國立政治大學應用經濟與社會發展英語
碩士學位學程
International Master's Program of Applied
Economics and Social Development
College of Social Sciences
National Chengchi University

碩士論文
Master's Thesis

加密貨幣設計之代理人基計算模型
Agent-Based Computational Modeling of Cryptocurrency Design

Student: Ude, Felix 吳立思
Advisor: Chen, Shu-Heng 陳樹衡

中華民國108 年7 月
July 2019

DOI:10.6814/NCCU201900519

加密貨幣設計之代理人基計算模型
Agent-Based Computational Modeling of Cryptocurrency Design

研究生：吳立思　　Student: Ude, Felix

指導教授：陳樹衡　Advisor: Chen, Shu-Heng

國立政治大學

應用經濟與社會發展英語碩士學位學程

碩士論文

A Thesis

Submitted to International Master's Program of Applied Economics and Social Development

National Chengchi University

中華民國108 年7 月

July 2019

DOI:10.6814/NCCU201900519

Abstract

Cryptocurrencies, such as Bitcoin, witnessed a surge in popularity during recent years. With the rise of attention, the discussion about a better design of these cryptocurrencies also increased, to solve issues like security problems and network congestion. Many suggested solutions require a total redesign of the cryptocurrency. This thesis looks into ways to redesign the cryptocurrency Bitcoin in a more subtle way, by only optimizing its current parameters. For that reason an agent-based computation model is used to simulate the Bitcoin market and its transaction system. Its parameters are optimized and compared to the real Bitcoin parameters. The results suggest a trade-off between security and economic efficiency, and that the real parameter values of Bitcoin are sub-optimal.

Keywords: Cryptocurrency, Bitcoin, Agent-Based Computation, Evolutionary Multi-objective Optimization

DOI:10.6814/NCCU201900519

Contents

List of Figures **III**

List of Tables **IV**

1 Introduction **1**

1.1 Research Motivation 1

1.2 Contribution 2

1.3 Organization 2

2 Literature Review **4**

2.1 Cryptocurrencies 4

2.1.1 Blockchain 4

2.1.2 Signature of Transactions 5

2.1.3 Bitcoin Mining 6

2.1.4 Development of Bitcoin 9

2.2 Economic Literature 10

2.2.1 Economic Analysis of Bitcoin 10

2.2.2 Agent-Based Computational Economics in Blockchains 14

2.3 Summary 15

3 Methodology **17**

3.1 Model Overview 17

3.2 Types of Agents 18

3.2.1 Chartist 18

3.2.2 User 19

3.2.3 Miner 19

3.3 The Model 22

3.3.1 The Bitcoin Market 22

3.3.2 The Transaction System 26

3.4 Initialization 28

3.4.1 Number of Agents and their Type Distribution 28

3.4.2 Agent's Wealth 29

DOI:10.6814/NCCU201900519

3.5 Calibration . . . 31
3.5.1 Realistic Parameters . . . 31
3.5.2 Optimization for Economic Efficiency . . . 32
3.5.3 Optimization for Economic Efficiency and Hashing Power . . . 34
3.6 Summary . . . 35

4 Findings **36**
4.1 Real Parameters . . . 36
4.1.1 Price Development . . . 36
4.1.2 Hashing Power Development . . . 37
4.1.3 Transaction Fee Development . . . 39
4.1.4 Wealth Development . . . 40
4.2 Optimized Wealth . . . 42
4.3 Optimized Wealth and Hashing Power . . . 43
4.3.1 Parameters . . . 44
4.3.2 Outcomes . . . 48
4.4 Summary . . . 51

5 Conclusion **52**
5.1 Review of Findings . . . 52
5.2 Application of Findings . . . 52
5.3 Limitations . . . 53
5.4 Future Work . . . 54

Bibliography **55**

A Table of Variables **62**

DOI:10.6814/NCCU201900519

List of Figures

1 The structure of a sequential archive 4
2 The original Bitcoin blockchain design 6
3 A Blockchain 7
4 Why you can't cheat at Bitcoin 8
5 Bitcoin Controlled Supply 9
6 Bitcoin Hashing Power and Difficulty since 2016 11
7 Number of Agents per type during Simulation 29
8 Simulated Average Bitcoin Price 37
9 Real and Simulated Bitcoin Price of a single Run 38
10 Simulated Average Hashing Power 38
11 Standard Deviation of Simulated Hashing Power 38
12 Real and Simulated Hashing Power of a single Run 38
13 Simulated Average Electricity Consumption 39
14 Simulated Electricity Consumption Standard Deviation 39
15 Simulated Average Transaction Fee 40
16 Unprocessed Transactions during the Simulation 40
17 Real Bitcoin Transaction Fees 40
18 Simulated Average Wealth per Agent Type 41
19 Simulated Wealth Standard Deviation per Agent Type 41
20 Simulated Average Cash per Agent Type 41
21 Simulated Average Bitcoin per Agent Type 41
22 Simulated Average Wealth per capita 42
23 Simulated Wealth Standard Deviation per capita 42
24 Convergence of the Optimization for Wealth 42
25 Parameters used by Generation for Wealth Optimization 43
26 Parameters found by NSGA-II for Wealth & Hashing Power Optimization 44
27 Daily Transaction Limit against Network Hashing Power 45
28 Daily Transaction Limit against Total Wealth 45
29 Daily Bitcoin Generation against Network Hashing Power 46
30 Daily Bitcoin Generation against Total Wealth 46

DOI:10.6814/NCCU201900519

31 Pareto Front of Optimization Objectives found by NSGA-II 48
32 Non-Dominated Pareto Front of Optimization Objectives 49

List of Tables

1 Agent Type Fraction 29
2 Daily Mining Reward in Simulation 31
3 Model's Stochasticity per amount of Monte Carlo Runs 34
4 Meta-Parameters for NSGA-II of Wealth Optimization 34
5 Meta-Parameters for NSGA-II of Wealth and Hashing Power Optimization 35
6 Simulation Results with Wealth optimized and real parameters 44
7 Mean and Standard Deviation of Optimized Parameters 45
8 Regression Results - Optimization Parameters against Hashing Power . 46
9 Regression Results - Optimization Parameters against Total Wealth . . . 47
10 Regression Results - Hashing Power against Total Wealth 49
11 Different Pareto Efficient Simulation Results compared with Real parameters 50

DOI:10.6814/NCCU201900519

1 Introduction

1.1 Research Motivation

Online or digital transactions have historically relied on a third party. Examples for such services are Paypal or during more recent days Apple Pay or Alipay. These providers serve as a centralized ledger keeper, which keeps track of all transactions and therefore prevents users from double spending their currency. This trust based system generally comes at a cost (Williamson, 1975) , either in form of transaction fees or as with many providers today, the provision of a user's data, hence a lack of privacy (Sholtz, 2001; Hoofnagle et al., 2012). This problem, as it was found out later, could be solved by using cryptocurrencies, which instead of a central authority, relies on a distributed ledger, called blockchain and the solving of cryptographic problems using computer hardware with an extrinsic incentive in form of new coins. The first of this kind, called Bitcoin, was invented by the person under the pseudonym Satoshi Nakamoto in 2009 (Satoshi Nakamoto, 2009), and was successful in removing the third party, hence essentially creating a trustless matter of transaction.

Even though trust is no longer necessary for transacting a currency, the system instead relies heavily on the network's security, which is considered to be equal to the combined computing power of the network (Pagnotta, 2018). Therefore, it is theoretically possible for a person or an entity to acquire enough computing power to take over the network, a so called majority attack. Thus, to make such an attack less likely, the network's computing power needs to be increased and maintained.

The security of a blockchain comes with a cost. As already mentioned, verificators receive a reward in the form of new Bitcoin, in addition to the transaction fees user attach to their transaction. During recent years the popularity of Bitcoin surged and so did transaction fees due to a higher level of congestion. With higher popularity many problems are discussed, such as the energy consumption issues due to the required hardware, the degree of decentralization with the degree of anonymity and the economic efficiency of cryptocurrencies, in comparison to the conventional trust-based systems in terms of high fees and mining costs. This thesis focuses only on the latter as well as the security issues mentioned earlier. While it appears inevitable that users of Bitcoin need to pay in some way for the service to maintain the security, it raises the question on how

DOI:10.6814/NCCU201900519

to optimally design a cryptocurrency facing these two objectives of network security and the economic efficiency.

1.2 Contribution

Many solutions to the security and efficiency problems in Bitcoin have been suggested. However, most of these require a total redesign of the cryptocurrency. This thesis aims to look into a more subtle design approach and tries to analyze the parameters of cryptocurrencies using the example of Bitcoin. The first relevant parameter for the Bitcoin currency is the block size of its blockchain, which directly influences the transaction limit of it. The other important parameter is the Bitcoin generation rate, which is the amount of Bitcoins created and given to the miners (i.e., validators) of the system at periodical time steps. The real parameters were picked by Bitcoin creator Nakamoto in 2009, who never justified both of these numbers, thus gives more reason to analyze these and see what would be better values for these parameters. For that purpose, an established agent-based model of a cryptocurrency is adapted and extended. The model includes different types of agents simulating users, chartists and miners, who interact in a near-realistic market for Bitcoin. Miners are able to invest in new hardware and sell their received Bitcoin to do so. Users and chartist buy and sell Bitcoin depending on specific rules. In addition, this thesis makes a contribution in creating a transaction system that mimics the real Bitcoin transaction system and letting users bid for their transaction in form of transaction fees. After the model has been established, a genetic algorithm is used to optimize the parameters of it.

According to the knowledge of the author, this writing is the first to analyze the trade-off between security and economic efficiency using a multi objective optimization approach. The findings of this paper suggest that there is a trade-off between network security and economic efficiency, and that no matter the preferences of designers and users of a cryptocurrency like Bitcoin, the currently used parameters are sub-optimal.

1.3 Organization

The second chapter of this thesis begins with an explanation of the technology and economy behind cryptocurrency, especially Bitcoin. It also introduces previous research on agent-based modeling in cryptocurrencies. The third chapter presents the agent-based

DOI:10.6814/NCCU201900519

model by detailing the different agent types, the market and the transaction system. This chapter will also cover the calibration of this model, which is divided into the following three parts. The first part is the calibration of the model with the currently used parameters of the real Bitcoin. It serves as a baseline to compare the effects of later improvements. In the second part, the model is being optimized for economic efficiency. In the last part, the model is being optimized for the two objectives of efficiency and network security combined. The fourth chapter presents and discusses the findings of this thesis. Finally, the last chapter concludes this thesis, summarizing the findings, mentioning its limitations and explain future work.

DOI:10.6814/NCCU201900519

2 Literature Review

This chapter has two main sections. Section 2.1 explains the technology of Bitcoin, including its blockchain or distributed ledger, its signature system, mining process and general recent developments. Section 2.2 summarizes economic literature dealing with Bitcoin, its design, limitations and economic implications. A literature review on agent-based modeling in cryptocurrencies and Bitcoin is also presented. Lastly, Section 2.3 gives a summary of the chapter.

2.1 Cryptocurrencies

The main difference between cryptocurrencies and standard online transaction services is that the former does not rely on trust or any third parties. Instead, it relies on a system of distributed verification of a ledger, which keeps track of all transaction in- and outputs (Chiu & Koeppl, 2017). This distributed ledger is often referred to as a blockchain, which is explained in the next section.

2.1.1 Blockchain

While Nakamoto might have invented the first working cryptocurrency in 2009, the underlying concept of a blockchain is much older. The first time blockchain was mentioned was by Haber & Stornetta (1991). Their idea was to implement a sequential archive, which would be able to verify the integrity of a digital file (e.g., a video or picture file). Every entry of this sequential archive includes a time stamp and a record of the file. In the Bitcoin blockchain, these files are the transactions. An entry also includes a hash value of the previous entry. See figure 1 for an example of such structure.

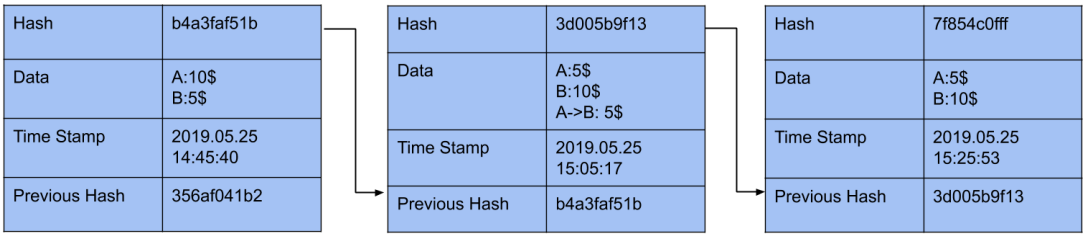

Figure 1: The structure of a sequential archive

A hash value is returned by a so-called hash function. Such functions are able to take

DOI:10.6814/NCCU201900519

any content of any length and transform it into a hexadecimal number of fixed length. The security of the blockchain relies on the fact that this transformation is irreversible, i.e., there is no way of inverting this process (Yermack, 2017). In addition, the chance that two contents have the same hash value is nearly equal to zero (Goldwasser & Bellare, 1996). Even a small modification of the content results in a complete different hash value. The consequence for a blockchain is, that if somebody tries to change a previous entry in the blockchain, s/he would also need to change all other following entries, since every block includes the hash value of the previous block and so forth (Yermack, 2017).

2.1.2 Signature of Transactions

To prevent other users from adding transactions that use one's own funds, a signature system was added to Bitcoin's protocol. It was proposed already in 1990 by Rompel (1990). Nakamoto used this idea. It works by means of asymmetric encryption with a two part key pair, one part called public key and the other private key. Each message encrypted with public part of the key can only be decrypted using the private key-part. A sender who desires to use his/her funds for a transaction, will use her/his private key and the receivers public key to sign a transaction, which is used again in a hash function, so that other users can confirm using his public address and his/her hash to show, that s/he actually received the funds at one point in time before. Looking at the blockchain all others can also confirm that s/he has not spent it yet (Satoshi Nakamoto, 2009).
Additionally, all transactions are hashed individually and those hashes are then combined and hashed again, where only the root hash is actually being added to a block. The hash values are stored in a *Merkle tree*. The structure of such *Merkle tree* is visualized in figure 2 together with a blockchain similar to the one presented in figure 1.

In order to validate transactions, it must be possible to reference all transactions ever made within the system, which requires the saving of the total blockchain on a computer's hard drive. The maximum size of a block was set to one megabyte (MB) by Nakamoto in 2010 (Yermack, 2017; Pappalardo et al., 2018). This block size limit of one MB, given an average transaction size of 250 bytes (Bitcoinfees.info, 2019) implies that the total transaction limit per block is set to around 4000. The maximum size of a

DOI:10.6814/NCCU201900519

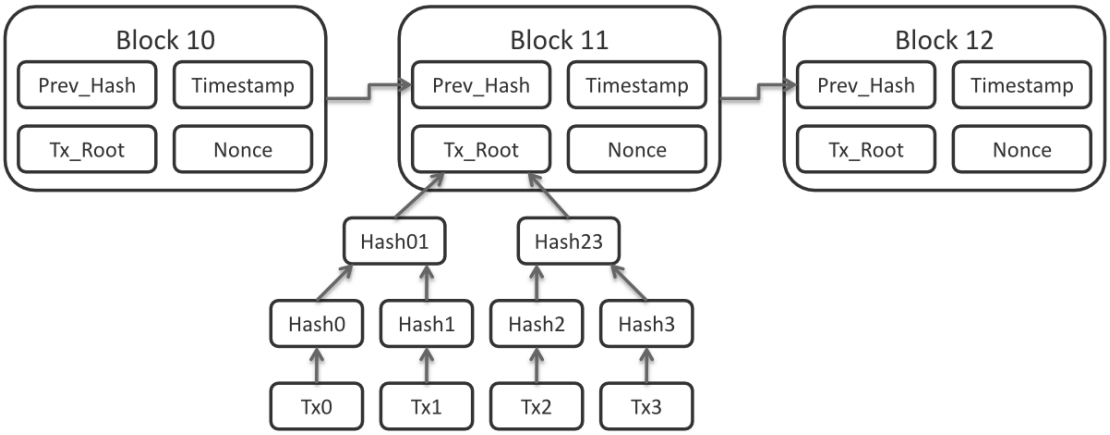

Figure 2: The original Bitcoin blockchain design
The blockchain as designed by Nakamoto with a Merkle tree for transactions.
WikiMedia (2013)

block was chosen arbitrarily and will be discussed in a later section.

Lastly, the total required disk space is growing steadily. Nakamoto predicted, given that there were no transactions, the blockchain size would grow by 4.2 MB a year. Since all the newly created blocks do have transactions included, the Bitcoin blockchain is actually currently growing by an increasing speed, with a growth of 50 GB since last year (as of 2019). The current size of the blockchain is close to 220 GB at the moment (Blockchain.com, 2019b).

2.1.3 Bitcoin Mining

A blockchain allows to verify the integrity of a file or in the case of Bitcoin the integrity of transactions. However, Haber & Stornetta (1991) stressed that in order to realize such a ledger system, it would need to be decentralized, meaning that each modification to the chain would have to be broadcasted into the public, so that everyone could access it and therefore control its integrity. A solution to this would be something like the inclusion of newspaper articles tp produce the hash values. Because newspaper articles are assumed to be hard to alter after they have been printed and because they are available in public this guarantees a high level of immutability of a sequential ledger. The immutability is an important feature for any cryptocurrency as it prevents anyone from changing the transaction history in their favor. Nakamoto however used a different approach to solve this problem. Instead of using real life events published in a newspaper, he implemented a ”Proof-of-Work”-system, which was inspired by the *hashcash* idea from Back (2002).

