LSTM) [6]. For a deeper insight into how these techniques work see the papers in the reference section, here only an explanation is given about how these techniques are being used in our model.

  In the learning phase the agent is being fed with 100 runs of simulated data (only Random and Chartist Traders were used here). First the bitcoin data at a time step is represented as a state with six features. These features consist of the current bitcoin price, difference between current and yesterdays bitcoin price, simple moving average (SMA) over both 15 and 30 days, relative strength index (RSI) and the bitcoin price minus the SMA over 15 days. Using this state representations it uses Q-learning to find the optimal action (buy, sell or pass) to take in a specific state. The reward of an action is defined by the height of profit it can make when buying at a current state and selling at another state and vice versa for loss. These state representations combined with their optimal action are then learned by the neural network. Now when including the Learning Trader in the simulation the agent uses the neural network to predict its optimal action in a specific state.

- Fast Traders represent betting traders and algorithms that try be ahead of the curve. Buying and selling when they think a maximum or minimum is reached, this is risky since there is no guarantee that the current peak is indeed a global maximum or minimum. To find a peak (local maximum or minimum) we look 2 days back in time. If the current bitcoin price and that of 2 days ago is higher than the bitcoin price of yesterday a local minimum is found so the trader will start to buy. For selling a local maximum must be found in which exactly the opposite occurs.

2.1.2 The Order Book

To order all the different buyings and sellings, the Order Book is used. The book is used int he Bitcoin market to make sure that there are no sales done twice, and as many orders as possible are processed. Every agent can create buy- or sell orders. These orders are saved in the order book to see which buy order can be connected to which sell order. If every buy order is connected to the best sell order, and vice-versa, all agents will have the best prices for their Bitcoins. To create this, we created two sorted lists with orders. The lists will be sorted on their selling/buying prices. The orders will be executed as long as the top two are willing to sell to each other. If this is not longer the case, the day will end and new orders can be accepted. The system is explained in the following algorithm:

2.1.3 Mining

To simulate the mining, the real Bitcoin data is used. At each time-step, the amount of Bitcoins is checked and compared to the amount of bitcoin there were at that time. If there were more bitcoins in real-life than in the simulation, Random Traders receive random bitcoins. The fact that mining in the real world costs money (power) is neglected. Each timestep, the number of missing bitcoins are divided by a random number of traders. The Bitcoin mining is estimated by the following formula:

$$M(t) = ((4.709 * 10^{-5})) * (t^3) - (0.08932 * (t^2)) + 98.88 * t + 78880)/100 - B(t) \quad (2)$$

Where M(t) is the amount of bitcoins that are mined at timestep t and B(t) is the amount of bitcoins in the model at timestep t

2.1.4 Market migration

At each time-step, traders the simulation checks whether the number of traders in the simulation is the equivalent to the real number of traders[REF] at that moment. If there are too little agents in the simulation, there are agents added, if there are too much, agents retire. When the traders retire, all bitcoins are sold for the current bitcoin price. If a trader has lost all his bitcoins, it can step out of the market. On the other hand, new traders can come into the simulation. These traders enter the market with amount money Ei per agent i. The amount of money the agents start with is given by a Pareto function, simulating the real data in creating few very rich starting agents, and much poor agents. When these agents come to the market, they will try to buy Bitcoin.

2.2 Experiment design

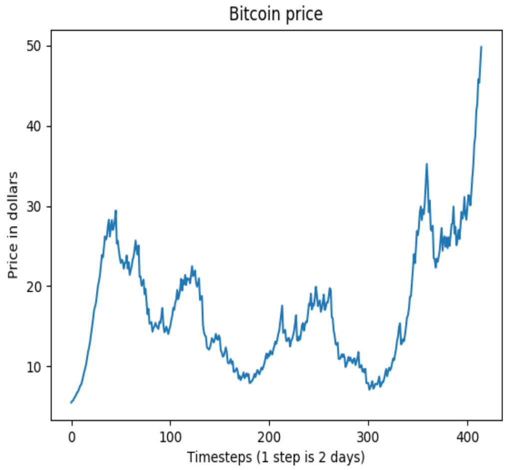
In our design we run the simulation and check the global price over time. At first we will do this one time, to check if the price is going the right way, and if we're correctly simulating the Bitcoin price. We will compare the graphs to the results of Cocco and the real data. We will then take an average of 100 runs, to see if we can see general rules in the system. We will make a graph with the average profit earned by the different kinds of traders. To compute the profit we subtract the initial wealth (wealth of the traders when entering the market) from its virtual wealth, so the Pareto function does not have an influence on the performance of the agents. This virtual wealth consist of the amount bitcoins times the current bitcoin price plus the current wealth of the traders. Here we will take an average over 100 runs as well.

