Pontifícia Universidade Católica
do Rio de Janeiro

Eduardo Mauro Baptista Bolonhez

A Nucleolus-Based Quota Allocation Model for the Bitcoin-Refunded Blockchain Network

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção.

Advisor : Prof. Bruno Fanzeres dos Santos
Co-advisor: Prof. Thuener Armando da Silva

Rio de Janeiro
May 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Eduardo Mauro Baptista Bolonhez

A Nucleolus-Based Quota Allocation Model for the Bitcoin-Refunded Blockchain Network

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção. Approved by the Examination Committee:

Prof. Bruno Fanzeres dos Santos
Advisor
Departamento de Engenharia Industrial – PUC-Rio

Prof. Thuener Armando da Silva
Co-advisor
Departamento de Engenharia Industrial – PUC-Rio

Prof. Luiz Eduardo Teixeira Brandão
Departamento de Engenharia Industrial – PUC-Rio

Prof. Frances Fischberg Blank
Departamento de Engenharia Industrial – PUC-Rio

Rio de Janeiro, May 25th, 2020

Eduardo Mauro Baptista Bolonhez

Graduated in Chemical Engineering and MSc. in Mechanical Engineering (hybrid renewable energy systens) by Pontifícia Universidade Católica do Rio de Janeiro. Intern at Petrobrás and Shell, trainee at Spirax Sarco, Financial Anaslyst at OI SA and currently R&D Engineer at TAESA.

Bibliographic data

Bolonhez, Eduardo Mauro Baptista

A Nucleolus-Based Quota Allocation Model for the Bitcoin-Refunded Blockchain Network / Eduardo Mauro Baptista Bolonhez; advisor: Bruno Fanzeres dos Santos; co-advisor: Thuener Armando da Silva. – 2020.

v., 78 f. : il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2020.

Inclui bibliografia

1. Engenharia de Produção – Teses. 2. Rede Blockchain;. 3. Bitcoin;. 4. Teoria dos Jogos Cooperativos;. 5. Programação estocástica;. 6. Conditional Value-at-Risk.. I. Santos, Bruno Fanzeres dos. II. Silva, Thuener Armando da. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. IV. Título.

CDD: 658.5

Acknowledgments

I would like to thank my family, especially my parents and my fiancee, as well as my advisors and all the staff at the Industrial Department of Puc-Rio. I would also like to thank PUC-Rio for the scholarship.
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Bolonhez, Eduardo Mauro Baptista; Santos, Bruno Fanzeres dos (Advisor); Silva, Thuener Armando da (Co-Advisor). **A Nucleolus-Based Quota Allocation Model for the Bitcoin-Refunded Blockchain Network**. Rio de Janeiro, 2020. 78p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Mining Bitcoins is an uncertain activity, and to perform it, players must compete in a process known as Proof-Of-Work. A miner may spend months or even years without positive cash flows on this process, while still incurring in the associated costs. This outcome has the possibility to drive them away from the technology, and the departure of members affects the network itself, as it cannot survive without the presence of miners. This work proposes to study the sharing of rewards in structures already presented in the network: miners joining forces and taking place in mining pools, sharing revenues and costs, thus having positive cash flows more often, reducing variability in gains. The revenues and costs are modeled, and a stochastic optimization model is proposed to find the optimal allocations that guarantee that all members stay within the pool. This group of miners is characterized by a coalition, studied through Game Theory. The behavior of the players is also subject of this study, and a monetary risk measure, by the form of CVaR (Conditional Value at Risk) is used to represent the miner's risk profile and consequences to the optimal allocations. While there is no strict benefit from being part of a pool for a single block, there is financial gain when looking at multi-period, and the average time to correctly guess a hash decreases when players join forces in a pool. A gain in mining probability by being in the pool would raise the average reward of the coalition and allow for financial benefit even in single period. We observe that intuitive sharing allocations such as through computational power and equally dividing rewards may not guarantee the stability of the pool, mainly when longer periods of time are considered. Said stability is possible in the future without fixed incomes, but with changes to the variable rewards and the costs of mining. Lastly, three different objective functions representing three ideas to share the rewards within the nucleolus are compared and a method is proposed to collectively use at least two of them, aiming increased "fairness" in the sharing of rewards.

Keywords

Blockchain Network; Bitcoin; Cooperative Game Theory; Stochastic Programming; Conditional Value-at-Risk.

Resumo

Bolonhez, Eduardo Mauro Baptista; Santos, Bruno Fanzeres dos; Silva, Thuener Armando da. **Um Modelo para Alocação de Quotas baseado em Nucelolus para a Rede Blockchain Remunerada por Bitcoin**. Rio de Janeiro, 2020. 78p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Minerar bitcoins é uma atividade incerta, e para realizá-la, os participantes competem em um processo chamado *Proof-Of-Work*. Cada participante pode passar meses ou até anos sem fluxos positivos de caixa, enquanto os custos se mantém. Isto pode afastá-los da tecnologia e a saída de membros afeta a própria rede, que não sobrevive sem a presença de mineradores. Este trabalho propõe estudar o compartilhamento de recompensas em estruturas já existentes na rede: mineradores se juntando em pools de mineração e dividindo receitas e custos, assim diminuindo a variabilidade e gerando fluxos positivos de caixa mais constantes. A receita e custos são modelados, e um modelo de programação estocástica é proposto para encontrar as alocações ótimas que garantem a permanência dos membros no pool. Este grupo de é caracterizado por uma coalizão, estudado através de Teoria dos Jogos. O comportamento dos jogadores também é de estudo neste trabalho, e uma medida monetária de risco, na forma de CVaR (Conditional Value at Risk) é usada para representar o perfil de risco do minerador e as consequências para as alocações ótimas. Embora não haja benefício estrito em fazer parte do pool para um único período de análise, há ganho financeiro quando se analisa em múltiplos períodos, e o tempo médio para se acertar um hash diminui quando os participantes se juntam em um pool. Um ganho na probabilidade de mineração ao fazer parte de um pool aumentaria a receita média da coalizão, trazendo ganhos financeiros mesmo em um único período de análise. Divisões intuitivas de recursos, como por poder computacional ou igualitária podem não garantir estabilidade do pool, principalmente considerando períodos longos de tempo. Tal estabilidade é possível em um futuro sem receitas fixas de mineração, se ocorrerem também mudanças nas receitas variáveis e custos. Três funções objetivo diferentes representando três idéias de partilha de recompensa são comparadas e uma metodologia é proposta para uso conjunto de pelo menos duas destas, com objetivo de aumentar a “justiça” na divisão das recompensas.