DOI:10.6814/NCCU201900519

It involves the solving of a cryptographic problem, that is the search for a value, called *nonce*, which when included in a hash function, returns a hash value with a specific number of leading zeros. This guarantees that once a nonce has been found and the nonce produced hash value is included in a block, the content of the block can only be changed with redoing the mining. The chance of finding this nonce would only increase with the increase of computational power, i.e., the hashing capability of the network. However, the difficulty of this task was set by Nakamoto, so that on average every 10 minutes a new block can be mined. The difficulty will adjust dynamically, if the actual mining time differs from ten minutes (Satoshi Nakamoto, 2009) . The difficulty is defined as:

$$\text{Difficulty} = \frac{\text{Expected time for mining 2016 blocks}}{\text{Last time for mining 2016 blocks}} \tag{1}$$

where the expected time for mining 2016 blocks is exactly two weeks. Therefore, on average the difficulty will be adjusted that, given a constant network power, the next 2016 blocks would need two weeks to be mined.

Since every miner mines his own head block, there are actually more than one chain head at any given time. The majority decides which blockchain is the correct one by choosing the longest one, i.e., the chain with most blocks. Sometimes, there might be a disagreement between miners on which chain is the longest. This can happen, when two or more miners at the same time mine a new block.

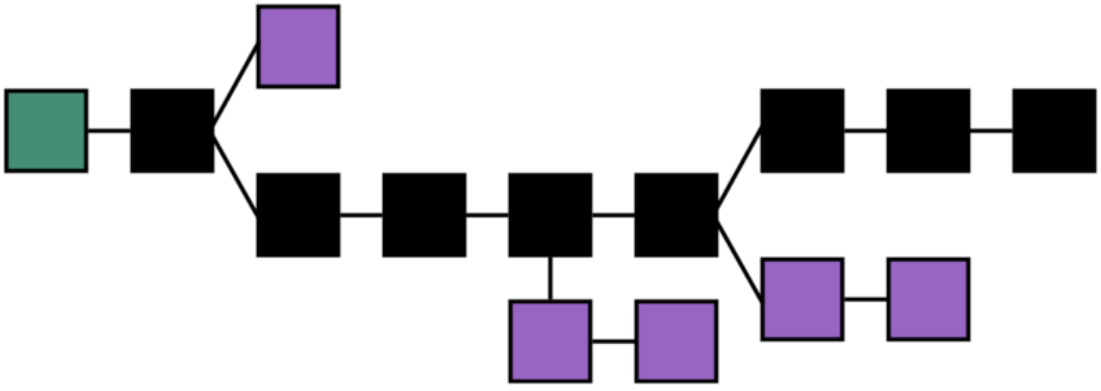

Figure 3: A Blockchain
A blockchain with the genesis block in green, the longest chain in black and the orphan blocks in purple.
Bitcoin Wiki (2019a)

DOI:10.6814/NCCU201900519

As an example shows in figure 3, these miners continue to mine the next block, which leads to more than one head, this process is called forking. Eventually a forked chain will be longer than all others and the network will agree to use that chain (here in black). The miner who lost the race, will undo his changes, abandon his version of the chain and also use the longest chain from now on. The blocks, which are not used anymore as a consequence of this, are referred to as orphan blocks (here in purple) and will be discarded by the miner (Decker & Wattenhofer, 2013).

A blockchain like this has the advantage that if a malicious users wants to change a previous block, s/he not only has to rework that block, but also all the blocks following, in the same time as everyone else is solving the newest block. Figure 4 summarizes this well.

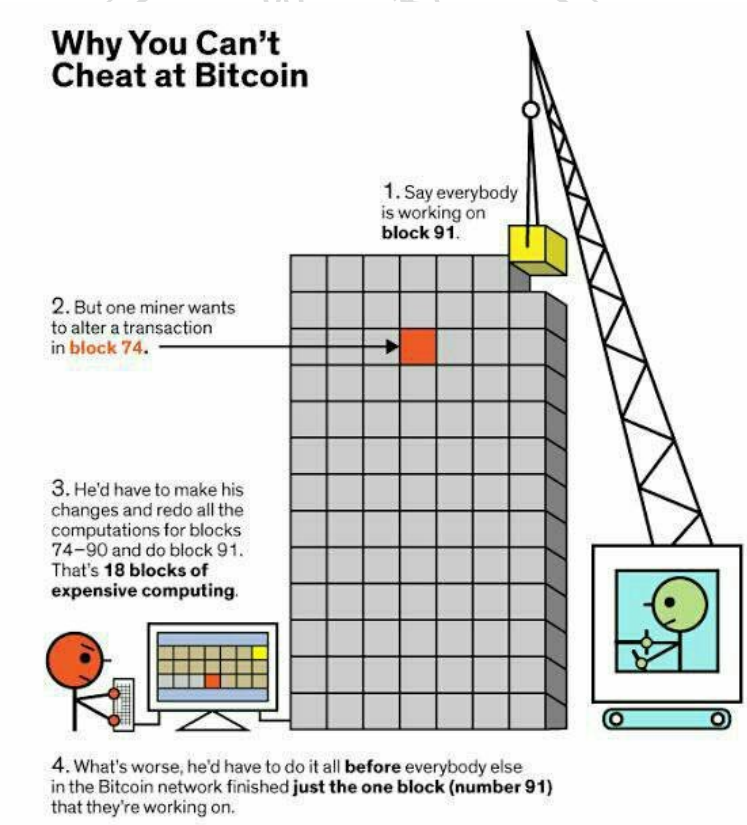

Figure 4: Why you can't cheat at Bitcoin
Montgomery (2015)

A requirement for the blockchain to work is that a sufficient number of miners are actually mining at any given time. Technically one miner would be enough to run the protocol by himself. But the higher the number of miners, the higher the degree of decentralization. Also, the more miners, the higher would also be the hashing power of

DOI:10.6814/NCCU201900519

the Bitcoin protocol. This directly effects the security of the blockchain, as a malicious miner would need more capital to buy enough hashing power to outmine all other miners in the system. In order to incentivize the mining, Nakamoto added a reward for it in form of new Bitcoins. "The steady addition of a constant of[sic] amount of new coins is analogous to gold miners expending resources to add gold to ciculation." (Satoshi Nakamoto, 2009) Hence, the name mining was introduced shortly after. Furthermore, he made it possible to attach transaction fees to transactions. Nakamoto forsaw that an incentive would increase the networks support. He initially set the Bitcoin reward to 50 Bitcoins for mining one block and also determined that this reward should half every 210,000 blocks, which is roughly equal to 4 years. As this is equal to a geometrically decreasing function, the rewards will eventually run against zero and the overall supply of Bitcoins will stop growing at around 21 million bitcoins as shown in figure 5. This implies that in the future the incentives to mine the nonce are solely depending on the transaction fees.

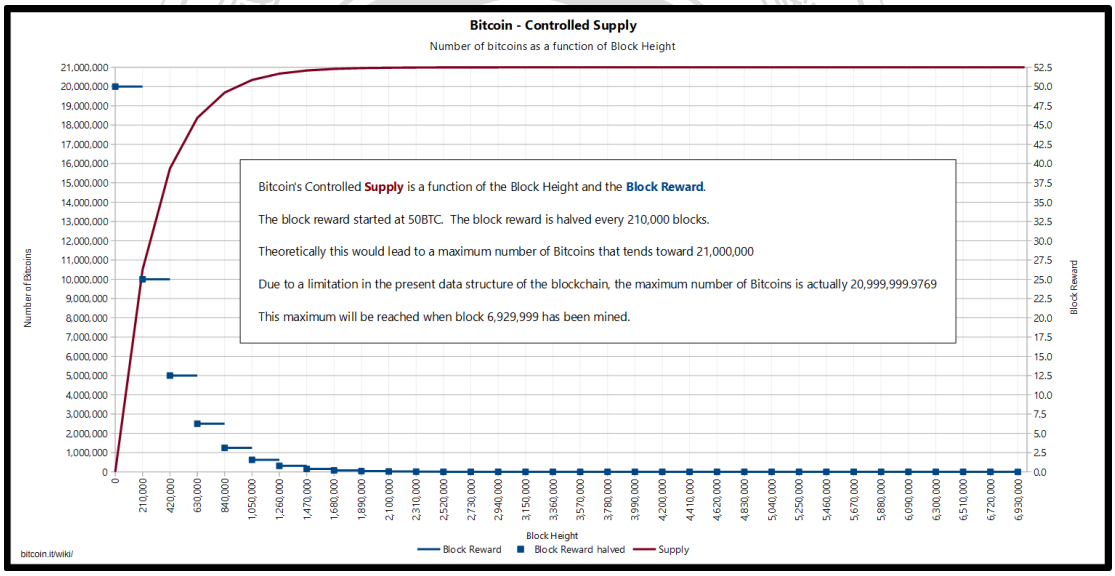

Figure 5: Bitcoin Controlled Supply
Bitcoin Wiki (2019b)

2.1.4 Development of Bitcoin

Similar to stock exchanges, the Bitcoin professional online exchanges allowed many people to purchase Bitcoins with fiat currencies, just like any other currency do. A rising demand during the last years has increased the value of Bitcoin. This increase in price also increased the mined block reward expressed in US-Dollar. Even with the

DOI:10.6814/NCCU201900519

lowest price during the last two years of around 2,000 US-Dollars, with the inclusion of transaction fees, there is a mining reward of 5 million US-Dollar every block (i.e., every ten minutes) (Blockchain.com, 2019h). This led to an arms race by the miners, who kept investing in their hashing capability in order to increase their chance of finding the nonce in the next block and therefore receiving the block award. As mentioned in the previous section, the difficulty of finding the nonce adapts dynamically to the hashing power of the network. Figure 6 shows the development of this difficulty and the corresponding network hashing power from mid 2016 to mid 2019. With the rising investment occurred another phenomenon. In order to decrease their risk, miner begun to work together in finding a nonce. This collaboration is called a *Mining Pool*. These pools make up to around 90.4% of the total network as of today (Blockchain.com, 2019d).

The raise of hashing power was accompanied by a rise of energy consumption. During its peak time in August 2018, the total Bitcoin network had an estimated energy consumption of 73 tera watt hours per year (Digiconomist.net, 2019). For comparison: That is more than the whole country of Austria consumed in 2016 (CIA, 2016). Due to the surge in demand since 2017, the network also witnessed a huge congestion problem. Since the Bitcoin protocol has a limited block size capacity, which was discussed in the previous section, the transaction fee rose substantially. The mean transaction fee increased to 1.89 US-Dollar. However, for short periods of time, the transaction fee did increase to 37.49 US-Dollars per transaction (Bitcoinfees.info, 2019).

2.2 Economic Literature

2.2.1 Economic Analysis of Bitcoin

With raising popularity of cryptocurrencies and Bitcoin, the amount of economic literature on Bitcoin also grew. This subsection will introduce several topics. First, it introduces literature on Bitcoin adoption. The second part will introduce the literature on mining games including selfish mining. The third part of literature will cover transactions and their fees. The fourth part introduces the literature on security concerns in Bitcoin and the last part will cover the literature of design issues in Bitcoin.

This first part will begin discussing the literature on Bitcoin adoption as a currency. Luther (2016) writes that due to switching costs, the widely acceptance of Bitcoin is

DOI:10.6814/NCCU201900519

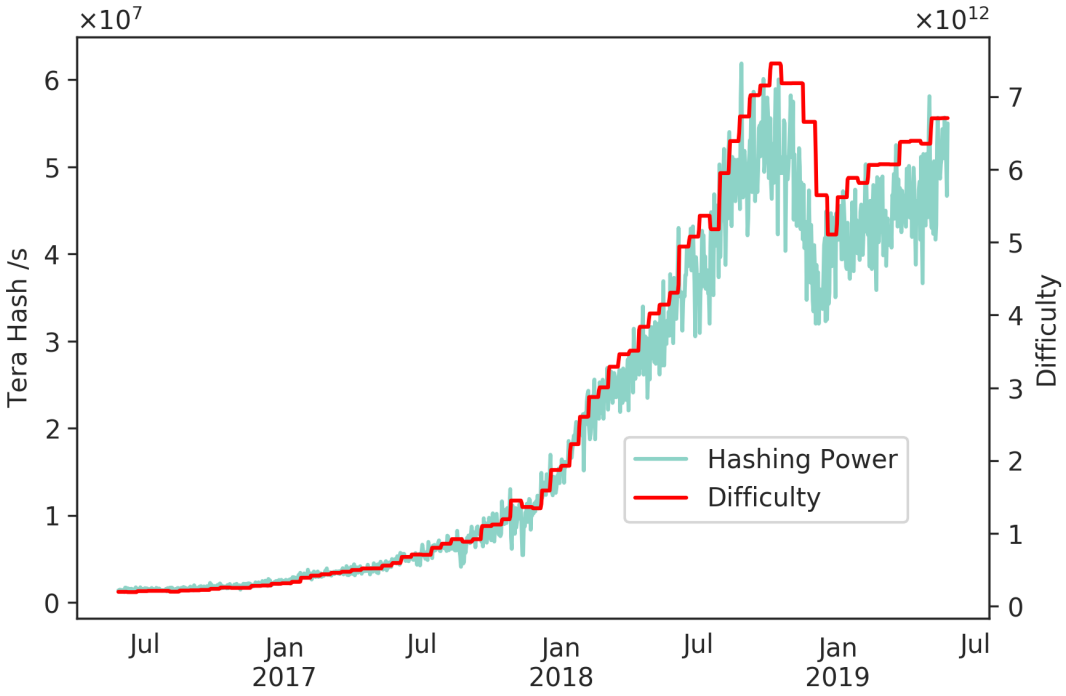

Figure 6: Bitcoin Hashing Power and Difficulty since 2016
Data from Blockchain.com (2019c) & Blockchain.com (2019a)

unlikely, unless there exists governmental support or the occurrence of hyperinflation. This is due to network effects and a possible lack of coordination. Since his writing, an example for a broader adoption due to hyperinflation actually has been witnessed in the case of Venezuela, where inflation rates of over 300,000% (Di Salvo, 2019) drove people towards the adoption of Bitcoin. Athey et al. (2016) analyzed the geographic distribution of Bitcoin users and they were able to differentiate them by their behavior patterns and times of usage. Catalini & Gans (2016) believed that Bitcoin and other cryptocurrencies theoretically can reduce the costs of verification and networking, hence the blockchain is challenging existing business models and opens up new ones at the same time.

A larger amount of research is committed to Bitcoin mining and its games. Altman et al. (2018) analyzed the mining procedure from a congestion game approach. They believed, that more miners increase the security of the overall network of a blockchain, since a higher hashing capability makes it harder for a single malicious miner to take over the blockchain. This higher robustness could increase the real value of the underlying cryptocurrency. On the other hand, the more miners there are on a given chain,

DOI:10.6814/NCCU201900519

the higher the negative mining externalities for all miners, because everyone's chance of mining a block (finding the correct nonce) decreases with the amount of miners in the network. This rises the risk of the investment of miners, find Cong et al. (2018). They write that as a consequence miners are inclined to collaborate in mining pools in order to decrease their overall risk. Since a bigger pool leads to a lower risk, mining pools tend to centralize. Nonetheless, they self-regulate themselves in not becoming too big, as in fear of a majority attack, the whole currency might collapse, self damaging the pools.

Another aspect of mining falls under the term of *selfish mining*, where the assumption, that every miner always mines the newest block and broadcasts his findings immediately, is dropped. Carlsten et al. (2016) say that only relying on transaction fees, as it is currently planned in the future of Bitcoin, would harm the overall security of the Bitcoin network, as miners would only mine a block when the transaction fees in prospect would outvalue the mining cost. They state that their hypothesis only holds in a world where the block size remains vastly larger than the demand for transactions. Given the developments of recent years, it can be said though that this assumption is not always true. Goren & Spiegelman (2019) confirm that selfish mining, or *smarter mining* as they call it, will lead to a reduction of the network security, but also has the potential to decrease the energy consumption as a consequence. Sapirshtein et al. (2016) find that selfish mining is a real threat to Bitcoin. Given that the total amount of miners and their hashing power is not too large, it becomes profitable and feasible for miners to mine selfishly. The selfish mining strategy outlaid by the authors includes an intentional forking of the blockchain in order to force other miners to abandon their currently mined block.

The following literature focuses more on the transactions and their fees inself. Kroll et al. (2013) say that they do not believe that transaction fees will play any important role in the future of the Bitcoin ecosystem, given the current protocol, since miners will always underbid their competitors and therefore decreasing the fees.[1] Easley et al. (2019) included the waiting time for a transaction to be picked up by a miner in their analysis and concluded that even if there is a high amount of transaction fees that this

[1]It appears, that the authors did not foresee the increase of demand of transactions on such a big magnitude as one could observe during recent years.

DOI:10.6814/NCCU201900519

does not guarantee a long term equilibrium for the network security. If the waiting time for a transaction of users becomes too long, they might leave the blockchain all together in order to optimize their personal utility in a similar fashion as a miner which leaves a blockchain to optimize his revenue. Therefore, fees are also a way to balance the demand for transactions and decrease the waiting time for users, which gives the transaction fee a crucial role to stabilize the network security. Pappalardo et al. (2018) point out that the Bitcoin protocol in its current form is not efficient. Miner tend to ignore transactions with smaller fees, leaving 42% of all transactions still unprocessed after the first hour of issuing and discover that 20% of transactions still have not been processed after 30 days. The majority of these transactions have a small volume, therefore the nominal efficiency of the Bitcoin blockchain is still on an "acceptable" level. However, the authors questions how well Bitcoin is suited as a time stamping system for small transactions with a condition as described.

There is also a focus on security concerns in the literature. Pagnotta (2018) writes that the Bitcoin price and the network security are positively correlated and jointly determined. Users value a higher security of the network and are willing to pay a higher price for it, while miners respond positively to a higher Bitcoin price and invest in more hashing power, therefore increasing the network security. Budish (2018) believed that the continuous flow of payments for miners must be large enough to prevent a majority one-off attack of the system. Buterin (2016) states that the costs for security in form of the verificators or miners are inevitable. He argues that there should be an optimal combination of inflation and transaction fees which minimizes the dead weight loss for the transaction market. Furthermore, he discusses other ways for the redistribution of the transaction fees. He proposes the idea to either distribute the fees among all miners no matter who mined a block or to burn the fees all together (therefore appreciating the currency). He concludes that eventually it is a tradeoff. If the fees are distributed, there is a high level of certainty of Bitcoin supply, but the level of security is less certain and if the fees are burned the Bitcoin supply is less certain, but the level of security is.