3 Results and Analysis

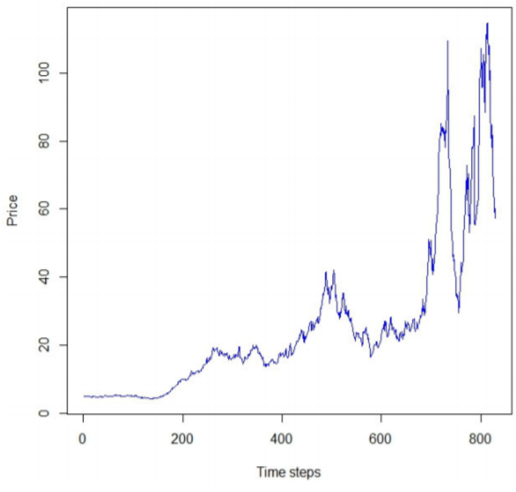
In this section we discuss the result that the simulation has generated. We discuss the results of the Bitcoin price in Section 3.1 and the differences between the strategies in Section 3.2.

3.1 Simulated bitcoin market

In Figure 3 the results of the simulation of our own artificial Bitcoin market (1a) and that of Cocco et al. (1b) are shown. These plots represent the average Bitcoin price over a time period of 830 days, from 2012-01-01 till 2014-04-10, using only Random and Chartist Traders. In Figure 2 also the real value of the Bitcoin price is plotted over this specific time period. As you can see we managed to create a market that shows some of the characteristics of the real Bitcoin market. As for example its volatile behaviour and the possibility to increase or decrease its value in rapid time. However, compared to the results of Cocco et al. the model underfits the real Bitcoin data much more. The reason for this is the model's sensitivity to market migration. Cocco et al. showed in their paper that the real Bitcoin price is highly related to the amount of unique addresses (number of traders) on the market (see Figure 2) which ensures the fat tail distribution. Although the same kind of market migration mechanism is used in our model we did not manage to show this fat tail distribution of the Bitcoin price as Cocco et al. did. A possible reason for this is that the threshold value for Chartist traders for when to buy or sell is different in our model because they did not mention this threshold value in their paper. However, we did try different thresholds values but still agents would buy often when they entered the market.



(a) Own model.



(b) Model of Cocco et al. [4].

Figure 1: The average bitcoin price over 100 runs in time period of 830 days.

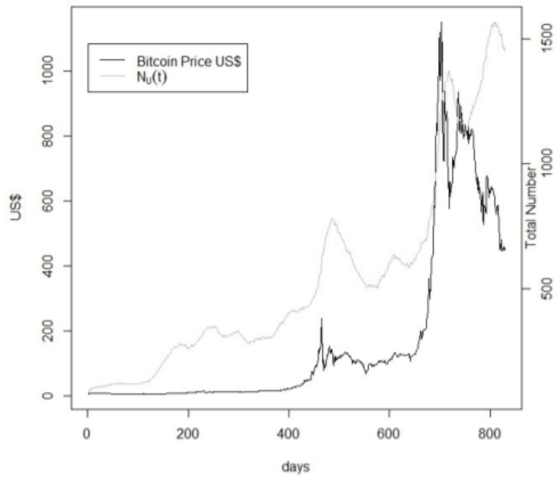
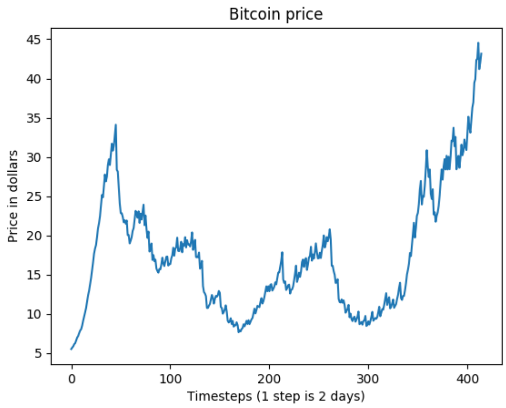


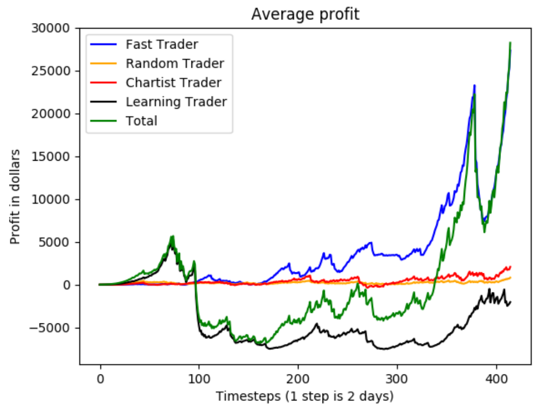
Figure 2: The real price of Bitcoins and the number of unique addresses in the market.

3.2 Different strategies

To give a comparison between the different trading strategies the profit (wealth plus the amount of bitcoins times the current bitcoin price) for each type of trader over time is plotted in Figure 3b. Besides, the average bitcoin price over 100 runs is plotted again in Figure 3a but now with the Learning and Fast Traders included. Here you can see that the plot is almost equal to the plot of Figure 1a. This makes sense because we only included one Learning and one Fast Trader so their influence on the market self would be minimal. In the discussion a detailed explanation is given about the behaviour of different trading strategies and why they make or did not make profit.