Palavras-chave

Rede Blockchain; Bitcoin; Teoria dos Jogos Cooperativos; Programação estocástica; Conditional Value-at-Risk.

Table of contents

List of figures

List of tables

List of Abreviations

N – Number of players in the pool

$\mathcal{N}$ – The grand coalition in the pool

S – subgroup of N, a coalition such that $S \subseteq \mathcal{N}$

n – A player within N

T – The number of blocks considered

n_{header} – average number of block headers a player n will guess

q_S – The probability that S has to mine the next block

h – hashrate of a player n in the bitcoin network

hp – hahpower of a player n in the bitcoin network

$\tilde{\gamma}_{t,S}$ – Binary random variable indicating the success or the failure in mining a block, modelled by a Bernoulli distribution

$\tilde{\Gamma}_S$ – Random variable indicating the number of blocks mined by S

$\tilde{\pi}_t$ – Revenue from mining a block

π^f – Deterministic income per block

$\tilde{\pi}_t^v$ – The variable income per successfully mining block $t \in \{1, \dots, T\}$

C_S – The cost a coalition S incur in try to guess the next block $t \in \{1, \dots, T\}$

$R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})$ – The profit of a coalition S $\subseteq \mathcal{N}$ in the next $t \in \{1, \dots, T\}$ blocks

$R_N(\tilde{\pi}_t, \tilde{\gamma}_{t,N})$ – The profit of the grand coalition N in the next $t \in \{1, \dots, T\}$ blocks

$\Phi(A)$ – Characteristic (or value) function of a cooperative game for a stream of profit A

x_n – Allocation of player n obtained by the optimization model

ϵ_S – Vector containing the numerical difference between being in or out of the pool for every S $\subseteq \mathcal{N}$

ϵ – Maximum value of ϵ_S that is minimized in the optimization model, indicating if there is nucleolus or not for S $\subseteq \mathcal{N}$

$O(\Phi)$ – Value of a coalition S $\subseteq \mathcal{N}$ has when it is not mining in the pool

$P(\Phi)$ – Value of a coalition S $\subseteq \mathcal{N}$ has when it is not mining with the pool

$\mathcal{R}(x_n, R_N(\tilde{\pi}_t, \tilde{\gamma}_{t,N}))$ – Income a player n $\in N$ obtains by participating in the pool, with a quota of x_n

1 Introduction

Like any other tool or technology, what is currently used as money (or monetary value for trading) has had many representations throughout history. Gold, Silver, Copper, metal coins, paper currency, debit cards, etc. (Eichengreen, 2019). The monetary system the world currently uses is the fiduciary one, in which there is no financial coverage (like there used to be with gold/dollar). This system is based on the trust that the ones owing the money will pay their debts. Also, money exchanges are the responsibility of governments and banks, the first being able to print new money at will.

These aspects generate criticism from different economists. They say that being a very centralized model, together with the fiduciary aspect, benefits inflation and at long-term, creates unemployment (Friedman, 1977). Also, individuals have no real knowledge of the amount of money available being transferred through the system, but the institutions that they use have full knowledge of their monetary activity. With more and more scandals arising from the lack of digital privacy, leaking of personal data (Newman, 2015) and misuse of client's money (Naheem, 2015), an increasing number of people want control to be, at least partially, back in their hands.

There is a technology with the potential to mitigate these problems. That technology is the bitcoin, backed by its own *Blockchain* network. bitcoin is a cryptocurrency proposed in a 2008 article by Satoshi Nakamoto (Nakamoto, 2008), a pseudonym of unknown origin. It does not have financial physical coverage nor works by the currently used fiduciary system. It uses real people to verify each transaction in its *blockchain* network, called the miners. Each transaction needs to be verified and registered (within chained blocks), and thus the *blockchain* works as a "*ledger*". The network is maintained, secured and expanded by the process known as Proof-Of-Work (Nakamoto, 2008) in a decentralized environment run by a network of computers (Muftic, 2016), in which the miners take part. In this process, every mining participant takes part in a "guessing game" of information presented in the new block, using computational power. The network is not hosted in any specific server in the world, but is present in every computer that uses it, making it more difficult to be attacked, since a malicious agent would have to attack all computers to succeed. Each specific *blockchain* "ledger" has its own types of validation and consolidation, but in principle they do it via distributed voting (Mattila, 2016), (Naucler, 1950) through a consensus on the state of the ledger.

The miners are the ones that maintain the network operational and vote (with their computational power) for changes. Without them, people would retain their bitcoins in their private wallets, but would not be able to negotiate them. In the absence of miners, no bitcoins can be traded, and the network stops.

Taking part in the Proof of Work demands high computational power as the bitcoin network adjusts the difficulty of the information to be "guessed" so that the time of completion of each block remais approximately 10 minutes (Dwyer, 2015). Therefore, miners with higher computational power (in the world of bitcoin, translated as "*hashpower*") are more likely to succeed. Participating in this game without having a fixed income creates great uncertainty as the participant could spend months, or even years, without positive cash flows. One solution is to be a part of a mining pool (Dev, 2014). A mining pool constitutes a coalition (Lewenberg, 2015), a particular case of a cooperative game. By joining with other miners, players share revenues and costs while also raising their chances of successfully mining. The revenue could be lower in value than mining a block alone, but it is a trade-off. The miner receives lower revenues more frequently rather than receiving a larger revenue occasionally. Players remain honest in the network due to incentives provided by game theory (Back, 2016). Studying the interactions between players in this competitive environment is of interest so that there are incentives for the players to remain in the network, and thus maintaining Bitcoin itself alive (Kiayias, 2016), (Eval, 2013). Game theory is in the core of bitcoin since its birth, when Nakamoto analyzed incentives in a simple revenue distribution model.

Even though bitcoin was born to decentralize the payment environment, dangers are presented at each extreme scenario. A "51 percent attack" (Kiayias, 2016), (Eval, 2013), where a pool retains a majority of the network and therefore controls it, is possible, but not desirable. Many sharing methods are available throughout game theory (Proportional, Sharpley Value, Nucleolus etc.), and this work proposes to analyze a sharing method that brings more value to each miner when participating in a pool, giving him incentives to remain operating in the network. As the European Commission states (Muftic, 2016), 'the major contribution of Bitcoin is the solution of how to establish trust between two mutually unknown and unrelated parties to such extent that sensitive and secure transactions can be performed with full confidence over an open environment, such as the Internet'. The "fairness" within the sharing of revenues is also at the center of this study.