The last section of economic research focuses on the design questions in regard to scalability and economic performance of Bitcoin and blockchain. According to Ma et al.

DOI:10.6814/NCCU201900519

(2018), free entrance to the Bitcoin mining market is the main cause for the high energy consumption. The authors suggest two possibilities to reduce the energy waste. Firstly, restrict miners from entering the blockchain freely.[2] Alternatively, removing the inflation (i.e., the new Bitcoin generation) and relying solely on transaction fees. Croman et al. (2016) find that a reparameterization of the current Bitcoin protocol should only be the beginning and is not sufficient to solve the scalability problem of Bitcoin in the long term. They propose a variety of technical improvements and additions to the protocol that might be better suited to address Bitcoins problems. Huberman et al. (2019) also find that Bitcoin does not scale enough with its current protocol. They argue for a flexible block size limit, which reacts to the demand of transactions. Therefore, balancing the delay costs for users and the revenue for the miners. Lastly, Chiu & Koeppl (2017) analyze how well Bitcoin is serving as means of payment. Their analysis finds that with peak demand, Bitcoin is 500 times more costly than other traditional currencies. Given the welfare as the overall surplus of a modeled market, using Bitcoin as a currency, the welfare loss totaled with 1.41%. The authors then proceed to establish an equation-based model to calculate the parameters for optimal welfare and conclude that a Bitcoin protocol with zero transaction fees and a small inflation would minimize the mining costs and maximize the total welfare.

2.2.2 Agent-Based Computational Economics in Blockchains

The literature on agent-based computation of blockchains or cryptocurrencies is comparatively small. Terna et al. (2016) simulated the adoption of Bitcoin wallets. The authors adapted an epidemic model lent from health care research to analyze the spread of trust and adoption of Bitcoin. One of their findings is the lack of trust in cryptocurrency which is correlated with the amount of transactions and the amount of agents in an agent's proximity that uses a cryptocurrency as well. Due to a lack of confidence in cryptocurrencies, agents do not like to hold a cryptocurrency for a long time, selling it therefore quickly. This leads to a higher amount of transactions, which then allows for the generation of confidence in cryptocurrencies in the model.

Cocco et al. (2017) modeled the Bitcoin blockchain with a trading system using two kinds of agents. The first one called random trader, trades Bitcoin for diversification or

[2] This would imply a smaller degree of decentralization of the blockchain.

DOI:10.6814/NCCU201900519

liquidity, similar to the zero-intelligence agent by Gode & Sunder (1993) without any real constraints besides their own budget. The other category is called chartist, which used different strategies involving genetic algorithms to optimize its revenue. They could confirm that the daily trading returns of absolute prices are indeed autocorrelated and follow a negative exponetialy shaped cumulative distribution function with a fat tail. In their next work, Cocco & Marchesi (2016) modeled the Bitcoin blockchain including the mining system. It added the new agent type miner. This category could acquire mining hardware and generate new Bitcoins. Its goal is also to maximize its revenue. The authors were able to reproduce stylized facts, such as the unit root and volatility clustering. This thesis is based on this model. In their newest publication, Cocco et al. (2019) solely focus on the mining aspect of Bitcoin and ignores the trading aspect. It compares the profitability of mining Bitcoin to that of mining gold and comes to the finding that mining Bitcoin is superior compared to mining gold. The authors state that if everything included, the costs of using Bitcoin might actually be smaller than those of a traditional currency.

Zhou et al. (2017) use a simple agent-based model to try to capture the effect of the amount of agents. Their findings, while limited, say that the amount of agents directly affects the Bitcoin price and the deal rate of transactions. A high number of agents decreases the price fluctuation and the overall deal rate is higher. Finally, Lee et al. (2018) first use inversed reinforced machine learning to deduct trading rules from the real Bitcoin transactions. In a second step they applied these rules to an agent-based model in order to forecast the price and performance of Bitcoin. They were able to establish an 80% directional accuracy of their price prediction within the first six days of forecasting.

2.3 Summary

This chapter gave an introduction to the technology and design of Bitcoin, including its blockchain and parameters such as the block size and the Bitcoin generation. It also discussed current developments of Bitcoin and the resulting problems, such as congestion, high electricity consumption and security concerns. The second part summarized the economic literature of cryptocurrencies and Bitcoin. Several papers come to the conclusion, that the current protocol is not efficient from different point-of-views. Dif-

DOI:10.6814/NCCU201900519

ferent approaches have been theorized and proposed, such as totally new protocols and different parameters for the blockchain. Lastly, some literature has been introduced on agent-based modeling of a cryptocurrency. The model used in this thesis will be explained in the next chapter.

DOI:10.6814/NCCU201900519

3 Methodology

In order to get more insight into the design of a cryptocurrency and its calibration, an agent-based model is created. The goal of the model is to find a better way to design a cryptocurrency that allows for a high security and high economic efficiency simultaneously in a Pareto optimal fashion. Such an agent-based model approach has been shown to be successful in the economic literature on policy design. Chen & Chie (2008) for example used an agent-based model for the design of a lottery market, while Marks (2006) introduces market design in agent-based modeling in a more general fashion.
In the first step, a model was created that tries to simulate the real cryptocurrency Bitcoin. This chapter will describe the model and its agents in detail.[3] Section 3.1 will begin describing the model by giving an overview of its core features and agents. Section 3.2 will explain the different types of agents used. Section 3.3 will describe the model with its market and transaction system. Section 3.4 describes the initialization of the model. After the model has been created, it will get calibrated and optimized which is described in Section 3.5. Lastly, in Section 3.6, the chapter will be summarized.

3.1 Model Overview

The model used is based on an existing one by Cocco & Marchesi (2016), which was then adapted and extended. The core features of the original model include a Bitcoin market, where agents can sell and buy Bitcoin in a realistic fashion. Furthermore, agents are able to create (i.e., mine) new Bitcoins. The full list of the features of the original model is:

- Three kinds of agents, including chartists, random traders/users and miners;
- A realistic order book to imitate a real cryptocurrency market;
- Agents join over time and decide to engage in the trade and/or to mine Bitcoins;
- A power law wealth distribution for existing and later added agents;
- Miners are all in mining pools, meaning that they have a steady stream of income. This is because since 2010 miners collaborate in groups to share their hashing

[3]The full code can be found under `https://www.comses.net/codebases/3d2dfc87-78e0-47ff-ad8f-7c8f5f13fcdd/releases/1.0.0/`

DOI:10.6814/NCCU201900519

power to decrease their investment risk by avoiding unnecessary mining (Cocco & Marchesi, 2016). The miner agents also can decide at given time points to invest in or divest their hardware.

This list of features was then extended by a transaction system, similar to the one used in Bitcoin as of today. It includes a transaction capacity limit, that allows users to submit transactions and set a fee for their transactions as well as letting miners take fees in general into their investment consideration. This transaction system allows us to analyze the development of the transaction fees as well as a better representation of the mining network hashing power. The model simulates each day between February 23rd in 2014 and April 3rd 2018 in a single time step, totaling 1500 simulation steps, where one step is equal to a whole day. This particular time frame was choosen as it was used by the creators of the underlying model from Cocco et al. (2017). It also captures the time before and during the time of Bitcoins highest congestion around the end of 2017 (Bitcoinfees.info, 2019).

3.2 Types of Agents

The model used in this thesis includes three types of agents: Chartists, users and miners. These will be explained in the following sections in more detail.

3.2.1 Chartist

A chartist is an agent who trades for profit. S/he will place a buy order when s/he speculates that prices will rise and a sell order otherwise. Both of these orders are place on the artificial exchange market. Every chartist has a time window T, that has a mean of 20 and a standard deviation of 1[4] to be used for buy and sell decisions. This approach has been firstly introduced by Arifovic (2002). If the price increases in T was over the threshold of 0.01, s/he will place a buy order and a sell order if otherwise. 10% of the chartists follow a contrary strategy, meaning the chartist will sell when the price increase is over 0.01 and buy if it is not the case. Every chartists entering the market always places a buy order. This approach can also be found in Raberto et al. (2003). In contrast to the user agents, chartist do not participate in the transaction system.

[4] A list of all variables used in this thesis can be found in appendix A.

DOI:10.6814/NCCU201900519

3.2.2 User

A user is an agent who participates in the market either for diversification of his/her portfolio and/or for the usage of Bitcoin as a form of currency. S/he does not follow any trends with his/her orders. In fact, the probability for sell and buy order is the same for a user at any given time after s/he has entered the market. Every user who is entering the market, issues a buy order. Since one of the reasons for his/her participation in the market is the usage of Bitcoin as a currency, s/he also has the ability to make a transaction using his/her own Bitcoin funds, at any given time step. Since the capacity of transactions per time step is limited, s/he will always add a fee that allows the transaction to be picked up by miners. More on transactions is provided in section 3.3.2.

3.2.3 Miner

Miners are agents that participate in the model by generating new Bitcoins with their mining hardware. They only participate in the market by selling their Bitcoins, if they require more cash in order to invest in hardware or to pay their electricity bills. At the beginning of the simulation, the total hash power of the total Bitcoin network was 28,314 giga hash/s (GH/s) (Blockchain.com, 2019e). This number, as well as other following variables, have been divided by the factor of 5000. This was done as to adjust the model, so that it could be run within computational limits. This hash rate was then divided by the expected initial amount of miners (i.e., 9) to calculate the initial hashing capability of one miner, resulting in a hashing capability of 0.629 GH/s per miner[5]. The hashing power of available hardware is getting cheaper over time, as the technology continues to improve, whereas the power consumption per hashing capability is decreasing over time. Cocco & Marchesi (2016) collected the average hashing power for the existing hardware with corresponding prices and electricity consumption in the time frame between 2011 and 2015. They estimated two curves through their aggregated data with the following forms:

$$R(t) = 8.635 \cdot 10^4 \cdot e^{0.006318 \cdot t} \tag{2}$$

[5] 28,314GH/s / 5,000 / 9 = 0.629 GH/s

DOI:10.6814/NCCU201900519

where $R(t)$ is the average hash rate per second and US-Dollar with $\frac{H}{s \cdot \$}$.

$$\xi(t) = 4.649 \cdot 10^{-7} \cdot e^{-0.004055 \cdot t} \tag{3}$$

where $\xi(t)$ is the average electricity consumption for one hour in Watts per hash and second with $\frac{W}{H/s}$.
$\xi(0)$ times the initial 0.634 GH/s per Miner gives the initial electricity consumption per miner with 1697.11 W. New miners entering the market as well as existing miners, who are able to upgrade their hardware use these two equations to determine their new purchased hashing power $r_{i,u}(t)$ (with i for the miner and u for the hardware) and its corresponding electricity consumption $e_{i,u}(t)$. The total hashing power of a miner is consequently the sum of all his/her purchases or

$$r_i(t) = \sum_{s=t_i^E}^{t} r_{i,u}(s)$$

where t_i^E is the entry time of a given miner. The corresponding electricity consumption can be expressed as:

$$e_i(t) = \sum_{s=t_i^E}^{t} \epsilon \cdot \xi(s) \cdot r_{i,u}(s) \cdot 24 \tag{4}$$

where ϵ is the electricity price per W/h. It is fixed to 0.07 US$, which was the average during the model time frame in China, where the majority of miners today are located (CEIC, 2019). The 24 refers to the hours of the day, since one time step in the model is a whole day. Miners who are already in the model and make the decision to invest, use a fraction $\gamma_{1,i}(t)$ of their fiat cash. Additionally, they will sell a fraction $\gamma_i(t)$ of their Bitcoin and use the revenue as well. Hence, the new purchased hardware will have a hashing power of:

$$r_{i,u}(t) = [\gamma_{1,i}(t)c_i(t) + \gamma_i(t)b_i(t)p(t)]R(t) \text{ ; if } t > t_i^E. \tag{5}$$

where c_i is the fiat cash and b_i the Bitcoins held by Miner i and $p(t)$ is the current Bitcoin price.
Miners entering the market will always make a purchase of mining hardware, this time only using the fraction $\gamma_{1,i}(t)$ of their fiat cash. Hence, the hashing power of the initial

DOI:10.6814/NCCU201900519

bought hardware for miners is equal to:

$$r_{i,u}(t) = \gamma_{1,i}(t) c_i(t) R(t) \text{ ; if } t_i^E > 0 \tag{6}$$

The values for $\gamma_{1,i}(t)$ and $\gamma_i(t)$ are the same as in Cocco & Marchesi (2016) and both have a log normal distribution. $\gamma_{1,i}(t)$ has a mean of 0.15 and a standard deviation of 0.15, whereas $\gamma_i(t)$ has a mean of 0.175 and a standard deviation of 0.15. Since miners are not allowed to take a credit, their γ are always set to 1 in case that $\gamma > 1$.

Miners are not able to purchase at any given time. Instead, they make a decision whether to invest or divest on average every 60 days, which is another value taken from Cocco & Marchesi (2016). This average has a normal distribution and a standard deviation of 6 days and the next decision time step is determined by each individual miner every time an investment decision has been made by him/her. Since by assuming all miners belong to a mining pool, their generated revenue is proportional to their hashing power in comparison to the total networks hashing power. The total daily income of miners is the sum of the newly mined Bitcoins per day plus the total sum of transaction fees within the transaction limit submitted at that time step, which can be expressed as:

$$b_{Tot}(t) = B(t) + \sum_{j=1}^{tlimit} f_j(t) \text{ ; if } L(t) \geq tlimit \tag{7}$$

$$b_{Tot}(t) = B(t) + \sum_{j=1}^{L(t)} f_j(t) \text{ ; if } L(t) < tlimit \tag{8}$$

where $B(t)$ is the amount of Bitcoins mined per day, $f_j(t)$ is the fee of transaction j at time t and $\sum_{j=1}^{tlimit} f_j(t)$ or $\sum_{j=1}^{L(t)} f_j(t)$ is the total sum of fees depending if $L(t)$, the amount of transactions in that time step, exceeds the transaction limit or not.

This revenue is now distributed among the miners in proportion to their hashing power.

$$b_i(t) = \frac{r_i(t)}{r_{Tot}(t)} b_{Tot}(t) \tag{9}$$

where $r_{Tot}(t)$ is the networks total hashing power.

At any decision time step, every miner decides whether the revenue of the new hardware

DOI:10.6814/NCCU201900519

outweighs the connected costs. This constraint is expressed as follows:

$$e_{i,u}(t) < \frac{r_{i,u}(t)}{r_{Tot}(t)} b_{Tot}(t) p(t) \tag{10}$$

where $r_{i,u}(t)$ is the hashing power and $e_{i,u}(t)$ the electricity consumption of hardware u of miner i. If s/he decided against the investment, s/he will issue a sell order equal to $\frac{\gamma_i(t)}{2} b_i(t)$ to cover his/her electricity expenses.

Additionally, every miner is able to divest his/her hardware for the price of $R(t)$. S/he will do so, if the electricity expenses are 20% larger than the revenue associated with that specific hardware. That means a miner keeps track of all his/her hardware individually. The constraint for divest is fulfilled if:

$$1.2 e_{i,u}(t) \geq \frac{r_{i,u}(t)}{r_{Tot}(t)} b_{Tot}(t) p(t) \tag{11}$$

3.3 The Model

The model used in this thesis has two main components:

- A Bitcoin market which is similar to a real life cryptocurrency exchange, where traders can place buy and sell orders.
- A transaction system, which allows users to send Bitcoins to other users and attach a transaction fee to it.

The following sections will explain these components in greater detail.

3.3.1 The Bitcoin Market

The first main component of the model is the Bitcoin market. It is modeled after real life Bitcoins exchanges or other trading exchanges in general. More specifically, an order book is created. This approach has also been used previously in Raberto et al. (2005) and Cocco & Marchesi (2016).

Order Book

The order book keeps record of all the buy and sell orders with their respective amount in Bitcoin, the remainder of the transaction for partial transactions, the corresponding

DOI:10.6814/NCCU201900519

limit prices and the expiration time of an order. The following will explain these terms in more detail.

Buy Order: The amount of a buy order is proportional to the available fiat cash $c_i(t)$ of a trader i. That excludes any fiat cash that is currently used in pending orders. The amount of a buy order is defined as:

$$buy_i(t) = c_i(t) \cdot \beta \tag{12}$$

where β is a variable pulled from a log normal distribution with average 0.25 and standard deviation of 0.2. In the case that $\beta > 1$, β is set to 1, since a trader is not able to take any credit.

Sell Order: Similar to equation (12) is the amount for a sell order determined.

$$sell_i(t) = b_i(t) \cdot \beta \tag{13}$$

β is the same as in equation (12), $b_i(t)$ is the Bitcoin held by agent *i*. Short selling is also not possible.

Limit Prices: The limit price is a price at which a trader wishes to perform his/her order. This mechanism works similar to a limit price from a real world exchange platform. A limit price can also have a value of zero, in which case it is regarded as the market price, meaning that a trader wishes to perform his/her order with the best available price. Each agent category has their own probability to issue an order with a market price. 0.2 for users, 0.7 for chartists and 1 for miners. These numbers are chosen and picked by Cocco & Marchesi (2016), who argue, that chartists and miners are the types of agent who have the greatest desire to trade at the best price, chartists for profit and miners for liquidity reasons.

An agent is willing to perform his/her buy order only if the sell limit is below or equal to his/her buy limit. The limit price for a buy order is given as:

$$buylim_i(t) = p(t) \cdot N_i(1.05, \sigma_i) \tag{14}$$

DOI:10.6814/NCCU201900519

Similarly, an agent is only willing to perform his/her sell order if the buy limit is greater or equal to his/her sell limit. The limit price for a sell order is given as:

$$selllim_i(t) = p(t)/N_i(1.05, \sigma_i) \tag{15}$$

For both equations (14) and (15), $p(t)$ is the current Bitcoin price and $N_i(\mu, \sigma_i)$ is a Gaussian distribution with a mean of 1.05. This distribution with this mean is supposed to represent limit prices, that are not fully rational. Additionally, agents are on average willing to pay a small amount more than the market price would suggest. σ_i is supposed to represent the volatility of the price in the model. This approach has been used in Raberto et al. (2001). The standard deviation of the Gaussian distribution is given by:

$$\sigma_i = K\sigma(\varphi_i) \tag{16}$$

where K is a constant set to 2.5 and $\sigma(\varphi_i)$ is the standard deviation of the price in the time window φ which is set to 20. Furthermore, σ_i has the following constraint:

$$\sigma_{min} = 0.003 \leq \sigma_i \leq 0.01 = \sigma_{max} \tag{17}$$

where the same values as in Cocco & Marchesi (2016) for σ have been used.