(a) The average bitcoin price over 100 runs.



(b) The average profit of each type of trader over 100 runs.

Figure 3: Simulation results when both Learning and Fast Traders are included in the model.

4 Conclusion

In the conclusion we discuss the results in Section 4.1 and will discuss the relevance of the results in Section 4.2.

4.1 Discussion

In this paper we constructed an heterogeneous agent-based model to research different strategies in Bitcoin trading. Four strategies are discussed: the Random Trader, the Chartist, the learning agent and the Fast Trader. A Bitcoin environment was simulated including an Order Book, mining mechanism and migration of traders.

4.1.1 Random Trader

For the Random Trader, we can see that its wealth is slowly increasing with the Bitcoin price. This can be explained by the fact that Random Traders do not take into account what the price is doing. The profit the Random Traders make can be explained by the climbing Bitcoin price. The Random Traders do not make much profit, but keep the price of the bitcoins more stable.

4.1.2 Chartist Trader

For the Chartist trader, we found that the profit of this trader increases over time. Since the Chartists only buy when the price is going up, their wealth gets bigger. However, when the price is at a peak and the peak is in the last few steps, the Chartist can decide to buy when the price is going down because its look-back time varies between 0 and 30 days. This means that the average over this period of time is increasing but the Chartist decides to buy when its actually decreasing. For this reason the Fast Trader is introduced.

4.1.3 The Fast Trader

The Fast Trader is by far the best performing agent of the simulation. As it checks whether a peak has been passed or not, the agent can buy when the Chartists are still willing to sell, the same happens for selling. The main point here are the peaks of the price. By doing a lot of micro transactions, the Fast Trader can always get the best prices when it's trading. The graphs show very clear that the Fast Trader is the best strategy to use.

4.1.4 The Learning Trader

Lastly, the Learning Trader only loses money. The agent has been learning on different simulations, and then put into a new simulation. In the different simulations, it learned that the Bitcoin price overall, is generally climbing. Therefore, it assumes that it is always better to buy as much as possible. This means that the Learning Trader immediately starts buying as many bitcoins as possible. Therefore the price is climbing, but when the Learning Trader stops buying, the price drops again. This results in a lot of losses for the Learning Trader, since it buys when the price is high. The fact that we included the Learning Trader also diminished the fat tail phenomenon, as the Learning Trader immediately starts buying.

All in all we can say that the Fast Trader performs best. This can be explained by the agent is the fastest to respond to changes in the climbing or falling of the price. The Learning Agent is performing very poor, due to the fact that it has learned the wrong idea.

The model we created can be used to analyze more theories. As the parameters are easily adjusted, different parameter settings can be tested. This way, the number of agents can be experimented with, to see the effect on the Bitcoin price for different ratios of strategies. Next to this, more research can be done to the Learning Trader. For this experiment, the Learning Agent was trained on data that was always ending higher than it started. In other research, the learning agent can be trained on all different crypto-currency data, which also contains currencies that have end without worth. The effect of transaction costs can also be very interesting for this simulation. This way, the profitability of the simulation and the agents can be measured. This improves the relevance and external validity a lot. To improve the external validity more, the simulation should be a more close resemblance of the real Bitcoin data. At the moment, the simulation does not show the fat tail phenomenon, which the simulation of Cocco could show. For future research it is important to get these kind of properties in the simulation.

4.2 Relevance

We found that a fast trading agent, looking only to the minima and maxima of the Bitcoin price, has the best strategy for trading. However, in the real Bitcoin market there are small transaction fees needed to trade, making it harder to do micro-transactions. To research this, transaction fees should be added to the model, to 'punish' the Fast Trader for the micro-transactions. This paper tried to show the possibility of simplicity of human trading. Whereas most traders claim to have very complex strategies for trading[2], we tried to evaluate this idea by showing that a relatively simple agent can make a lot of profit. This shows that the complexity of human trading can also reduced to small and simple rules. However, this is one explanation, of course it is possible that there are complex rules that work very well. By combining the human trader knowledge into algorithms, the trader mind can be examined to find the way in which humans trade.

References

[1] Franklin Allen and Risto Karjalainen. Using genetic algorithms to find technical trading rules. 51, 02 1993.

[2] Brad Barber, Reuven Lehavy, Maureen McNichols, and Brett Trueman. Can investors profit from the prophets? security analyst recommendations and stock returns. *The Journal of Finance*, 56(2):531–563, 2001.

[3] Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. Bitcoin: Economics, technology, and governance. *The Journal of Economic Perspectives*, 29(2):213–238, 2015.

[4] Luisanna Cocco, Giulio Concas, and Michele Marchesi. Using an artificial financial market for studying a cryptocurrency market. *Journal of Economic Interaction and Coordination*, 12(2):345–365, Jul 2017.

[5] Luisanna Cocco and Michele Marchesi. Modeling and simulation of the economics of mining in the bitcoin market. *PLOS ONE*, 11(10):1–31, 10 2016.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system.

[8] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.