In this work, we analyze the sharing of rewards in the bitcoin-refunded blockchain network through Cooperative Game Theory. The allocation of rewards is based on the nucleolus approach for a group of miners forming a coalition. That coalition is represented by a pool in the network. The nucleolus is the set of all allocations where players in the pool have a higher value remaining together than mining alone, and such preference is measured by a value (or characteristic) function. Players in a pool might reduce their amplitude in gains, but receive rewards more often, reducing variability in gains. An optimization model is proposed to obtain the optimal allocations, using the probability to mine the next $t \in \{1, \ldots, T\}$ blocks, as well as the profit to mine and the risk profile (in the form of the Conditional Value-At Risk) in the context of the network.

An illustrative example is studied, and the optimal allocations for 3 (three) different number of blocks ahead are obtained. The existence of the nucleolus for each case is shown and justified, and an extra gain in probability to mine in the form of a function (dependent on the number of players in the coalition) is proposed and the new optimal allocations obtained. The stability of the pool for intuitive allocations is also tested, and compared for the different number of blocks. Lastly for the illustrative example, a visual representation of the core is obtained, showcasing the value in mining with the pool.

The model is then tested with a real setup of the Network from 2019, showing results that corroborate with the findings in the illustrative example. The future without fixed incomes and the stability of the pool are analyzed, indicating alternatives for the nucleolus to be maintained in that scenario. As a last study, we propose 3 (three) separate objective functions to be used in the optimization model, aiming to treat the "fairness" of the rewards sharing. A methodology is proposed to "fairly" allocate the rewards while maximizing the value in the pool.

This work is laid out as follows. Section 2 describes the bitcoin and *Blockchain* network and how mining pools bring more value to players. Section 3 describes the mathematical models used in this work, such as the main assumptions, results for profit, revenue and costs of mining and the probability to mine and the monetary risk measures. In Section 4, game theory used is explained and the optimization model is presented. Section 5 shows the results of this study, detailing the base case chosen and the differences for the sensibility analysis, as well as the real case application. Finally, Section 6 concludes the work and discusses extensions and future research.

2
Bitcoin and Mining

The present chapter aims to introduce the basic concepts behind Bitcoin and Blockchain as well as mining pools and how they operate and some reward sharing methods for said pools.

2.1
The history of Bitcoin

A Blockchain is a digital distributed ledger that contains a log of information in chronological order that increases with time. The information is gathered into structures called blocks, each one containing transactions and fees. Blocks are added to the Blockchain by miners, responsible for verifying and adding data to the chain. Each block has a timestamp and is linked to the previous one, thus forming a chain that is validated by the network.

The concept of a digital ledger was born concomitantly to Bitcoin with Satoshi Nakamoto's article (Nakamoto, 2008), but indeed they are different technologies. Bitcoin is a DApp (decentralized application operating in peer to peer), a solution that runs on a Blockchain to serve different purposes. Nakamoto's article proposes a “peer-to-peer version of electronic cash”, that would allow online payments without the need for a financial institution, while preventing double-spending (spending the same Bitcoin twice). Nakamoto solves the issue of double-spending by proposing consensus and a universal ledger. Figure 2.1 shows the difference between centralized (current model) and decentralized models, using Bitcoin and Blockchain. The term "B" stands for blockchain, and "SC" stands for "Smart Contract". A Smart Contract contains lines of code that execute predetermined actions when certain terms and conditions are met.

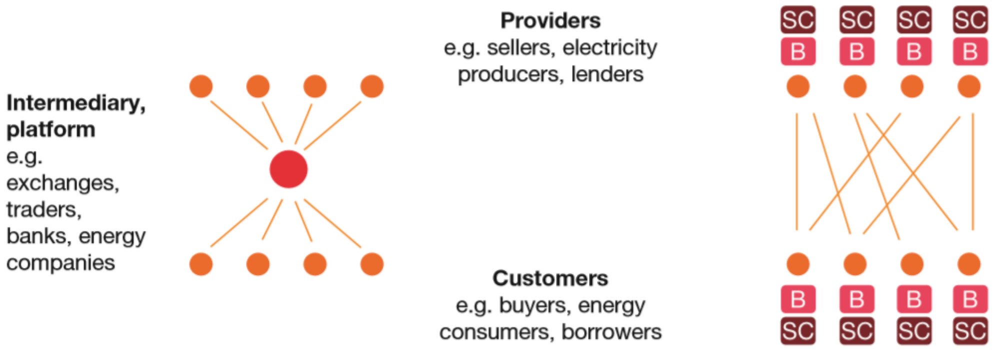

Intermediary, platform
e.g. exchanges, traders, banks, energy companies
Providers
e.g. sellers, electricity producers, lenders
Customers
e.g. buyers, energy consumers, borrowers
SC
B

Figure 2.1: Centralized versus decentralized system (PwC, 2015).

To be able to perform as designed, the Bitcoin network relies on the miners operating in the Proof-Of-Work method. In this method, participants use computational power to guess the correct hash of the subsequent block. A hash algorithm turns an arbitrary amount of data into a fixed-length hash. Each block contains a hash on its header which serves for identification. Miners participate in the Proof-Of-Work with cutting-edge hardware that can perform several hash calculations per second, the *hashrate*. The percentage of computational power a player owns divided by the total power in the network is called *hashpower* (hp).

As the total hash of players in the network fluctuates over time, the Bitcoin network adjusts the difficulty (or the amount of hash generated by the miners to guess a block's header) to have a new block mined on average every 10 minutes. The miner that correctly guesses the hash claims the rewards. Once a miner declares having successfully mined a block, the other players can verify if that block is valid, and if so, the block is added in a chronological chain subsequent to the previous block mined. Figure 2.2 shows the difficulty, in hashes, to mine a block since the beginning of the network.