Expiration Time: Every order has an associated expiration time. It has the following form:

$$Exp_i(t) = Round(t + pat_i) \tag{18}$$

where t is the day of the order and pat_i is the patience of the ordering agent i. It is 0 for chartists, meaning their order will expire if not performed in that time step. This is the case since chartists wish to follow market trends. Users have a log normal distribution from which they retrieve their patience for each time they issue an order. It has a mean of 3 and a standard deviation of 1. Miners who always issue market price orders, have an infinite patience, meaning their order will stay in the market until it is executed and can not be cancelled (even though the investment decision would be different at a later time). $Round$ is a function that always rounds the expiration to the nearest integer.

Price Clearing

DOI:10.6814/NCCU201900519

The price clearing mechanism used in this model is similar to the one from Raberto et al. (2005).

In every simulation step, after all agents were able to issue their order, the order book is sorted in respect to the limit prices. For buy orders, the list is sorted in descending order, and for sell orders in ascending order. The model now compares the two top orders from the buy and the sell order list. A match occurs if at least one of the following conditions is fulfilled:

$$buy_i(t) \geq sell_j(t) \tag{19}$$

$$sell_j(t) \equiv \text{"0"} \tag{20}$$

$$buy_i(t) \equiv \text{"0"} \tag{21}$$

where $buy_i(t)$ is the limit price of the buy order and $sell_j(t)$ is the limit price of the sell order.

Paraphrased, this means that either the buy limit price needs to be greater or equal to the sell limit price and/or at least one of the orders is a market price order.

In case that the amount of Bitcoins in the buy and sell order does not match, the order with the smaller amount is fully executed. The fully executed order will be removed from the order book and the non-fully executed one will remain with the remainder of that transaction. This process will be repeated until the top orders of the two order books do not match anymore. After the last match has occurred, all expired orders will be removed.

Price Formation: the price for a matched transaction p_T is formed by the following four rules:

1. If both sides are taking a limit order:

$$p_T = \frac{buy_i(t) + sell_j(t)}{2} \tag{22}$$

2. If only $buy_i(t) \equiv \text{"0"}$:

$$p_T = max(sell_j(t), p(t)) \tag{23}$$

DOI:10.6814/NCCU201900519

3. If $buy_i(t) > 0$ and $sell_j(t) \equiv$ "0":

$$p_T = min(buy_i(t), p(t)) \tag{24}$$

4. If $buy_i(t) \equiv$ "0" and $sell_j(t) \equiv$ "0":

$$p_T = p(t) \tag{25}$$

where $p(t)$ is the current Bitcoin price. Every time p_T has been determined, it is henceforth used as the new $p(t)$.

3.3.2 The Transaction System

The second main component of the model is the transaction system. It aims to replicate the occurrence of transaction fees in the real Bitcoin network. Transactions are processed on the blockchain of Bitcoin, in contrast to all buy and sell orders which are processed on a market exchange that occurs off the blockchain.

The transaction system in this model is essentially a list of all transactions that have been broadcasted by the user agents. It is always sorted in descending order by the fees f_i. As the real Bitcoin protocol, the model has a transaction limit called $tlimit$. Only transactions which are within $tlimit$ are able to be executed in that simulation time step. In every simulation step, a user agent has a fixed probability of 20% to make a transfer. In terms of the number of transactions made by each agent, the median value of three weeks of daily transactions added to the real Bitcoin network was used and then divided by the amount of user agents at that time (the data used was from 24th April 2016 as no earlier data is available) (Blockchain.com, 2019g).

All other transactions are considered to happen over a Bitcoin exchange which therefore are not appended to the transaction system of the Bitcoin network. The amount of Bitcoins a user sends within a single transaction is determined by a normal distribution with an average of 143 US-$ and a standard deviation of 20. The values of this mean was taken from the median of the transaction volume of all transaction in the modeled time frame (BitInfoCharts, 2019b), whereas the standard deviation was picked by the author to include a small degree of stochasticity. Even though all the transaction fees are expressed in US-Dollar, the model internally uses the corresponding amount of

DOI:10.6814/NCCU201900519

Bitcoin[6]. The recipient of the transaction is randomly chosen from the pool of users.

Fee determination: A user will attach a fee to his/her transaction based on the number of transactions $L(t)$ and the following rules:

$$\text{1. If } L(t) \geq tlimit: \qquad f_i = (f_{last}(t) + \alpha) \cdot N(\mu, \sigma) \tag{26}$$

$$\text{2. If } L(t) < tlimit: \qquad f_i = (f_{last}(t) - \alpha) \cdot N(\mu, \sigma) \tag{27}$$

$$\text{3. If } L(t) = 0: \qquad f_i = (f_{last}(t-1) - \alpha) \cdot N(\mu, \sigma) \tag{28}$$

where $N(\mu, \sigma)$ is a normal distribution with the mean of 1 and standard deviation of 0.01. $f_{last}(t)$ is the fee of the last transaction of this time step still within $tlimit$, whereas $f_{last}(t-1)$ was the last fee within $tlimit$ of the last time step. α is a constant set to 0.01US-$. Using these rules, a user always tries to have his transaction picked up by miners. If the number of transactions is larger than the transaction limit he will increase his fee just by a small amount to outbid the lowest fee currently still within the transaction limit. If the number of transaction is lower than the transaction limit he will offer a fee slightly lower than currently lowest fee. The user is not offering a minimal fee, since s/he still expect other users to make a transaction. This way fees are able to lower in times of low demand. Lastly, if there are no other transactions a user has no current fee to compete with, but still attaches the lowest fee within limit from the previous day lowered slightly as s/he expects a lower, yet prevelant demand for transactions. As mentioned, all transactions have their fees attached in Bitcoin, however a transaction also holds the information of a fee expressed in US-Dollar, which is actually the information used for $f_{last}(t)$.

Even though there is no minimum fee required by Bitcoin protocol (Kaskaloglu, 2014), a user will always attach a minimum fee to prevent miners from being indifferent to process their transaction. The absolute minimum fee for a transaction is set to $1 \cdot 10^{-8}$ Bitcoin (called one satoshi). Therefore:

[6]Therefore, each user uses to the current Bitcoin price to calculate the amount s/he wants to send even if the value he initially wanted to send might change in the future due to appreciation.

DOI:10.6814/NCCU201900519

$$\text{For } \frac{f_i}{p(t)} < 10^{-8}, \qquad f_i = \frac{p(t)}{10^8}. \tag{29}$$

As with orders, transaction have an expiration date, which is set to:

$$ExpOrd_i(t) = Round(t + patOrd_i) \tag{30}$$

where t is the current simulation step, $patOrd_i$ is a log normal distribution with a mean of 1 and standard deviation of 0.01 and $Round$ is a function that always round to the nearest integer.

At the end of every time step, all transactions within $tlimit$ are executed. The recipient receives his/her designated amount. The total sum of all fees within $tlimit$ will be distributed among all miners in proportion to their hashing power. All executed and expired transactions will be removed, all other transactions remain in the list, still in sorted order for the next time step.

3.4 Initialization

3.4.1 Number of Agents and their Type Distribution

The amount of agents entering the model at any simulation time step is determined by a generic exponential function, which was taken from Cocco et al. (2017). The authors estimated the amount of participants in the Bitcoin network based on observations over time. The amount was slightly adapted to fit with the time frame used in this model, 2014.02.23 to 2018.04.03, and has the following form:

$$N(t) = 1.77 \cdot 10^4 \cdot e^{0.002465 \cdot (t+1878)} \tag{31}$$

where $N(t)$ is the total amount of agents at simulation step t.

There is little data available for the distribution of the kind of participants in the Bitcoin network. As for 2019, there were nearly 35 million Bitcoin wallet users (Statista, 2019). It is estimated that the current amount of miners in Bitcoin is around one million

DOI:10.6814/NCCU201900519

(Buybitcoinworldwide, 2019). Therefore, the probability for an agent at any time step to be a miner is set to 0.025.

Cocco & Marchesi (2016) stipulated that of all the agents which are not miners, 70% are users (in their paper generally referred to as a random trader) and 30% are chartists. Table 1 gives the fractions of agents, their average initial and average final number. Figure 7 shows average the development of the agents number per type.

Table 1: Agent Type Fraction

Agent Type	Fraction	Average Initial Amount	Average Final Amount
Miner	2.5%	9.21	358.09
User	68.25%	245.1	9802.14
Chartist	29.25%	103.69	4221.77

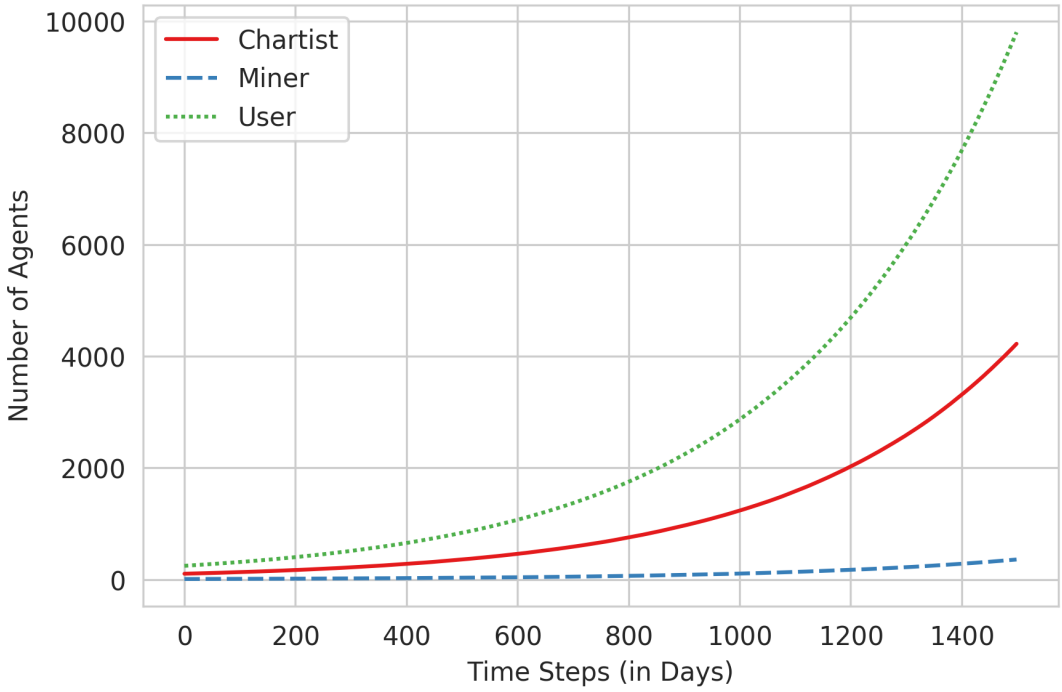

Figure 7: Number of Agents per type during Simulation

3.4.2 Agent's Wealth

The wealth distribution of the agents is following a power law distribution. This approach has also been used in Cocco & Marchesi (2016) and Raberto et al. (2003). A

DOI:10.6814/NCCU201900519

power law has the general form of:

$$z(x) = \frac{C}{x^{\delta}} \qquad \text{(Newman, 2005)} \tag{32}$$

Rewritten for the purpose of this thesis, it has the following form:

$$b_{i,0} = \frac{b_{max}}{i^{\delta}} \text{ , } i \text{ belongs to } j | t_j^E = 0 \tag{33}$$

where $b_{i,0}$ are the initial Bitcoins of initial agent i, b_{max} the initial Bitcoins of the wealthiest agent and δ is a constant for the power law. The total amount of Bitcoins for $t = 0$ is equal to:

$$b(0) = \sum_{i=0}^{N(0)} \frac{b_{max}}{i^{\delta}} = \sum_{i=0}^{N(0)} b_{i,0} \tag{34}$$

where $N(0)$ is the initial amount of agents.

Equation (33) is used several times. Once for the Bitcoins of the agents at the beginning of the simulation. Here, like in Cocco & Marchesi (2016), δ is set to 1.

b_{max} is calculated by solving equation (34) for b_{max} with $b(0)$ set to 60% of the actual amount of the day at the beginning of the simulation, since like in Cocco & Marchesi (2016), it is assumed that only 60% of the Bitcoins are in circulation. $N(0)$ was retrieved from equation (31). This gives a b_{max} of 1,156,094 Bitcoins, which is then divided by the computational factor of 5000.

The second time it was used to calculate the cash for the initial agents with the same delta and a maximum cash of 20,587. The last time the equation was used, was for the cash of all the agents which are entering at a later point. For that purpose δ was set to 0.6 and the cash of the richest trader was set 10 times higher as the cash from the richest initial trader. Their initial Bitcoins are zero. A pool of cash values was generated in advance with the size of the number of agents which will join the model later. Every time a new agent joins, a random value is pulled and removed from the pool and then assigned to that agent. This approach and the parameters used are all similar to Cocco & Marchesi (2016).

DOI:10.6814/NCCU201900519

3.5 Calibration

This thesis aims to analyze optimal design possibilities of a cryptocurrency. For that purpose, the two parameters, the transaction limit $tlimit$ and the initial Bitcoin generation $B(0)$, henceforth called $bitgen$, were picked. These two variables were chosen, because both of these are part of the original protocol of Bitcoin and were never really justified by their creator Satoshi Nakamoto. Additionally, these two variables can be adjusted comparatively easily, by a new consensus of all Bitcoin participants who run a full version of the Bitcoin protocol on their computer, implementing a hard fork. The following sections will describe the three different calibration approaches of this thesis.

3.5.1 Realistic Parameters

The first calibration process involved using the real Bitcoin parameters to have a baseline value to compare the other calibration processes against. As described in section 2.1.2, the current Bitcoin block has a size limit of 1 MB. Since an average transaction has a size of 250 bytes, this results in a block transaction size limit of 4000. As the model simulates on a daily basis, this number is multiplied by 144 (6 blocks per hour and 24 hours a day). Afterwards, it is divided by the computational factor of 5000. Therefore, $tlimit$ is set to 115 for this calibration.

The amount of Bitcoins mined per block was already described in section 2.1.3. As described, the block mining reward halves every 210,000 blocks, which equals approximately 4 years. Therefore, the mining reward does not stay the same during the whole simulation. However, for simplicity reasons, $bitgen$ is only expressed in its initial value, even though it always halves after 210,000 blocks. Table 2 shows the values used for the mining reward during the simulation runs.

Table 2: Daily Mining Reward in Simulation

Date	Real Bitcoin Reward	Simulated Bitcoin Reward
2014.02.23 - 2014.11.11	3600	0.72
2014.11.12 - 2018.04.03	1800	0.36

DOI:10.6814/NCCU201900519

3.5.2 Optimization for Economic Efficiency

The second calibration step is the optimization for economic efficiency in the model. Economic efficiency can be considered as the sum of the economic benefits minus the costs (Varian, 1996). For this thesis the economic efficiency is therefore considered as the total wealth in the model's system. This optimization objective is similar to the one from Chiu & Koeppl (2017). In their paper, Chiu & Koeppl used a theoretical model to optimize the welfare of a market which uses Bitcoin as a currency. They defined welfare as the aggregated surplus of the market minus the expenditures, i.e., the mining costs. The aim of this optimization process is now to verify their finding, which is that the optimal calibration of a blockchain to maximize the welfare is a low mining block reward and no transaction fees. This thesis uses a similar function to measure the economic efficiency, which is considered as the overall wealth and it is defined as:

$$\Pi(t) = \sum_{i=1}^{n} b_i(t) \cdot p(t) + c_i(t) + \frac{r_{tot}(t)}{R(t)} \tag{35}$$

where $b_i(t)$ are agent i's Bitcoins, $c_i(t)$ his fiat cash, $p(t)$ the Bitcoin price at time t and $\frac{r_{tot}(t)}{R(t)}$ the current worth of the miners' hardware in US-Dollar.[7]

Since the relation between the parameters and the wealth is not known and might not be linear, a brute force approach would be very computation-intensive and therefore unfeasible. Instead, a multi-objective evolutionary algorithm (MOEA) is used to maximize the wealth. This approach has been used for multi-objective optimization of agent-based model before by Narzisi et al. (2006) and is still used in this case here with only one objective due to its very efficient nature.

The multi-objective evolutionary algorithm used in this thesis is called Non-dominated Sorting Genetic Algorithm II (NSGA-II), developed by Deb et al. (2002) and will be briefly explained in the following:

Generally NSGA-II, just like any other MOEA optimizes f with m objectives and the following form:

$$\textbf{Optimize: } f(o) = (f_1(o), ...f_m(o)) \tag{36}$$

[7] As users and chartists can not purchase hardware, thus their hardware's value is equal to 0.

DOI:10.6814/NCCU201900519

Subject to: $o_l < o < o_u$

where o is a vector of optimization variables with the lower and upper limit of o_l and o_u (Seah et al., 2012). A solution o_i dominates solution o_j if the following conditions are fulfilled: 1. o_i is better or equal in all objectives as o_j. 2. o_i is better than o_j in at least one objective. Thus, a solution that is not dominated by any other solution is called non-dominated.

Just like most other evolutionary algorithm, NSGA-II starts by picking a random set of parameters (their genetic code here) and uses the outcomes for the creation of a new generation, that is a new set of parameters. The new offspring is created using mutation and crossover including a tournament selection to determine which parameters to use for the crossing. NSGA-II has the following special features: First of all, for the fitness evaluation, all results are sorted into classes of Pareto fronts, that is by how many other candidate solutions they are being dominated. For candidate solutions of the same front, a crowding distance is used which tries to diversify the offspring and prefers results that are further away from other solutions (in terms of their euclidean distance of their parameters to that of other) (Calle, 2017). Secondly, it follows an elitist principle, i.e., only the top performing 50% of the candidate solutions in terms of their fitness evaluation will be included for the crossover and mutation for the next generation (Deb et al., 2002).