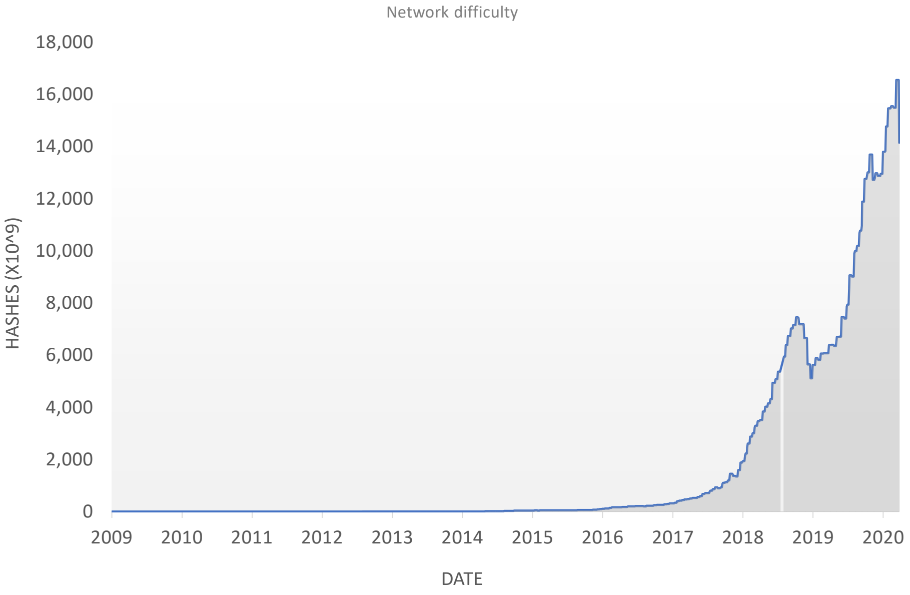


Figure 2.2: Bitcoin's Blockchain difficulty to mine.

Miners create new blocks that contain transactions from users throughout the world, but for a user to be able to trade Bitcoin for goods, he needs to have a wallet that "holds" their Bitcoins. A wallet is a user's private key, and a public key is used to identify the user transactions. The concept of public-key cryptography is widely used to certify that only the owner of a private key can access his data. Also, the use of public and private keys mathematically related ensures authentication (Diffie, 1976), (Merlinda, 2019), and guarantees that an action is performed by a player with the right to do it, or the real owner of the currency to be traded.

To illustrate the evolution in the use of the technology, figure 2.3 shows that Bitcoin's market value has greatly changed ever since its debut, while figure 2.4 shows the numbers of transactions in the network, per day, since 2009.

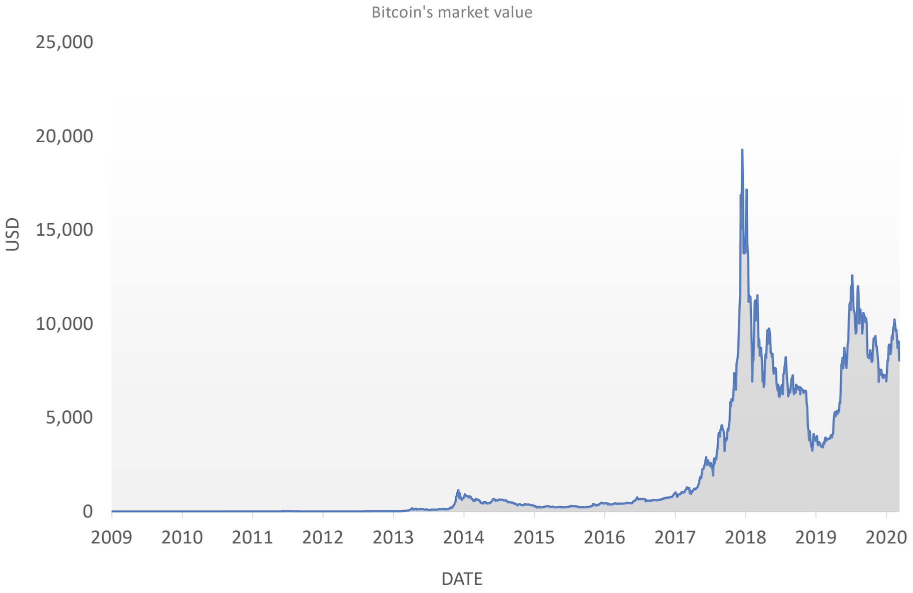


Figure 2.3: Bitcoin's worth in dollars, since its beginning.

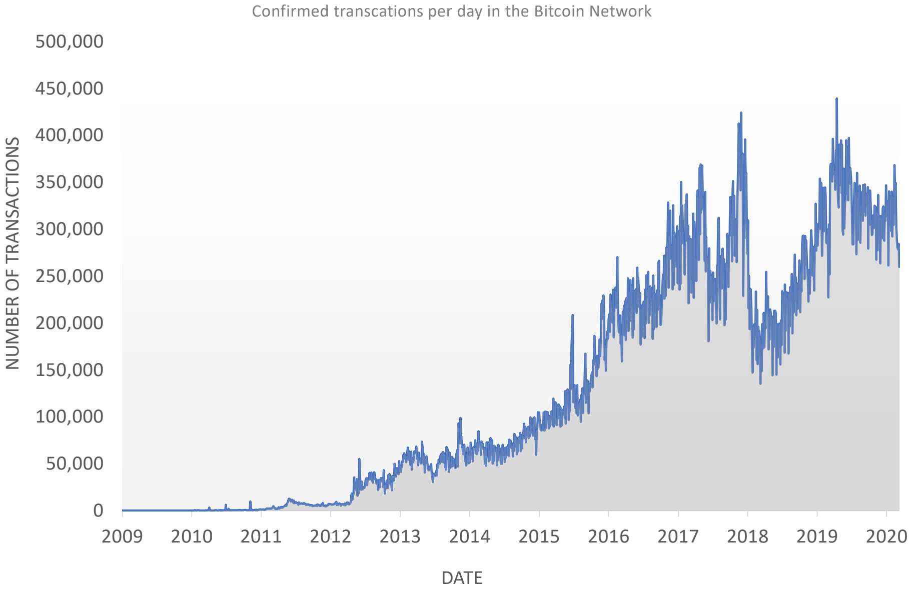


Figure 2.4: Number of daily transactions at the Bitcoin Network, since its beginning.

Bitcoin has been going through ups and downs, with the coin's value reaching its highest value (almost U$D 19,000) in late 2017. When the price of Bitcoin reached its lowest in 2019, there was also a drop in the network difficulty. These changes might indicate that miners were turning off their equipment or not using them 24 hours per day (in direct relation to a drop in the market price) or not even mining in the Bitcoin Network, using their computers on other networks, like BitcoinCash. Even with the drop in the network difficulty, the confirmed transactions per day were rising, showing that there was a demand by the users of Bitcoin. The Bitcoin market is relatively new, and it may be seeing its first steps towards an equilibrium. For that reason and to keep the network running, it is crucial to study and generate ways in which the miners do not turn off their equipment or change networks even at times of adversity. Also important is to assure to those miners that the act of mining is still profitable for them, since they are the ones that register new transactions in the network. In order to reduce the uncertainty in mining, one such way might be joining forces in a pool.

2.2 Mining pools

Miners use their computers to keep the network functioning. Without them, Bitcoin owners would retain their money in their private wallets and would not be able to commercialize it. The act of mining Bitcoin is risky and uncertain. Guessing the correct hash of a block, while competing against many other participants, produces constant costs and has a high uncertainty about the profits.

The miner that finds the specific hash earns a fixed and a variable revenue in Bitcoins. The former is currently 12,5 Bitcoins (in March 2020), and this value is halved every 210.000 new blocks created [1]. The latter is composed by the sum of taxes from the transactions of a mined block. Once the halving process reduces the fixed income to almost zero, the miners will earn only the variable revenue.

When Bitcoin started, there were fewer miners, and the difficulty was lower (as shown in figure 2.2). At those times, miners had more humble equipment to mine in the network, and likely did it from their homes. As time passed and the technology attracted more supporters, the network rose the mining difficulty, demanding more powerful computers. Those lone miners saw their hash power dilute to minimal values, and with a lower hash power,

[1] https://qz.com/681996/everything-you-need-to-know-about-the-bitcoin-halving-event/, accessed november 26th, 2019

the volatility in profit rises. That means: with greater difficulties to mine, players with lower hashpower have almost no chances of mining a block, with the possibility of spending months or even years without a positive cash flow. Miners with higher hashpower are more likely to succeed.

One alternative to ensure more constant returns (Kroll, 2013) is to participate in a mining pool, where a group of miners share their revenues and costs, increasing their chances of positive cash flows. When mining within a pool, a miner's reward will be lower for a block than if done by himself, but even when he does not mine (but someone else in the pool does), he will also have a profit. Lone miner exchanges fewer full block rewards for more constant, although lower, rewards. As these complete rewards tend to be widely spaced in time, it is a good trade, which enables a miner to remain in the network and users of Bitcoin to continue to transaction their currency. The choice to be part of a bitcoin mining pool implies a diversification scheme. As (Chatzigiannis, 2019) states, if a miner has the chance to select between various cryptocurrencies to lend his computational power, he is diversifying his potfolio in a similar way as proposed by the Markowitz Modern portfolio theory (Markowitz, 1952). There are some known methods of sharing rewards between members of a pool in Bitcoin, illustrated in the next section.

2.3 Reward sharing

The concepts of "difficulty" and "shares" must be explained before listing the rewarding methods most commonly used in Bitcoin. The Bitcoin network has a *difficulty* (hereby denoted D) to mine a block at any given time, which translates to how much hash power has been deployed by the miners in the network at a given time. Now, a *share* is an accounting method to keep the miners honest and to be used as a tool for the sharing of rewards in a pool, and has no actual "value". The number of shares found by a miner is proportional to the amount of hashes the miner calculated in an attempt to find a block for the pool (Rosenfeld, 2011), Lastly, "rounds" is the time between one block found by the pool, to the next.

On the topic of how to share the rewards, each mining pool is free to choose the method it desires and pleases its players the most. Also, to be able to mine for a pool, a player must register an account within the pool, download a software (like CGminer or BFGminer) and configure it with the hardware used. Lastly, the player has to indicate its wallet adress to receive any payout. One example of sharing in bitcoin mining pools is the Pay-Per-Share method. In this method, an operator absorbs all the variance from mining and rewards immediately each player according to the shares contributed to the pool. However, once the next block is successfully mined, the operator keeps all the rewards for himself. It is the reward system with the highest risk for the operator, with a high chance of the pool going bankrupt, since the operator is always paying miners, even if they fail to mine the next block.

Another well-known method is the proportional, where an operator centralizes decisions in the pool and receives a fee, which is based on the total reward earned in mining the block. This operator then divides the rest of the amount according to the number of shares each player contributed. The operator and participants only receive their money when the pool mines a block.

A miner can choose when to mine for a pool, deciding where he contributes with his hash rate. This can lead to *pool hopping* and effectively benefits dishonest players and is inherent to proportional share. *Pool hopping* comes from the change in value of shares with time. Those submitted early in a round are more worthy than those submitted later, and if a pool takes too long to mine a block, there will be more shares to be rewarded.

The Slush method was designed to combat *pool hopping* and is a score-based method. It is based on the proportional approach, but here, each share credits a player with a score that depends on the time spent by a given player in a round, and the rewards are later distributed according to this score. The more time has passed, the higher the score.

A more modern method that proposes to solve *pool hopping* is the Pay-Per-Last-N-shares. Here, the rewards are distributed not according to the shares submitted by players right after the previous block was found, but only for those who did it recently (in the last N shares), and thus not allowing players to benefit from being “early in the round” and leaving the pool only to collect the rewards if a block is mined. In reality, with this behavior , the pool is no longer applying the concept of “rounds”.

3 Miners Net Revenue Modelling and Measure of Value

With the ideas behind Bitcoin, Blockchain and Mining Pools presented, we now discuss some of the concepts used to propose our model. The probability to mine is defined, and the Binomial and Bernoulli Probability models associated with the nature of mining and their influence are discussed. We also show the Bitcoin reward system and revenue, costs and profit of mining. The risk measure tool of choice is the Conditional Value-at-Risk (Rockafellar, 2002), which is also discussed. Lastly, the definitions of a coalition are presented, and we propose, gathering all that concept, a Measure of Value for our work.