To make the optimization process feasible, some compromises have to be made. First of all, the NSGA-II have been reduced to 1000 from the original 1500 simulation steps. Additionally, the number of Monte Carlo simulation runs for each NSGA-II candidate solution has been picked, that allows for a robust mean, but could decrease the overall computational time. Each candidate solution is then run for the amount of Monte Carlo runs and the outcomes passed back to the algorithm. This approach has also been used in Narzisi et al. (2006). We conducted a series of simulation runs using the model calibrated with the real $tlimit$ and $bitgen$ values. Table 3 shows the average for both the wealth and the overall hashing power, as the latter will be used in the last calibration process. Based on the results, 25 Monte Carlo runs were chosen to estimate the wealth and hashing power of each candidate solution during the NSGA-II optimization process, as the average wealth and hashing power appeared to sufficiently stabilize at that amount of runs. The upper limit for $tlimit$ was set 70 times larger than the real one and 10 times

DOI:10.6814/NCCU201900519

for $bitgen$.

Table 3: Model's Stochasticity per amount of Monte Carlo Runs

No. of Monte Carlo runs for each candidate solution	Average wealth (Std.)	Average hashing power (Std.)
1	14794	13,277,608
5	13476 (2763)	11,885,676 (1,855,772)
25	12672 (2277)	11,073,409 (1,390,767)
50	12904 (2226)	11,168,782 (1,352,951)
100	12795 (2295)	11,047,399 (1,418,213)

For the implementation of NSGA-II the python package *pymoo* by Blank & Deb (n.d.) was used. All default parameters have been used, including a binary tournament selection, a crossover rate of 0.9 and a mutation rate equal to 0.5 (1 divided by the number of parameters). All other meta-parameters have been listed in table 4.
The meta-parameters for the NSGA-II in this calibration are listed in table 4.

Table 4: Meta-Parameters for NSGA-II of Wealth Optimization

Objective	Maximize: $\Pi(t)$
$\mathbf{bitgen_l}$	0
$\mathbf{bitgen_u}$	7.2
$\mathbf{tlimit_l}$	0
$\mathbf{tlimit_u}$	8,050
No. of Monte Carlo runs for each candidate solution	25
No. of simulation steps per run	1000
Size of the population	20
No. of generations	20

3.5.3 Optimization for Economic Efficiency and Hashing Power

The last calibration step is the optimization for economic efficiency and hashing power. The optimization of a cryptocurrency just for the economic efficiency or in this case the overall wealth might be not a feasible solution, since the network's security is neglected. In the case of Bitcoin and similar cryptocurrencies, the network's security relies completely on its hashing power (Pagnotta, 2018). It is therefore a vital aspect of the design and should also given a high degree of attention. The network's hashing

DOI:10.6814/NCCU201900519

power, introduced in section 3.2.3, is defined as:

$$r_{tot}(t) = \sum_{i=1}^{j} r_i(t) \tag{37}$$

where $r_i(t)$ is the hashing power of miner i and j is the amount of all miners at t. This time the NSGA-II algorithm optimizes for two objectives, which is the maximization of $\Pi(t)$ and $r_{tot}(t)$, where wealth and NSGA-II both have been introduced in the previous section. The optimization in this step will not return one optimal value, but instead an optimal Pareto front of values, that are all not dominated by each other. This is expected to result in a function that represents the trade-off between wealth and hashing power of a cryptocurrency. For that reason, the population size of this optimization was increased in order to receive a larger number of optimized outcomes that can be used to derive a more accurate function. The final meta-parameters for the NSGA-II are the following:

Table 5: Meta-Parameters for NSGA-II of Wealth and Hashing Power Optimization

Objective	Maximize: $\Pi(t)$ Maximize: $r_{tot}(t)$
$\mathbf{bitgen_l}$	0
$\mathbf{bitgen_u}$	7.2
$\mathbf{tlimit_l}$	0
$\mathbf{tlimit_u}$	8050
No. of Monte Carlo Runs for each candidate solution	25
No. of Simulation steps per run	1000
Size of the population	40
No. of generations	20

3.6 Summary

This chapter described the model used in this thesis. It was created to mimic the real Bitcoin, its market and transaction system. Initially, it was also calibrated with the real Bitcoin parameters. Furthermore, the model was optimized in two parts using a multi-objective genetic algorithm. First only for the maximization of the total wealth and second for the maximization of both total wealth and network hashing power in a multi-objective fashion. The results of these three calibrations will be presented in the next chapter.

DOI:10.6814/NCCU201900519

4 Findings

This chapter will describe the findings of this thesis. Section 4.1 presents the simulation results of the model calibrated with the real Bitcoin parameters. It will explain in detail the findings for the different outcomes, such as price, wealth and hashing power of the Bitcoin network. Section 4.2 describes the results of the optimization for wealth of the model. It will explain the optimization results of the genetic algorithm and the outcomes of the model using the optimized parameters. Section 4.3 describes the results of the second optimization for wealth and network hashing power. It will also explain the relation between the input parameters and outcomes as well as the Pareto optimal relation between the two outcomes. Lastly in section 4.4, a summary will be made.

4.1 Real Parameters

The first calibration used the real Bitcoin parameters. That means a daily block reward of 0.72 Bitcoins and a transaction limit of 115. The following will present the results of this first calibration.

4.1.1 Price Development

Figure 8 shows the average price development of the Monte Carlo simulation. It reached an average of 10,116 US-Dollar in the last simulation step. One can observe, since new agents with new cash enter the market at every period, that the price is generally always rising. However, since this figure is only representing the average this is actually not the case in all runs of the simulation. Figure 9 shows the Bitcoin price development of a single simulation run in comparison to the development of the real Bitcoin price development.

One can observe that the simulated Bitcoin price of that run initially has a higher degree of volatility, while later on it does not match the volatility degree of the real Bitcoin price. The degree of volatility of the model is created by the chartists, which pick up small upwards and downward trends and amplify them. The average continuous upward trend of the simulation price is caused by the nature of the model. More specifically, as shown in equation (31), new agents will continue to join at each simulation step, bringing more wealth in form of cash and have no choice but to acquire Bitcoin, hence

DOI:10.6814/NCCU201900519

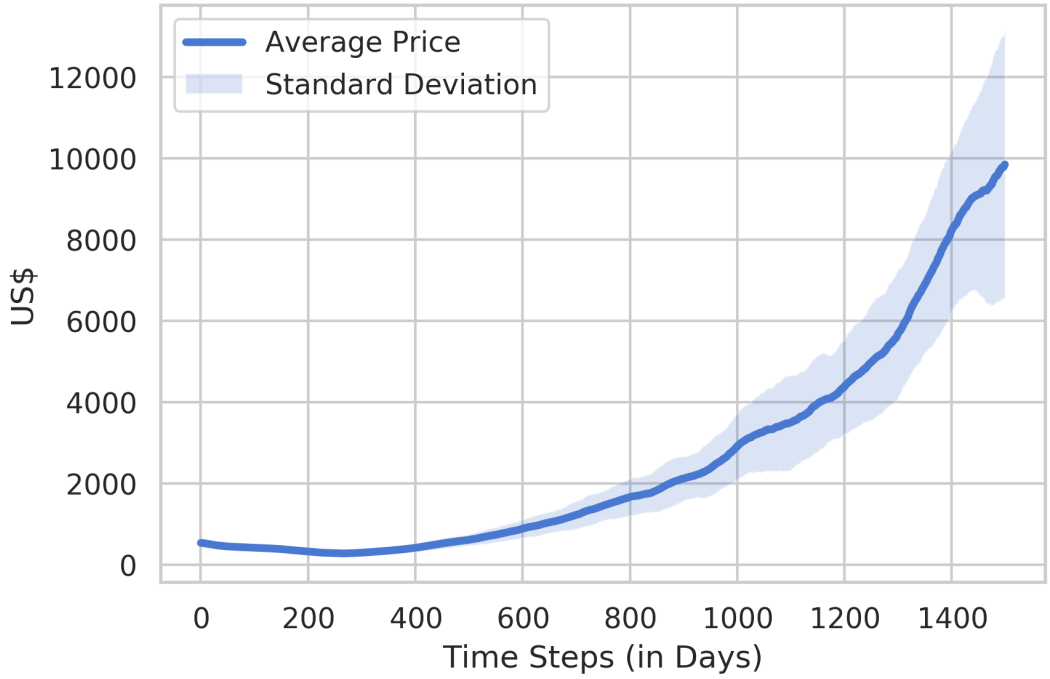

Figure 8: Simulated Average Bitcoin Price

the price always continues to increase in the long run. This clearly is a too simplistic assumption and would need to be modified in the future to reflect the reality more accurately.

4.1.2 Hashing Power Development

Figure 10 and 11 show the simulated average of the hashing power and its standard deviation. The final average hashing power level of the simulated network is $7.38 \cdot 10^{17}$ H/s, which is equal to 730,000 tera-hash per second. This number can be explained by the on average continuously rising Bitcoin price, which also increases the reward the miners receive daily. Falling hardware prices and increasing hardware efficiency over time allowed for miners to increasingly invest in hashing power.

Figure 12 shows the hashing power development of the real Bitcoin network and a single simulated run remultiplied with the computational factor of 5000 for comparison reasons. While the model is able to capture the upward trend of the real network hashing power, it outgrows it by a power degree of two, since the real hashing power of the network at the last simulated day was only $2.82 \cdot 10^{19}$ H/s, compared to the $3.68 \cdot 10^{21}$ H/s of the simulated one.

DOI:10.6814/NCCU201900519

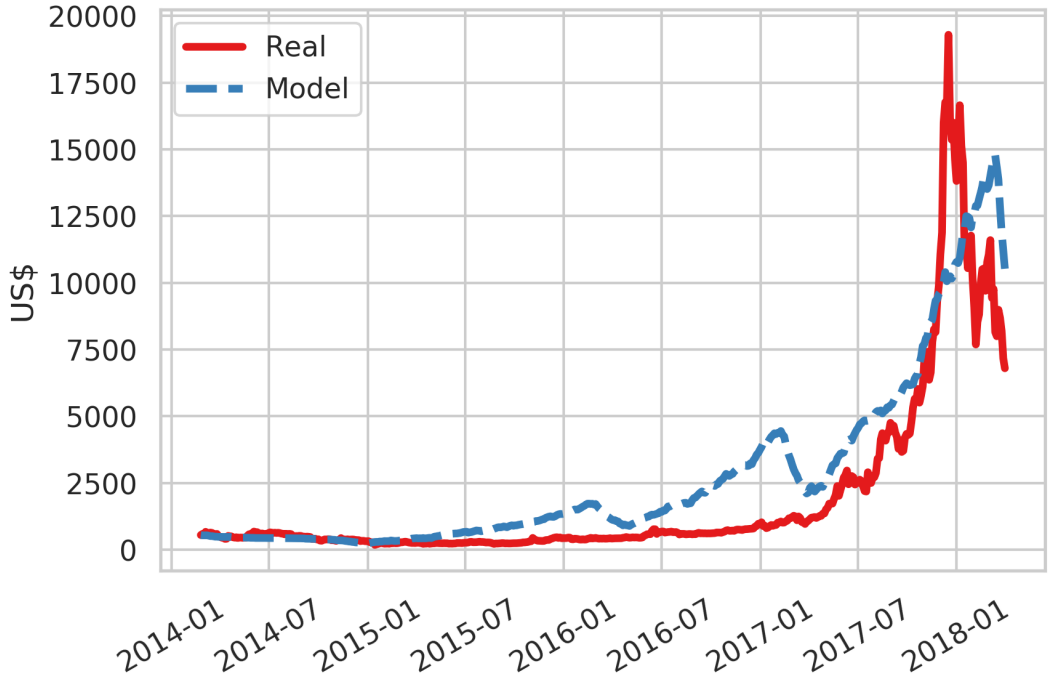

Figure 9: Real and Simulated Bitcoin Price of a single Run
Data from Blockchain.com (2019f)

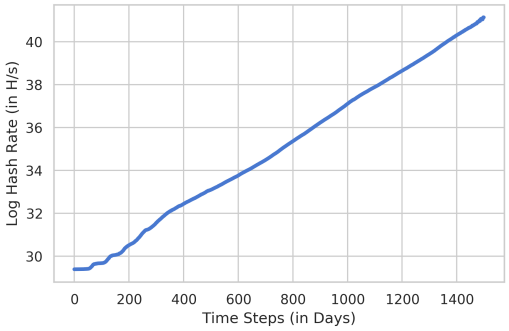

Figure 10: Simulated Average Hashing Power

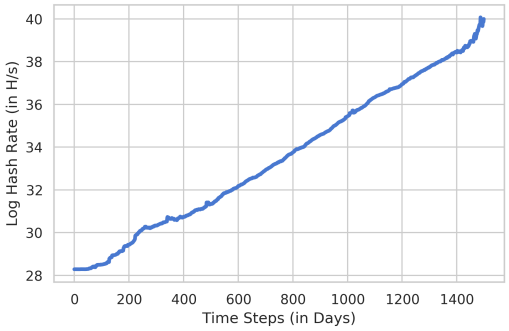

Figure 11: Standard Deviation of Simulated Hashing Power

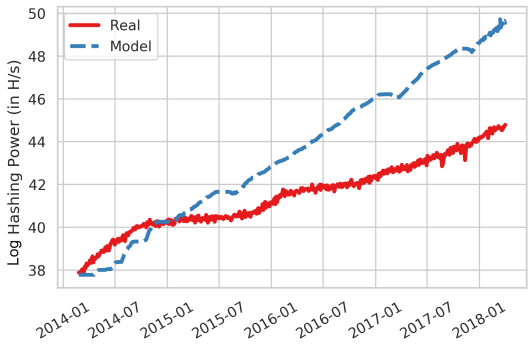

Figure 12: Real and Simulated Hashing Power of a single Run
Data from Blockchain.com (2019e)

DOI:10.6814/NCCU201900519

This difference might be explained by the models underlying exponential functions (2) and (3) for the hashing power per US-Dollar and the electricity consumption per Hash/s, which might not be realistic enough to reproduce real values. Additionally, the generated revenue plays an important role for the miners and their investment in hashing power. The simulated Bitcoin price was also higher on average than the real one, hence the simulated miners did have a higher incentive to invest more.

The average electricity consumption corresponding to the network's hashing power is shown in figure 13 and 14. It continues to increase with an exponential growth, since the decreasing electricity costs per hash per second can not outset the increasing hash per dollar and the miners continuous investment. The simulated electricity consumption reaches a value of 4.9 million kilo watt per hour.

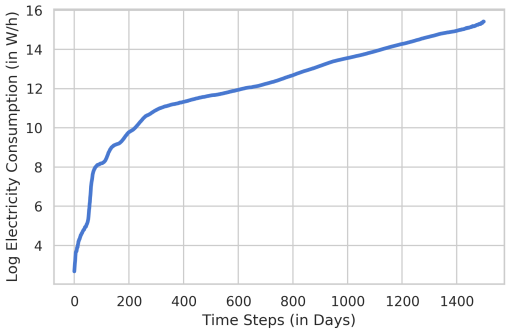

Figure 13: Simulated Average Electricity Consumption

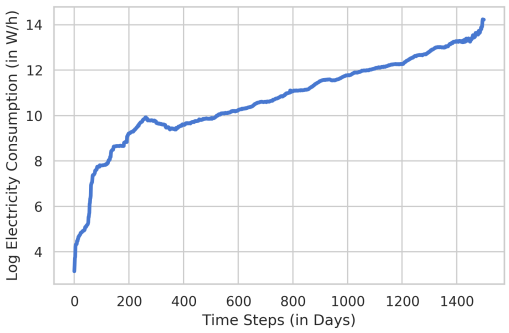

Figure 14: Simulated Electricity Consumption Standard Deviation

4.1.3 Transaction Fee Development

A major addition to the model was the transaction system. Figure 15 shows the average transaction fee with the corresponding standard deviation. Until the 620th simulation step, the transaction fee stays at under two cents US-Dollar. After that step, it continuously rises until the end, where it reaches a transaction fee of 22.54 US-Dollar. This pattern can be well explained by figure 16. It shows the average unprocessed transactions at every time step, meaning the amount of transactions that were not within $tlimit$ at the end of each simulation step. It becomes greater than zero for the first time on the 374th simulation step. Therefore, it can be observed that these two figures are highly related, as the amount of unprocessed transactions with a certain time delay influences the transaction fee.

DOI:10.6814/NCCU201900519

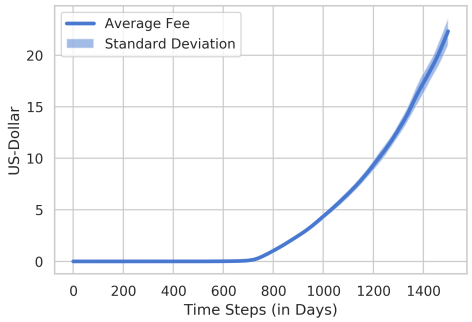

Figure 15: Simulated Average Transaction Fee

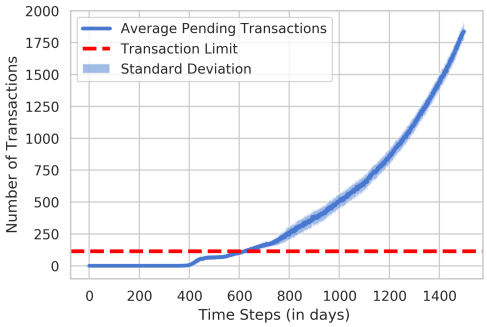

Figure 16: Unprocessed Transactions during the Simulation

Figure 17 shows the real average transaction fee during the simulated time frame. The simulated model was able to reproduce some of its patterns. One can observe that the real fees also stayed low until they suddenly begun to raise. The model, however, did not accurately reproduce the time step at which the real fees begun to raise. Also, the real fee level was not fully reached and lastly, the simulated fees continued to rise, despite what happened for real Bitcoin. This might be explained by the continuous addition of users to the model with a fixed tendency to make a transaction, therefore smoothing the growth of the transaction fees unrealistically.