This work studies a bitcoin network with N total players of interest participating in the Proof-of-Work, with n being a player within $\mathcal{N} = \{1, 2, \ldots, N\}$. A coalition S is a subgroup of those N players, i.e., $S \subseteq \mathcal{N}$. For nomenclature purpose, hereinafter, we refer to $\mathcal{N}$ as the grand coalition and the set of all coalitions by $\rho(\mathcal{N}) = \Big\{\{1\}, \{2\}, ..., \{N\}, \{1, 2\}, ..., \mathcal{N}\Big\}$, i.e., the powerset of $\mathcal{N}$. We call N^T the total of players and N^R the "rival players", or players in the Bitcoin network that are not mining in the coalition, such that $N^T = N + N^R$. For simplicity, they are represented by a single player called "rivals" for representation in modelling the problem. The number of blocks considered in the analysis are represented by T.

The network difficulty (D) is adjusted according to the computational power (hp) of each player. A player that has a set of machines working in the *Proof-of-Work* that produces 100 hashes per second, with the network total being 1000 hashes per second, has 10% of the hashpower in the network.

3.1 The probability to mine

According to (Rosenfeld, 2011), a miner with hashrate h, mining for a period of time τ will calculate a total of $h\tau$ hashes, and so will guess on average $n_{headers}$ headers of a block, being D the difficulty of the network at any given time and for a block t:

$$n_{header} = \frac{h\tau}{2^{32}D}. \tag{3-1}$$

The hashrate h is the number of hashes calculated per second by the machine used by the player. It is, therefore, different than *hp*, hereby proposed as the percentage of computational power a player owns in relation to the grand total of hashrate in the bitcoin network: $h = hp_n \times H$, where H is the sum of hashrates for all players in the Bitcoin network: $H = \sum_{n \in N} h_n$. It can be said that the probability q_S that a coalition $S \in \rho(\mathcal{N})$ successfully mines the next block is directly related to the hashpower of each player $n \in S$:

$$q_S = \sum_{n \in S} hp_n + f(S), \qquad \forall\ S \in \rho(\mathcal{N}), \tag{3-2}$$

where $f(S)$ is a function that maps the increase in the mining likelihood of coalition $S \in \rho(\mathcal{N})$ by playing as a group. In other terms, it shows how much working together raises the probability to mine a block. If $f(S)$ is equal to zero, then the probability to mine is the sum of the hashpowers (*hp*) of each player in *S*. The rate of success in mining the next block can be adequately characterized by the result of a Bernoulli trial with probability q_S, since each new block has a result independent from the previous one, with the same probability q_S. In this work, we denote $\tilde{\gamma}_S \in \{0, 1\}$ as a binary random variable indicating the success (returning value of 1) or the failure (returning value of 0) in mining a block, modelled by a Bernoulli distribution with probability q_S, i.e., $\tilde{\gamma}_S \sim \text{Bernoulli}(q_S)$.

In fact, note that in the context of a set of T blocks ahead, the joint probability representation becomes a Binomial distribution, based on a series of independent Bernoulli trials. For nomenclature purposes, hereinafter, we refer to $\tilde{\gamma}_{t,S}$ as the probability that a coalition $S \in \rho(\mathcal{N})$ mines a block $t \in \{1, \ldots, T\}$, and $\tilde{\Gamma}_S \in \{1, \ldots, T\}$ a random variable indicating the number of blocks mined by coalition $S \in \rho(\mathcal{N})$ within the next T blocks, i.e.,

$$\tilde{\Gamma}_S = \sum_{t=1}^{T} \tilde{\gamma}_{t,S} \sim \text{Binomial}(T, q_S), \qquad \forall\ S \in \rho(\mathcal{N}). \tag{3-3}$$

3.2 Revenue, Costs and Profit on Mining

Reflecting the dynamics of payment per miner in a block (fixed + variable), the revenue from mining a block $t \in \{1. \ldots, T\}$ is:

$$\tilde{\pi}_t = \pi^f + \tilde{\pi}^v_t, \qquad \forall\ t \in \{1. \ldots, T\}. \tag{3-4}$$

In Equation (3-4), $\tilde{\pi}_t$ is the revenue from mining the block $t \in \{1.\dots,T\}$, π^f is the deterministic income per block, which, in this work, without loss of generality, is assumed fixed within the maturity of analysis, since the length of steps ahead is assumed sufficiently small for the halving process does not takes place, and $\tilde{\pi}_t^v$ is the variable income per successfully mining block $t \in \{1.\dots,T\}$.

Currently, a miner that successfully mines a block receives a fixed reward (countdown) of 12.5 BTC (Bitcoins) as already stated in Section 2.2. Bitcoin's network adjusts the difficulty of the *Proof of Work* so that each block takes 10 minutes on average for the header to be guessed. In this work it is assumed, for simplification, that each block takes exactly 10 minutes to be mined. For periods of time small enough to maintain π^f unchanged by the halving process or the adjusted network difficulty to express any considerable changes, it is acceptable to consider that a block is mined every 10 minutes. The variable income per block is a random variable, and works as the sum of all the fees paid by each individual that has a transaction added to the block mined.

Costs of mining are represented by operational costs (for example, costs of electrical energy consumption) and capital costs (computers and peripherals to make up the mining machines). For the purpose of this work, it is considered that all players use their machines uninterruptible within the maturity of analysis. In this context, let C_S to denote the cost a coalition $S \in \rho(\mathcal{N})$ incur in trying to guess the next block $t \in \{1,\dots,T\}$.

The profit of a coalition $S \in \rho(\mathcal{N})$ in the next $t \in \{1,\dots,T\}$ blocks can be written as the difference between $\tilde{\pi}_t$ (when coalition S successfully mines a block) and the respective costs C_S as follows:

$$R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S}) = \sum_{t=1}^{T} \Big(\tilde{\pi}_t \, \tilde{\gamma}_{t,S} - C_S \Big), \qquad \forall \; S \in \rho(\mathcal{N}). \tag{3-5}$$

Equation (3-5) shows that, whenever the result of $\tilde{\gamma}_{t,S}$ is zero for some $t \in \{1,\dots,T\}$, the coalition S does not mine the block t and, hence, it has negative cash flow.

3.3 Measure of Value

The "gain" a player obtains from joining a pool is measured by a Measure of Value (Φ), here proposed as a convex combination between the expected return and a risk measure tool, named the Conditional Value-at-Risk ($CVaR_\alpha$) (Fanzeres, 2015):

$$\Phi(A) = \lambda CVaR_\alpha(A) + (1-\lambda)\mathbb{E}[A]. \tag{3-6}$$

where λ is the deterministic weight associated with the risk aversion of a player, and A the return of the player at any given scenario. The higher λ is, the more risk averse the player is. When mining alone, a player is likely to have positive cash flows spread over time with long periods between them (depending on his hahspower, though it also not likely that a single player owns enough computational power in the network to mine successfully over short periods of time). The influence of these payments must be considered and CVaR acts in this way.

Conditional Value at Risk ($CVaR_\alpha$) is a coherent risk measure (Artzner, 1998) (a risk measure that satisfies four axioms: translation invariance, subadditivity, positive homogeneity and monotonicity) that assess the expected loss in the worst $(1-\alpha)$-fraction of cases. Essentially, this is the expected value in the area to the tail of the density up to the $(1-\alpha)$-quantile (as seen in figure 3.1). Monetary risk measures are important tools to simulate the game with the risk profile of each player, when investors try to design portfolios on the basis of a comfortable trade-off between the risk of loss and the possibility of profit. The four axioms that $CVaR_\alpha$, as a coherent risk measure (Street, 2009), follows are:

- Monotonicity: A portfolio A_2 that has always better returns than another portfolio A_1, should always have less risk as well.
  If A_1, $A_2 \in L$ and $A_1 \leq A_2$, then $CVaR_\alpha(A_1) \leq CVaR_\alpha(A_2)$
- Superadditivity: The diversification principle. The combination of two portfolios cannot be worse than the sum of them separately.
  If A_1, $A_2 \in L$, then $CVaR_\alpha(A_1 + A_2) \geq CVaR_\alpha(A_1) + CVaR_\alpha(A_2)$
- Positive Homogeneity: The risk of a position is proportional to its size.
  If $\delta \geq 0$ and $A \in L$, then $CVaR_\alpha(\delta \text{ A}) = \delta\ CVaR_\alpha(\text{A})$
- Translation Invariance: Adding a fixed amount "a" of capital reduces risk by the same amount "a".
  $CVaR_\alpha(\text{A + a}) = CVaR_\alpha(\text{A}) + \text{a}$

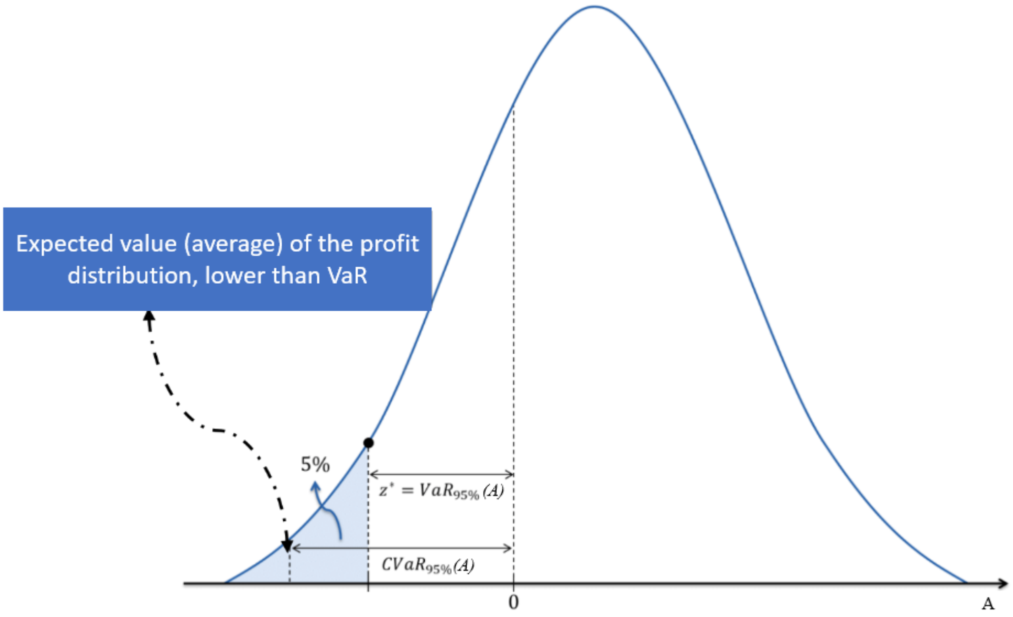

Expected value (average) of the profit distribution, lower than VaR
5%
z* = VaR95% (A)
CVaR95%(A)
0
A

Figure 3.1: Visual representation of VAR and CVaR for a probability distribution.

$CVaR_\alpha$ can be obtained in many ways. The more often used is represented by equation (3-7):

$$CVaR_\alpha(A) = \max_{z \in \mathbb{R}} \left\{ z - \frac{\mathbb{E}[z - A]^+}{(1 - \alpha)} \right\}. \tag{3-7}$$

It can also be interpreted as the average of all values between the a random variable A and the Value-at-Risk (VaR) of the distribution, as (Street, 2009) states:

$$CVaR_\alpha(A) = \mathbb{E}[A \mid A \leq VaR_\alpha(A)]. \tag{3-8}$$

And as stated in (Rockafellar, 2002), the CVaR$_\alpha$ of a discrete random variable $\tilde{A} \sim \{A_\omega\}_{\omega \in \Omega}$ can be obtained as the result of the following optimization problem:

$$CVaR_{\alpha}(\tilde{A}) = \max_{z,\delta_{\omega}} z - \frac{1}{(1-\alpha)} \sum_{\omega \in \Omega} p_{\omega}\delta_{\omega} \tag{3-9}$$

subject to:

$$\delta_{\omega} \geq 0, \qquad \forall\, \omega \in \Omega \tag{3-10}$$

$$\delta_{\omega} \geq z - A_{\omega}, \qquad \forall\, \omega \in \Omega. \tag{3-11}$$

with p_{ω} the probability of scenario $\omega \in \Omega$. Combining equations (3-5) and (3-6), we obtain the complete representation of Φ and the associated value function v of a coalition $S \in \rho(\mathcal{N})$:

$$\begin{aligned} v(S) &= \Phi(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})) \\ &= \lambda\, CVaR_{\alpha}\Big(R_S\big(\tilde{\pi}_t, \tilde{\gamma}_{t,S}\big)\Big) + (1-\lambda)\,\mathbb{E}\Big[R_S\big(\tilde{\pi}_t, \tilde{\gamma}_{t,S}\big)\Big], \qquad \forall\, S \in \rho(\mathcal{N}) \end{aligned} \tag{3-12}$$

Equation (3-12) denotes that, the higher λ, more $CVaR$ is influencing the value of Φ. The choice behind this convex combination of risk and return is done exactly to allow us to measure how the value of a player in the pool might differentiate depending on various risk profiles.