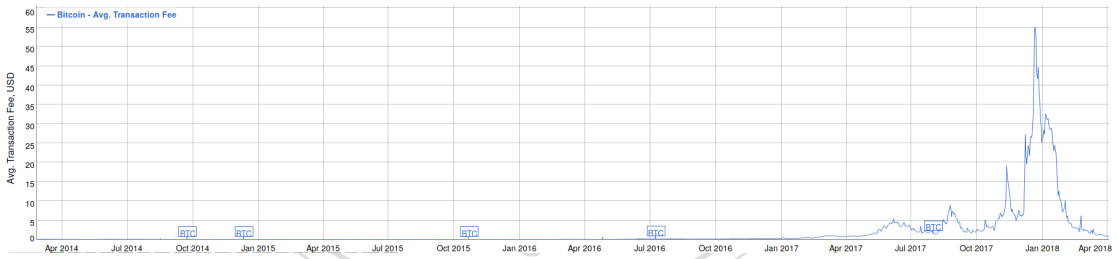

Figure 17: Real Bitcoin Transaction Fees
BitInfoCharts (2019a)

4.1.4 Wealth Development

Figure 18 and 19 show the development of wealth per agent type and the corresponding standard deviation. As described in section 3.5.2, the wealth is defined as the sum of cash and the number of held Bitcoins multiplied by the current Bitcoin price at that simulation time step. Figure 20 and 21 separate the wealth into cash and Bitcoins at any given time step.

As one can see, the wealth of chartists and users is continuously rising, while the wealth

DOI:10.6814/NCCU201900519

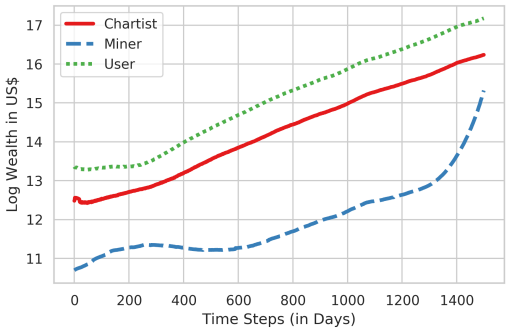

Figure 18: Simulated Average Wealth per Agent Type

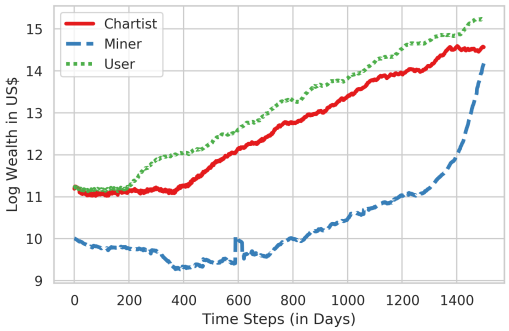

Figure 19: Simulated Wealth Standard Deviation per Agent Type

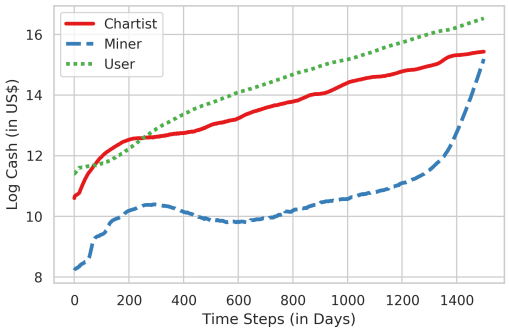

Figure 20: Simulated Average Cash per Agent Type

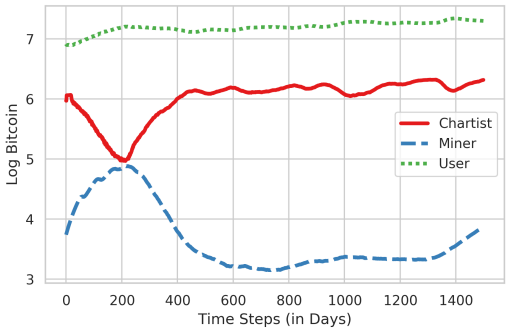

Figure 21: Simulated Average Bitcoin per Agent Type

of miners experiences initially a lower volatility until nearly the end of simulation where the volatility nearly reaches the same level as that of chartists and users. This might be explained by the fact that miners sell their Bitcoin at a lower frequency than users and chartists and never place buy orders. Also, the number of chartists and users is rising at a higher pace than that of miners. Since all new agents bring new cash, the wealth of those overall groups will always increase. The high standard deviation is likely caused by the volatile Bitcoin price, which has a great impact on the wealth.

Figure 22 and 23 show the wealth development per capita and the corresponding standard deviation (See figure 7 for the development of the agents number per type). The wealth of a single miner increases initially and then drops below the level of wealth of the chartists and users. It stays rather low until during the last 200 simulation steps, then it steeply grows and outweighs the wealth per capita of the chartists and users drastically. This also seems to be explained by the fast growth of the Bitcoin price in the later simulation steps. Interestingly enough, the wealth per capita of the chartists initially was larger than the one from the users, but at the end of the simulation it was slightly

DOI:10.6814/NCCU201900519

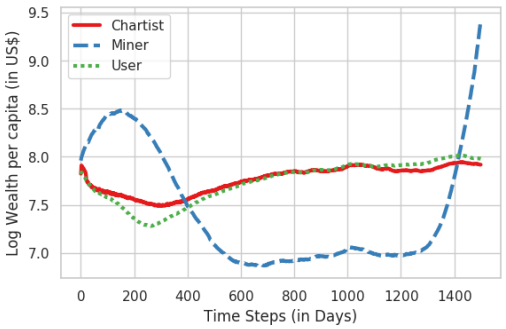

Figure 22: Simulated Average Wealth per capita

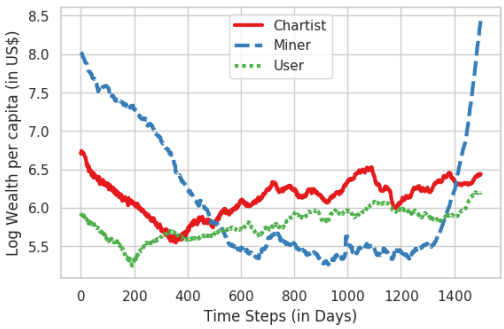

Figure 23: Simulated Wealth Standard Deviation per capita

lower.

4.2 Optimized Wealth

This section summarizes the findings of the model, which was calibrated to maximize the economic efficiency that was considered to be the overall wealth in the model. Therefore, as described in section 3.5.2, the NSGA-II was used. Figure 24 shows how the algorithm converged for the later generations. The overall wealth converges to 12.6 million US-Dollars for a simulation with 1000 simulation steps.

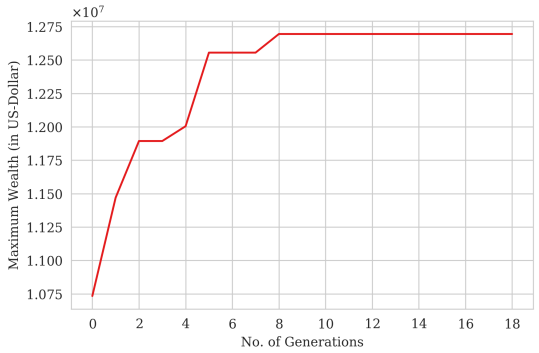

Figure 24: Convergence of the Optimization for Wealth

Figure 25 shows the selection of parameters during different generations. As one can observe, daily block reward tends to go against zero and the daily transaction limit between 2000 and higher. The final parameters for the best result of the last generation are 0.0048 Bitcoin per day and a daily transaction limit of 1414, which would be equal to a block size of 12.3 MB. These parameters would imply a decrease of the Bitcoin generation per day by 99.3 %, nearly abolishing all inflation and an increase of the block size by 1229.6 %, allowing the protocol to mitigate the whole demand for transactions.

DOI:10.6814/NCCU201900519

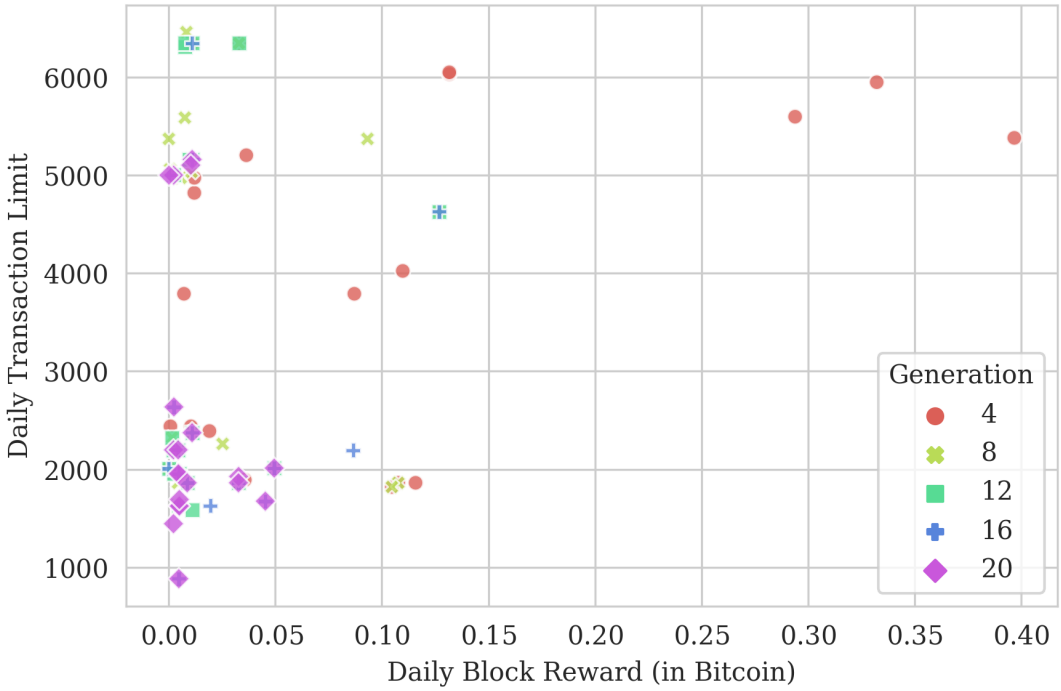

Figure 25: Parameters used by Generation for Wealth Optimization

After the optimization, the model was run in a Monte Carlo simulation with 100 runs for 1000 time steps each. Table 6 summarizes the results.

As one can see, transaction fees decreased to nearly zero and the average Bitcoin price increased to 3,832US-$. The overall wealth also increased by around 587,000 US-Dollar when optimized for wealth. However, the distribution of wealth has changed. Since the transaction fees have turned nearly zero, the users wealth increased but the wealth of the miner has decreased, since they no longer receive the fees as a mining reward at every time step. Due to the lower income of the miners, who then lost the incentive to invest in mining, the network hashing power has also decreased to $3.07 \cdot 10^{14}$ H/s.

4.3 Optimized Wealth and Hashing Power

This section summarizes the findings of the model, which was calibrated to maximize two objectives: the economic efficiency which is considered to be the overall wealth in the model and the network hashing power using the NSGA-II again.

DOI:10.6814/NCCU201900519

Table 6: Simulation Results with Wealth optimized and real parameters

Outcome / Parameters	**Wealth Optimized** *bitgen*: 0.0048 *tlimit*: 1414	**Real Parameters** *bitgen*: 0.72 *tlimit*: 115
Price ($)	3,832.32	2,796.35
Transaction Fee ($)	0.00004	4.25
Network Hashing Power (H/s)	$3.07 \cdot 10^{14}$	$1.26 \cdot 10^{16}$
Overall Wealth (M$)	11.88	11.29
Miner Wealth (M$)	0.11 (0.93%)	0.32 (2.83%)
User Wealth (M$)	8.43 (70.96%)	7.78 (68.91%)
Chartist Wealth (M$)	3.37 (28.36%)	3.20 (28.34%)

4.3.1 Parameters

Figure 26 shows all candidate solutions the genetic algorithm produced during its optimization process (800 in total). The candidate solutions depicted by a cross represent the optimal solutions, meaning that the outcome of those simulation runs using these parameters are not dominated by other outcomes. The parameter pairs resulting in the extreme outcomes are $(tlimit_1 : 2,520, B_1(t) : 0.134)$ for the highest hashing power and $(tlimit_4 : 3,871.93, B_4(t) : 9.65)$ for the highest wealth.

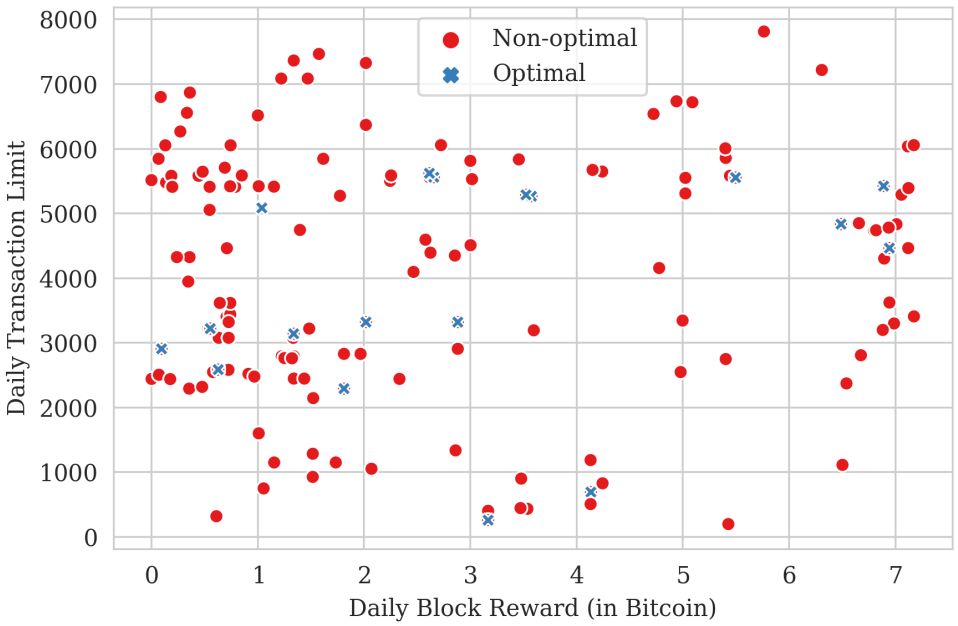

Figure 26: Parameters found by NSGA-II for Wealth & Hashing Power Optimization

800 candidate solutions from all generations found by NSGA-II (The points overlap due to parameters having the same values in different generations).

DOI:10.6814/NCCU201900519

Table 7 lists the mean and standard deviation for both parameters. Both have a relatively high standard deviation. This is caused by the nature of the NSGA-II, which tries to diversify the candidate solutions, hence choosing parameters which are less similar. The average of $tlimit$ is a lot higher than the real parameter of 115. It would imply an increase by 3372% or an increase of the block size to 33.72 MB. Nearly all $tlimit$ values are large enough to mitigate the whole demand for transactions of around 600 transactions per time step. The implication will be discussed in the next subsection. The average of $bitgen$ is also significantly larger than the real parameter of 0.72 with an increase of 396.5%.

Table 7: Mean and Standard Deviation of Optimized Parameters

	tlimit	bitgen
Mean	3,877.82	2.855
Standard Deviation	1,807.43	2.161
Min	194	0.0008
Max	7808	7.177

Figures 27, 28, 29 and 30 show each of the parameters in comparison to each of the outcome variables. It can be observed that $tlimit$ does not seem to have any obvious relationship to either of the outcome variables. However, the relationship becomes very clear for $bitgen$. There appears to be a positive relationship between $bitgen$ and the network hashing power and a negative one between $bitgen$ and the total wealth.

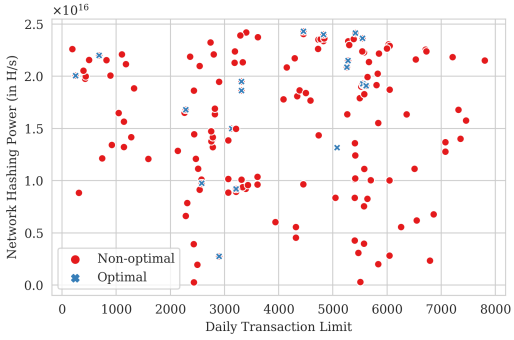

Figure 27: Daily Transaction Limit against Network Hashing Power

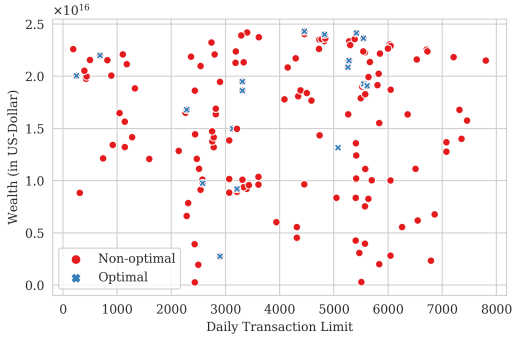

Figure 28: Daily Transaction Limit against Total Wealth

To further analyze the relationship between the optimized parameters and the corresponding outcome variables, two quadratic multiple regressions, one for each outcome variable, have been run. The first one is run against the network hashing power. Its output can be found in table 8.

DOI:10.6814/NCCU201900519

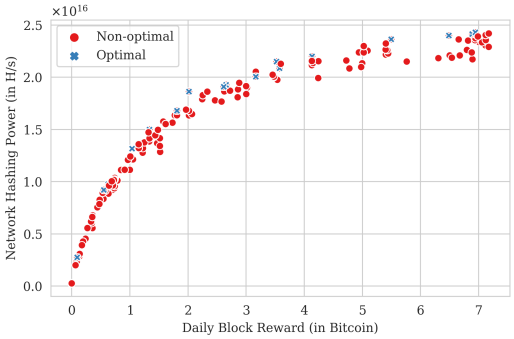

Figure 29: Daily Bitcoin Generation against Network Hashing Power

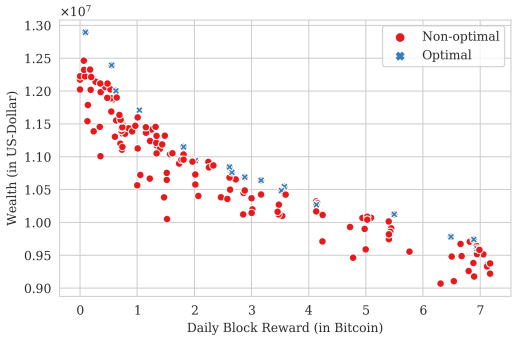

Figure 30: Daily Bitcoin Generation against Total Wealth

Table 8: Regression Results - Optimization Parameters against Hashing Power

Dep. Variable:	Hashing Power	**R-squared:**	0.962
Model:	OLS	**Adj. R-squared:**	0.961
Method:	Least Squares	**F-statistic:**	4975.
No. Observations:	800	**Prob (F-statistic):**	0.00
Df Residuals:	795	**Log-Likelihood:**	-28971.
Df Model:	4		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.999e+15	2.39e+14	12.570	0.000	2.53e+15	3.47e+15
tlimit²	-1.159e+08	1.44e+07	-8.042	0.000	-1.44e+08	-8.76e+07
tlimit	8.886e+11	1.11e+11	7.980	0.000	6.7e+11	1.11e+12
bitgen²	-6.777e+14	1.21e+13	-55.880	0.000	-7.01e+14	-6.54e+14
bitgen	7.325e+15	8.51e+13	86.113	0.000	7.16e+15	7.49e+15

The regressions has a high fit with an R-squared of 0.962. All coefficients are significant on a 99% confidence interval. Nonetheless, the effect of $tlimit$ is significantly smaller than the effect of $bitgen$. To make the difference clearer, the regression coefficients will be expressed as the partial derivation (Wooldridge, 2015).

$$\frac{\Delta \widehat{HashingPower}}{\Delta tlimit} = 8.89 \cdot 10^{11} - 2.318 \cdot 10^{8} \cdot tlimit \tag{38}$$

$$\frac{\Delta \widehat{HashingPower}}{\Delta bitgen} = 7.33 \cdot 10^{15} - 1.36 \cdot 10^{15} \cdot bitgen \tag{39}$$

Since for both $tlimit$ and $bitgen$ the value 0 was not within the observations, it can

DOI:10.6814/NCCU201900519

not be used to compare the effect. Instead, the mean value (see table 7) is used[8]. This results in a mean effect onto the network hashing power of $-9.88 \cdot 10^9$ for $tlimit$ and $3.45 \cdot 10^{15}$ for $bitgen$. The effect of $tlimit$ is outweighed by the effect of $bitgen$ by a magnitude of 6 and therefore nearly irrelevant. It can be therefore concluded, that the Bitcoin generation is mainly responsible for controlling the hashing power with a positive, yet diminishing effect as the slopes is decreasing with higher $bitgen$.

Table 9 shows the result of the second regression with the total wealth as the dependent variable. Again, the R-squared was very high with a value of 0.941. This time however, $tlimit$ is no longer significant, hence requires no farther analysis, but $bitgen$ remains significant on a 99% confidence level. The partial derivative is used again to express the effect of $bitgen$.

$$\frac{\Delta \widehat{TotalWealth}}{\Delta bitgen} = -6.81 \cdot 10^5 + 9.11 \cdot 10^4 \cdot bitgen \tag{40}$$

Using the mean value for $bitgen$, the average effect onto the total wealth is $-4.21 \cdot 10^5$. Thus, it can be concluded that as long as $tlimit$ is large enough to mitigate demand for transactions (which is on average at around 623 transaction at that time step), solely the Bitcoin generation is responsible for controlling the total wealth with a negative and decreasing effect as the slope decreases with higher $bitgen$.

Table 9: Regression Results - Optimization Parameters against Total Wealth

Dep. Variable:	Total Wealth	**R-squared:**	0.941
Model:	OLS	**Adj. R-squared:**	0.940
Method:	Least Squares	**F-statistic:**	3155.
No. Observations:	800	**Prob (F-statistic):**	0.00
Df Residuals:	795	**Log-Likelihood:**	-10932.
Df Model:	4		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1.214e+07	3.85e+04	315.573	0.000	1.21e+07	1.22e+07
tlimit²	-0.0034	0.002	-1.444	0.149	-0.008	0.001
tlimit	25.2858	17.956	1.408	0.159	-9.960	60.532
bitgen²	4.556e+04	1955.285	23.302	0.000	4.17e+04	4.94e+04
bitgen	-6.818e+05	1.37e+04	-49.710	0.000	-7.09e+05	-6.55e+05

[8]The reasoning of this methodology and its explanation is discussed by Wooldridge (2015)

DOI:10.6814/NCCU201900519

4.3.2 Outcomes

The outcomes of this optimization can be seen in figure 31. This figure represents the Pareto front. Again, a cross depicts the non-dominated solutions from the NSGA-II. As shown, there seems to be a linear relation between the wealth and the network hashing power. The extreme values for this Pareto front are a wealth of 12.68 million US-Dollars with a network hashing power of $2.52 \cdot 10^{15}$ H/s, and on the other side of the front a wealth of 9.85 million US-Dollars with a hashing power of $2.43 \cdot 10^{16}$ H/s.

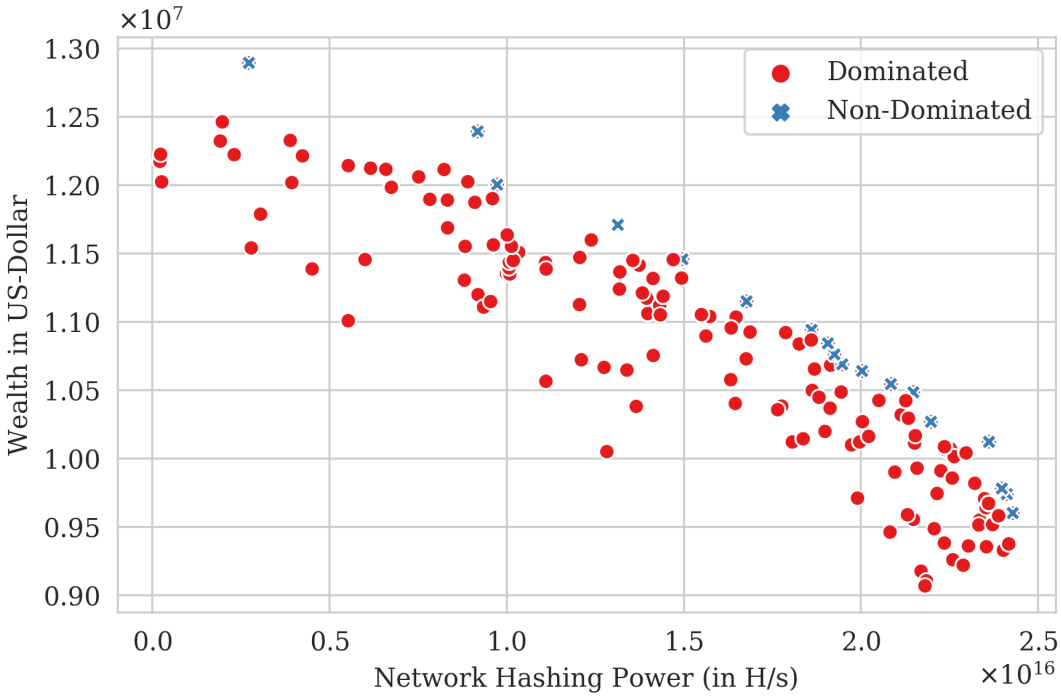

Figure 31: Pareto Front of Optimization Objectives found by NSGA-II
800 outcomes pairs (There is some overlapping, since parameters can be taken over to the next generation)

Figure 32 only shows the 18 non-dominated solutions from the results. The linear relation can be explained by the wealth distribution from the outcomes (which can be found for some of them in table 11). The higher the overall wealth, the less the wealth miners obtain. Thus, with higher overall wealth and less miners wealth, the miners invest less in hashing power, since their investment is proportional to their wealth. As a result, the hashing power is reduced. To capture this linear effect, a linear curve has been fitted, which regression results have been summarized in table 10.

The slope of this Pareto front is $-1.51 \cdot 10^{10}$, meaning that an increase of the networks hashing power by 10^{15} H/s (1 Petahash/s) would mean a decrease of 150,755

DOI:10.6814/NCCU201900519

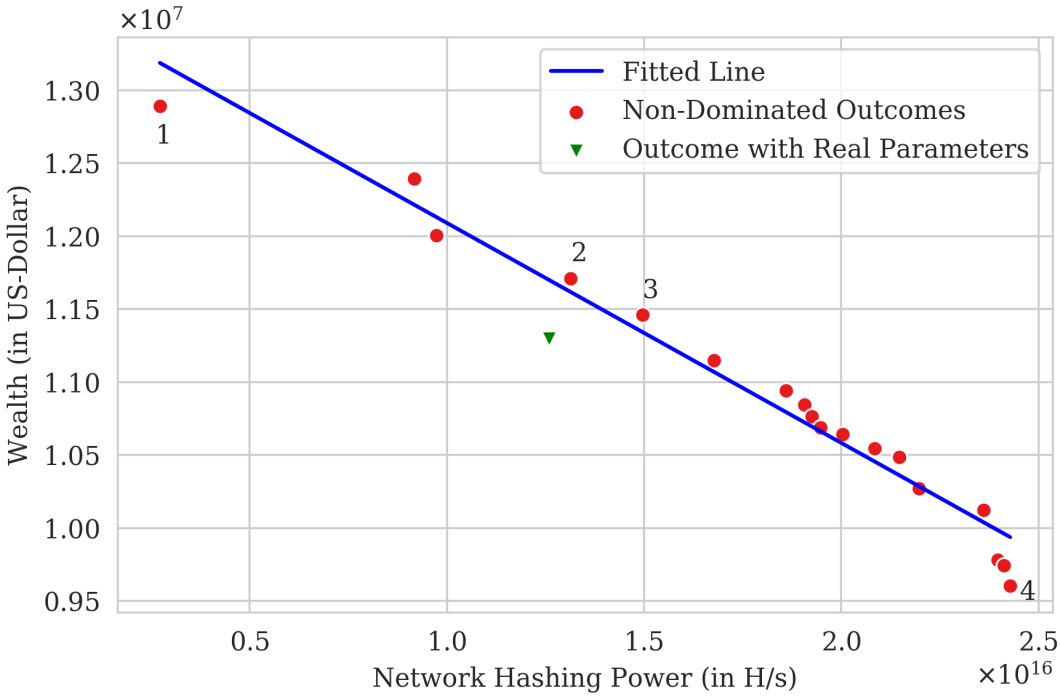

Figure 32: Non-Dominated Pareto Front of Optimization Objectives

Table 10: Regression Results - Hashing Power against Total Wealth

Dep. Variable:	Total Wealth	**R-squared:**	0.9689
Model:	OLS	**Adj. R-squared:**	0.9669
Method:	Least Squares	**F-statistic:**	497.8
No. Observations:	18	**Prob (F-statistic):**	0.000
Df Residuals:	16	**Log-Likelihood:**	-240.84
Df Model:	1		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	1.360e+07	1.275e+05	106.62	0.000	1.332e+07	1.387e+07
Hashing Power	-1.508e-10	6.757e-12	-22.31	0.000	-1.651e-10	-1.364e-10

US-Dollars of wealth. Furthermore, the model outcome with the real parameters were plotted next to the non-dominated ones as a green triangle. It can be observed in the figure, also by the numerical values in table 11 that the outcome with real parameters is dominated by all optimized outcomes by at least one objective, and it is also strictly dominated by two parameter pairs which are marked as (2)(3) in figure 32. Table 11 lists these two results (Outcome (2) and (3)) as well as the outcome for the maximal wealth and the maximal hashing power (Outcome (1) and (4)). The Bitcoin generation,

DOI:10.6814/NCCU201900519

as well as the transaction limit for the results (2) and (3), were significantly larger than the real parameters.

Table 11: Different Pareto Efficient Simulation Results compared with Real parameters

Outcome / Parameters	**(1) Maximum Wealth** $bitgen$: 0.134 $tlimit$: 2,520	**(2) Dominating Result** $bitgen$: 1.44 $tlimit$: 4,411	**(3) Dominating Result** $bitgen$: 1.861 $tlimit$: 2,722	**(4) Maximum Hashing Power** $bitgen$: 9.64 $tlimit$: 3,872	**Real Parameters** $bitgen$: 0.72 $tlimit$: 115
Price ($)	4,258.72	2,817.09	2,463,20	846.95	2,796.35
Transaction Fee ($)	0.00004	0.00003	0.00002	0.000008	4.25
Network Hashing Power (H/s)	$2.52 \cdot 10^{15}$	$1.32 \cdot 10^{16}$	$1.46 \cdot 10^{16}$	$2.43 \cdot 10^{16}$	$1.26 \cdot 10^{16}$
Overall Wealth (M $)	12.68	11.78	11.40	9.85	11.29
Miner Wealth (M $)	0.06 (0.47%)	0.21 (1.78%))	0.23 (2.02%)	0.39 (3.95%)	0.32 (2.83%)
User Wealth (M $)	8.91 (70.27%)	8.14 (69.10%)	7.99 (70.09%)	6.48 (65.79%)	7.78 (68.91%)
Chartist Wealth (M $)	3.71 (29.26%)	3.43 (29.12%)	3.20 (28.07%)	2.99 (30.36%)	3.20 (28.34%)

Once more, it can be observed that no matter what optimization criterion is chosen, the transaction fees always stay close to zero, meaning the $tlimit$ was always chosen by the genetic algorithm, to be large enough to mitigate the transaction demand. The main difference for all four outcomes is the Bitcoin generation which is the largest for the maximum hashing power with 9.64 Bitcoins and decreases with the hashing power until it is the lowest with 0.134 Bitcoins for the maximum wealth outcome. The higher the Bitcoin generation, the higher the inflation. Hence, the Bitcoin price does decrease as higher the Bitcoin generation is. Again, the wealth distribution has also changed. The wealth distribution for the maximum of wealth was 0.47 %, 70.27 % and 29.26 % for miners, users and chartists respectively. For the maximum of the hashing power, the wealth distribution now becomes 3.95 %, 65.79 % and 30.36 % for miners, users and chartists respectively. This means that with higher hashing power, the wealth of users decreases, whereas chartists and especially miners gain wealth. The decreasing worth of Bitcoin seems to be outweighed by the higher Bitcoin generation. Therefore, miners' wealth increased with the higher Bitcoin generation. This miners' higher wealth is also the reason that the network hashing power is higher than the hashing power of the model with the parameters for maximum wealth.

DOI:10.6814/NCCU201900519

4.4 Summary

This chapter described the findings of this thesis. Firstly, the results of the calibration with the real Bitcoin parameters were described. The model was able to reproduce a trend that is similar to the development of the real Bitcoin during part of the simulated time frame, even though it failed to capture any long time downward trends. Secondly, the results from the optimization for total wealth were explained. The optimal parameter combination found by the genetic algorithm corresponds to a large block size (larger transaction limit) and a lower Bitcoin generation than that used in the real Bitcoin network. This calibration has also been suggested by previous research such as by Chiu & Koeppl (2017) to improve current Bitcoin network. Lastly, for the two objective optimization the Pareto-front of optimal parameters and its corresponding outcomes have been analyzed. It was found that when the block size is chosen large enough to mitigate the whole demand for transactions, only the Bitcoin generation controls the desired outcome levels for total wealth and hashing power. Also, a linear trade-off between hashing power and total wealth could be demonstrated. The reason for this relation lies in the income distribution between the different kind of agents. With lower wealth for miner decreases the hashing power, but the overall wealth becomes larger.

DOI:10.6814/NCCU201900519

5 Conclusion

5.1 Review of Findings

This thesis' aim was to shed some light onto the design of a cryptocurrency with the example of Bitcoin, since many parameters used in today's cryptocurrencies were chosen arbitrarily or only picked with one specific objective to control for. As to the author's knowledge, this thesis is the first writing that analyzed the calibration of the two main parameters block size and Bitcoin generation for economic efficiency and security with an agent-based computational approach. It also made the contribution to the cryptocurrency field by simulating Bitcoin with a functioning market and mining process in addition to a transaction system. In the first step, to control whether the model has a realistic output, the real parameters for the block size and the Bitcoin supply were used. The results of this Monte Carlo simulation were aligned with previous research and could reproduce most of the variables realistically. In the second step, the model was optimized for the maximum economic efficiency, which was considered to be the wealth, using a genetic algorithm. The model was able to confirm other literature stating that in order to maximize the wealth, a low amount of inflation and no transaction fees should be implemented (Chiu & Koeppl, 2017). It was observed that when optimizing for wealth, the distribution changes in favor of the users and in disfavor of miners. Realistically, just optimizing for wealth might not be sufficient, as the network security, i.e., the network's hashing power, plays a significant role in Bitcoin usability, since users will not use a cryptocurrency if the security risk is too high. Thus, the model was optimized for wealth and hashing power in the last step, again using a genetic algorithm. The result was a Pareto front of optimal parameter combinations. This result indicates that there is a trade-off between network security and economic efficiency in cryptocurrencies, which can not be avoided. The higher the overall wealth (the higher the economic efficiency), the less wealth miners have, hence caused them to spend less on hashing power.

5.2 Application of Findings

The findings show, that no matter what preferences the designer of Bitcoin had in mind, the current real parameters used have much room for improvement. Since the current

DOI:10.6814/NCCU201900519

implementation of Bitcoin has a decreasing Bitcoin reward per block, it will solely rely on transaction fees in the future (Bitcoin Wiki, 2019b). The findings of this thesis contrarily show that the overall transaction limit (i.e., the block size) should be increased to a degree that it can mitigate the complete transaction demand and thus reduce the transaction fees drastically. Meanwhile, as found out by this thesis, the Bitcoin generation is solely important for controlling the preferred Pareto optimal combination of wealth and security.

5.3 Limitations

While the goal of this thesis was to use the current implementation of the Bitcoin protocol and try to optimize it by only changing its parameters, this approach has its limitations. First of all, the fixed model parameters were chosen with the assumption that there is an exponential growth of the number of agents participating in the model, which is the main reason why the model can not capture any long term downward trends. Furthermore, the model's accuracy is not always high enough to capture all variables' development realistically. On a broader note, the usage of a Proof-of-Work system is already challenged since a higher network security always comes with a higher energy consumption, which in times of an imminent climate crisis might not be the most supported choice. Furthermore, if the demand for Bitcoin transactions would further increase in the future, thus creating further congestion, the suggested adjustment to the Bitcoin parameters would not be enough and a constant increasing of the block size would be required and might not technically be possible. To solve these kind of problems in the long run, a more thoroughly redesign of the Bitcoin protocol would be required. Solutions to solve these problems are already being discussed. These include a totally different trustless proof system, called *Proof-of-Stake*, which would abandon the mining process and instead rely on the staking of funds for the verification process (King & Nadal, 2012). The problem of the congestion could potentially also be solved by a redesign of the blockchain. Some suggestions for example include a design of a blockchain, that dynamically adjust its block size, depending on the current network congestion (Huberman et al., 2019).

DOI:10.6814/NCCU201900519

5.4 Future Work

This thesis was able to give insight into the design of a cryptocurrency and provided some vital information for the calibration of a cryptocurrency like Bitcoin, without the need for complete structural changes. While it seems inevitable that one day the Bitcoin blockchain needs a whole redesign, the solutions provided in this thesis could help in the medium run to mitigate some problems cryptocurrencies are currently facing. Furthermore, a model has been created that is able to reproduce realistic results. However, its accuracy needs to be further improved and extended to support future research. More data would need to be aggregated to better calibrate some parameters. Especially the underlying exponential growth functions does not seem to reflect the reality accurately. Future work could include the development of an improved model that allows agents to decide whether to participate or not. More specifically, scholars from game theory in cryptocurrency mining games could help to improve the decision process of miners, giving miners further reasoning for their actions. If a higher level of accuracy can be achieved, further investigations could and should be conducted using this thesis' model, to analyze the effect of more drastic changes mentioned in the previous chapter.

DOI:10.6814/NCCU201900519

Bibliography

Altman, E., Reiffers, F. A., Menasché, D. S., Matar, M., Dhamal, S., & Touati, C. (2018). Mining competition in a multi-cryptocurrency ecosystem at the network edge: a congestion game approach. *1st Symposium on Cryptocurrency Analysis (SOCCA 2018)*.

Arifovic, J. (2002). Exchange rate volatility in the artificial foreign exchange market. In *Evolutionary computation in economics and finance* (pp. 123–134). Springer.

Athey, S., Parashkevov, I., Sarukkai, V., & Xia, J. (2016). Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. *Stanford University Graduate School of Business Research Paper*, *16*(42), 70. Retrieved from `https://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=2826674`

Back, A. (2002). Hashcash - A Denial of Service Counter-Measure. *Technical Report*(August), 1–10.

Bitcoin Wiki. (2019a). *Block chain.* Retrieved from `https://en.bitcoin.it/wiki/Block_chain`

Bitcoin Wiki. (2019b). *Controlled supply.* Retrieved from `https://en.bitcoin.it/wiki/Controlled_supply`

Bitcoinfees.info. (2019). *Bitcoin transaction fees.* Retrieved 2019.05.25, from `https://bitcoinfees.info/`

BitInfoCharts. (2019a). *Bitcoin avg. transaction fee historical chart.* Retrieved 2019.06.04, from `https://bitinfocharts.com/comparison/bitcoin-transactionfees.html`

BitInfoCharts. (2019b). *Bitcoin median transaction value historical chart.* Retrieved 2019.04.23, from `https://bitinfocharts.com/comparison/mediantransactionvalue-btc-sma90.html`

Blank, J., & Deb, K. (n.d.). *pymoo - Multi-objective Optimization in Python.* https://pymoo.org.

DOI:10.6814/NCCU201900519

Blockchain.com. (2019a). *Bitcoin difficulty.* Retrieved 2019.05.26, from https://www.blockchain.com/charts/difficulty?timespan=3years

Blockchain.com. (2019b). *Blockchain size.* Retrieved 2019.05.25, from https://www.blockchain.com/charts/blocks-size

Blockchain.com. (2019c). *Hash rate.* Retrieved 2019.05.26, from https://www.blockchain.com/charts/hash-rate?timespan=3years

Blockchain.com. (2019d). *Hashrate distribution.* Retrieved 2019.05.26, from https://www.blockchain.com/pools

Blockchain.com. (2019e). *Hash rate; the estimated number of tera hashes per second (trillions of hashes per second) the bitcoin network is performing.* Retrieved 2019.04.13, from https://www.blockchain.com/charts/hash-rate?timespan=all

Blockchain.com. (2019f). *Market price (usd).* Retrieved 2019.05.29, from https://www.blockchain.com/charts/market-price?timespan=all

Blockchain.com. (2019g). *Mempool size.* Retrieved 2019.05.29, from https://www.blockchain.com/de/charts/mempool-size?timespan=all#

Blockchain.com. (2019h). *Miner's revenue.* Retrieved 2019.05.25, from https://www.blockchain.com/charts/miners-revenue?timespan=2years

Budish, E. (2018). *The economic limits of bitcoin and the blockchain* (Tech. Rep.). National Bureau of Economic Research.

Buterin, V. (2016). On inflation, transaction fees and cryptocurrency monetary policy. *Ethereum Blog*. Retrieved 2016-07-26, from https://blog.ethereum.org/2016/07/27/inflation-transaction-fees-cryptocurrency-monetary-policy/

Buybitcoinworldwide. (2019). *How many bitcoins are there?* Retrieved 2019.04.25, from https://www.buybitcoinworldwide.com/how-many-bitcoins-are-there/#

DOI:10.6814/NCCU201900519

Calle, P. (2017). NSGA-II explained. *Analytics lab of University of Oklahoma*. Retrieved 2017-10-24, from http://oklahomaanalytics.com/data-science-techniques/nsga-ii-explained/

Carlsten, M., Kalodner, H., Weinberg, S. M., & Narayanan, A. (2016). On the instability of bitcoin without the block reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 154–167.

Catalini, C., & Gans, J. S. (2016). *Some simple economics of the blockchain* (Tech. Rep.). National Bureau of Economic Research.

CEIC. (2019). *China electricity price.* Retrieved 2019.04.13, from https://www.ceicdata.com/en/china/electricity-price

Chen, S.-H., & Chie, B.-T. (2008). Lottery markets design, micro-structure, and macro-behavior: An ace approach. *Journal of Economic Behavior & Organization*, *67*(2), 463–480.

Chiu, J., & Koeppl, T. V. (2017). The economics of cryptocurrencies–bitcoin and beyond. *Available at SSRN 3048124*.

CIA. (2016). *Electricity consumption - country comparision.* Retrieved 2019.05.26, from https://www.cia.gov/library/publications/the-world-factbook/rankorder/2233rank.html

Cocco, L., Concas, G., & Marchesi, M. (2017). Using an artificial financial market for studying a cryptocurrency market. *Journal of Economic Interaction and Coordination*, *12*(2), 345–365. doi: 10.1007/s11403-015-0168-2

Cocco, L., & Marchesi, M. (2016). Modeling and simulation of the economics of mining in the Bitcoin market. *PLoS ONE*, *11*(10), 1–42. doi: 10.1371/journal.pone.0164603

Cocco, L., Tonelli, R., & Marchesi, M. (2019). An agent based model to analyze the bitcoin mining activity and a comparison with the gold mining industry. *Future Internet*, *11*(1), 8.

DOI:10.6814/NCCU201900519

Cong, L. W., He, Z., & Li, J. (2018). Decentralized Mining in Centralized Pools. *Ssrn*. doi: 10.2139/ssrn.3143724

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., ... others (2016). On scaling decentralized blockchains. *International Conference on Financial Cryptography and Data Security*, 106–125.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Decker, C., & Wattenhofer, R. (2013). Information propagation in the Bitcoin network. *13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013 - Proceedings*. doi: 10.1109/P2P.2013.6688704

Digiconomist.net. (2019). *Bitcoin energy consumption index.* Retrieved 2019.05.26, from `https://digiconomist.net/bitcoin-energy-consumption`

Di Salvo, M. (2019). Why are venezuelans seeking refuge in crypto-currencies? BBC News. Retrieved 2019-03-19, from `https://www.bbc.com/news/business-47553048`

Easley, D., O'Hara, M., & Basu, S. (2019). From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*.

Gode, D. K., & Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, *101*(1), 119–137.

Goldwasser, S., & Bellare, M. (1996). Lecture notes on cryptography. *Summer course 'Cryptography and computer' security at MIT*.

Goren, G., & Spiegelman, A. (2019). Mind the mining. *arXiv preprint arXiv:1902.03899*.

Haber, S., & Stornetta, W. S. (1991). How to Time-Stamp a Digital Document. *Journal of Cryptology*, *3*(2), 99–111. Retrieved from `https://www.anf.es/pdf/Haber{_}Stornetta.pdf`

DOI:10.6814/NCCU201900519

Hoofnagle, C. J., Urban, J. M., & Li, S. (2012). Mobile payments: Consumer benefits & new privacy concerns. *Available at SSRN 2045580*.

Huberman, G., Leshno, J., & Moallemi, C. C. (2019). An economic analysis of the bitcoin payment system. *Columbia Business School Research Paper*(17-92).

Kaskaloglu, K. (2014). Near zero bitcoin transaction fees cannot last forever. *Proceedings of the International Conference on Digital Security and Forensics (DigitalSec2014)*.

King, S., & Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, *19*.

Kroll, J. A., Davey, I. C., & Felten, E. W. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries. In (Vol. 2013, p. 11).

Lee, K., Ulkuatam, S., Beling, P., & Scherer, W. (2018). Generating synthetic bitcoin transactions and predicting market price movement via inverse reinforcement learning and agent-based modeling. *Jasss*, *21*(3). doi: 10.18564/jasss.3733

Luther, W. J. (2016). Cryptocurrencies, Network Effects, and Switching Costs. *Contemporary Economic Policy*, *34*(3), 553–571. doi: 10.1111/coep.12151

Ma, J., Gans, J. S., & Tourky, R. (2018). *Market structure in bitcoin mining* (Tech. Rep.). National Bureau of Economic Research.

Marks, R. (2006). Market design using agent-based models. *Handbook of computational economics*, *2*, 1339–1380.

Montgomery, M. (2015). *Why you can't cheat at bitcoin.* IEEE Specturm. Retrieved from `https://spectrum.ieee.org/computing/networks/the-future-of-the-web-looks-a-lot-like-bitcoin`

Narzisi, G., Mysore, V., & Mishra, B. (2006). Multi-objective evolutionary optimization of agent-based models: an application to emergency response planning. *International Conference on Computational Intelligence*(Ci), 224–230.

Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, *46*(5), 323–351. doi: 10.1080/00107510500052444

DOI:10.6814/NCCU201900519

Pagnotta, E. (2018). Bitcoin as decentralized money: Prices, mining rewards, and network security. *Mining Rewards, and Network Security (October 26, 2018)*.

Pappalardo, G., Di Matteo, T., Caldarelli, G., & Aste, T. (2018). Blockchain inefficiency in the bitcoin peers network. *EPJ Data Science*, *7*(1), 30.

Raberto, M., Cincotti, S., Dose, C., Focardi, S. M., & Marchesi, M. (2005). Price formation in an artificial market: Limit order book versus matching of supply and demand. *Lecture Notes in Economics and Mathematical Systems*, *550*, 305–315. doi: 10.1007/3-540-27296-8_20

Raberto, M., Cincotti, S., Focardi, S. M., & Marchesi, M. (2001). Agent-based simulation of a financial market. *Physica A: Statistical Mechanics and its Applications*, *299*(1-2), 319–327. doi: 10.1016/S0378-4371(01)00312-0

Raberto, M., Cincotti, S., Focardi, S. M., & Marchesi, M. (2003). Traders' Long-Run Wealth in an Artificial Financial Market. *Computational Economics*, *22*(2-3), 255–272. doi: 10.1023/A:1026146100090

Rompel, J. (1990). One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the twenty-second annual acm symposium on theory of computing* (pp. 387–394).

Sapirshtein, A., Sompolinsky, Y., & Zohar, A. (2016). Optimal selfish mining strategies in bitcoin. *International Conference on Financial Cryptography and Data Security*, 515–532.

Satoshi Nakamoto. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. , 1–9. doi: 10.1007/s10838-008-9062-0

Seah, C. W., Ong, Y. S., Tsang, I. W., & Jiang, S. (2012). Pareto rank learning in multi-objective evolutionary algorithms. *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, 10–15. doi: 10.1109/CEC.2012.6252865

Sholtz, P. (2001). Transaction costs and the social cost of online privacy. *First Monday*, *6*(5).

DOI:10.6814/NCCU201900519

Statista. (2019). *Number of blockchain wallet users worldwide from 1st quarter 2016 to 1st quarter 2019.* Retrieved 2019.04.25, from https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/

Terna, P., Maggiora, M., & Battistoni, L. (2016). *Emerging cryptocurrency trust in an agent-based model* (Doctoral dissertation). Universita di Torino.

Varian, H. R. (1996). Differential pricing and efficiency. *First monday*, *1*(2).

WikiMedia. (2013). *Bitcoin block data.* Retrieved from https://de.wikipedia.org/wiki/Datei:Bitcoin_Block_Data.png

Williamson, O. E. (1975). Markets and hierarchies. *New York*, *2630*.

Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*. Nelson Education.

Yermack, D. (2017). Corporate governance and blockchains. *Review of Finance*, *21*(1), 7–31. doi: 10.1093/rof/rfw074

Zhou, Q., Zhang, Q., & Zhang, Q. (2017). Agent-based simulation research on bitcoin price fluctuation. *DEStech Transactions on Computer Science and Engineering*(aiea).

DOI:10.6814/NCCU201900519

A Table of Variables

Variable Name	**Notation**	**Description**
Simulation Time Step	t	Simulation time step of the model, where on step is equal to one day.
Time Window for Chartist	T	Time window for chartist they consider to make their order decision.
Hash Rate per Dollar	$R(t)$	Hash rate per Dollar for each simulation step with $\frac{H}{s \cdot \$}$.
Hash	H	A value returned by a hashing function.
Second	s	A time second.
Electricity Consumption	$\xi(t)$	The electricity consumption for one hour in Watts per hash and second with $\frac{W}{H/s}$.
Watts	W	Physical unit for power.
Hardware Hashing Power	$r_{i,u}(t)$	The hashing power of hardware u of miner i bought at time t .
Hardware Electricity Consumption	$e_{i,u}(t)$	The power consumption of hardware u of miner i bought at t.
Entry Time of Agent	t_i^E	Entry time of agent i.
Electricity Price	ϵ	The electricity price per W/h in US-Dollar.
Fiat Cash Fraction	$\gamma_{1,i}(t)$	The fraction of the fiat cash a miner uses for buying new hardware.
Bitcoin Fraction	$\gamma_i(t)$	The fraction of Bitcoins a miner sells to buy new hardware.
Agent's Fiat Cash	c_i	Fiat cash held by agent i.

DOI:10.6814/NCCU201900519

Agent's Bitcoins	$b_i(t)$	Bitcoins held by agent i.
Bitcoin Price	$p(t)$	Bitcoin price at time step t.
Transaction Limit	$tlimit$	The daily transaction limit of the model's blockchain.
Number of Transactions	$L(t)$	Number of transactions before the miners process them.
Bitcoin Generated per step	$B(t)$	The number of Bitcoins generated by the Bitcoin protocol every simulation step.
Total Bitcoin Income	$b_{Tot}(t)$	The total Bitcoin income for the whole network, including newly generated Bitcoins as well as transaction fees.
Networks total Hashing Power	$r_{Tot}(t)$	The networks total hashing power at step t.
Buy Order Amount	$buy_i(t)$	The amount of a buy order from agent i at step t.
Sell Order Amount	$sell_i(t)$	The amount of a sell order from agent i at step t
Order Fraction	β	The fraction of either Bitcoin or fiat cash used for an order depending on the type of it.
Buy Order Limit	$buylim_i(t)$	The limit for the buy order of agent i at step t.
Sell Order Limit	$selllim_i(t)$	The limit for the sell order of agent i at step t.
Gaussian Distribution	$N_i(1.05, \sigma_i)$	A Gaussian distribution with a mean of 1.05 and a standard deviation of σ_i.
Standard Deviation	σ_i	The standard deviation used for the Gaussian distribution. It is defined as $\sigma_i = K\sigma(\varphi)$.
Standard Deviation Constant	K	Constant used for standard deviation which is set to 2.5..

DOI:10.6814/NCCU201900519

Previous Standard Deviation	$\sigma(\varphi)$	The standard deviation for the last time steps with a time window of φ which is set to 20.
Order Expiration	Exp_i	The expiration date of an order issued by agent i at step t.
Rounding Function	$Round$	Rounds a value to the nearest integer.
Patience	pat_i	The patience for orders of agent i.
Transaction Price	P_T	Bitcoin Price used for a market transaction.
Transaction Fee	f_i	Transaction fee attached by agent i in Bitcoin.
Last Transaction Fee	$f_{last}(t)$	The fee of the transaction with the lower fee still within the transaction limit at simulation step t.
Expiration of Transaction	$ExpOrd_i(t)$	The expiration of a transaction issued by agent i at step t
Patience for Transaction	$patOrd_i$	The Patience of agent i for a transaction.
Total Number of Agents	$N(t)$	The total number of agents at step t.
Agent's Initial Bitcoins	$b_{i,0}$	The initial amount of Bitcoins of agent i.
Wealthiest Agent's Bitcoins	b_{max}	The amount of Bitcoins of the initial wealthiest agent.
Power Law Constant	δ	A constant used for the power law for the initial wealth distribution.
Parameter for Bitcoin Generation	$bitgen$	Similar to $B(t)$ gives $bitgen$ the amount of Bitcoins generated per day. For simplicity reasons only the initial value is listed, even though it follows the same function as $B(t)$.
Wealth	$\Pi(t)$	The total wealth in the model at step t.

DOI:10.6814/NCCU201900519