4 Cooperative Game Model

This chapter elaborates on the main definitions of cooperative Game Theory, justifying its use in this work, as well as showing a number of allocation methods in Game Theory and the formation of the nucleolus of the game. On this topic, a mathematical definition of the nucleolus is proposed and applied to the measure of value Φ, and the optimization model is established with the theory discussed so far.

4.1 Game Theory

The study of interactions choices between economic agents with different preferences and the outcomes of those interactions is called Game Theory. The outcomes can be produced by a cooperative game (where players interact directly towards common goals) or a non cooperative game (where players dispute towards the best outcome for each individually).

The formalization of Game Theory dates back to 1928, with the publication of *On The Theory of Games of Strategy*, by John von Neumann [1]. There, von Neumann tried to answer how, in a set of N players, one of them could play in order to achieve the most advantageous result. This paper pushed forward the field, stating that in a two-person game, it is always possible to find an equilibrium from which neither player should deviate unilaterally.

As Princeton University Press described in 2004, *The Theory of Games and Economic Behavior*, published in 1944 by the same John von Neumann with the economist Oskar Morgenstern, is “the classic work upon which modern-day Game Theory is based” [2] and advances in the work of the previous book.

Later, John Nash further expanded von Neumann’s contribution with his work *The Theory of Games of Strategy* with the *Nash equilibrium*, that states that every finite N-player in a non-zero-sum game has a clearly-defined strategy (equilibrium points in N-person games). In pa-

[1] https://towardsdatascience.com/game-theory-history-overview-5475e527cb82?gi=5f8d9c5f30ae, accessed february 11th, 2020

[2] https://press.princeton.edu/books/paperback/9780691130613/theory-of-games-and-economic-behavior, accessed february 11th, 2020

pers (Equilibrium points in n-person games, 1950), (The Bargaining, 1950), (Non-cooperative Games, 1951), (Two-person, 1953) he helped to define modern Game Theory, where for instance John Nash helped formulate solutions to cooperative games (Non-cooperative Games, 1951), (Two-person, 1953).

When two or more players interact aiming for better results in any situation, this is called a cooperative game (Jezic, 2016). A cooperative game is a pair (N,v) where v is commonly referred as the characteristic, or value function (Jezic, 2016). In the context of this work, v has already been defined in equation (3-12). As stated in chapter 3, a coalition $S \in \rho(\mathcal{N})$ is a subgroup of N total players participating in a pool. A fair distribution of the resources obtained by a coalition is one of the major fields of study in Game Theory, in particular, how to precisely define the meaning of fair. A characteristic, or value function is needed, and it must satisfy:

$$v(\emptyset) = 0, \tag{4-1}$$

$$v(\mathcal{N}) \geq \sum_{i=1}^{N} v(i). \tag{4-2}$$

Which means that the benefit of an empty coalition must be zero and that of N players must be at least the sum of the benefits of the individual players if no coalitions is formed (Barron, 2008). In other words, all the players must perform better when they are joining forces, than if they are playing on their own. That is represented by superadditivity (discussed in section 3.3).

4.2 Allocation Methods

There are several works in technical literature proposing how to share the rewards between participants of a coalition. The *proportional sharing* method is the most widely known and also most intuitive: each player is rewarded by their contribution to the coalition, thus being independent of coalitional synergic effects (Freire, 2017). For instance, in the context of this work, for a given coalition $S \in \rho(\mathcal{N})$, the proportional share x_n of a given player $n \in S$ can be defined as: $x_n = \frac{hp_n}{\sum_{n \in S} hp_n}$. Some results in this work are obtained by testing this allocation policy. Hereinafter, we refer to it as the “relative hashpower".

An alternative allocation method that takes into account the coalitional synergic effects is called the *Sharpley Value*. It was proposed by Lloyd Shapley in 1953 (Shapley, 1953) and aims to “allocate an amount proportional to the benefit each coalition derives from having a specific player as a member” (Barron, 2008). For a given coalition $S \in \rho(\mathcal{N})$, it states that the allocation (x_n) of a player $n \in S$ is a Shapley value if:

$$x_n = \sum_{S \subseteq \mathcal{N} \setminus \{n\}} \frac{|S|!(N - |S| - 1)!}{N!} \Big(v\Big(S \cup \{n\}\Big) - v(S)\Big), \tag{4-3}$$

The Shapley Value satisfies the axioms of Symmetry, Dummy Player and Additivity (Manea, 2017). Dummy player happens when for any coalition S in which a player n is not part: $v(S \cup \{n\}) = v(S)$. In other words, a dummy player does not contribute to the coalition. The additivity axiom states that, if v_1 and v_2 are two different games players participate, then: $x(v_1 + v_2) = x(v_1) + x(v_2)$ (here, x is the Shapley-Value). Or, the Shapley Value (or reward a player has as a result of the sum of two games), is equal to the sum of Shapley Values for those two games. Finally, With symmetry, if two players provide the same benefit to S, than they should have the same worth. Shapley Value tends to increase its complexity as the number of players grows due to its combinatory nature, being this a drawback (Freire, 2017). Also, there is a lack of isonomy (Junqueira, 2007), (Freire, 2017); players with same hashpower might have different costs, and allocation process is affected by players aggregation.This might occur due to players with higher hahspowers being less sensible in the input order in equation (4-3), used to obtain the allocations.

Aiming to minimize the lack of isonomy cited above, Aumann-Shapley is a method that proposes to split the resources into smaller agents in the coalition after just a fraction of the larger players had their allocations optimized (Faria, 2009). In other words, to apply Shapley's Value method after the smaller agents have been served, all players hashpower and costs would be divided in infinitesimal ones, and the Shapley Value would be applied as if each of those divisions were a player itself.

The method used in this work is that of the Nucleolus of a game, described in the following subsection.

4.2.1 Nucleolus of a Game

The nucleolus of a game (Kohlberg, 1971) is the set of all allocations in which the income of a sub coalition in S is higher when joining forces, than alone, meaning that the nucleolus has actions that benefits all players in the coalition. For example, if a coalition has three members, the nucleolus is stable (or exists), if there is an allocation between all members that will always benefit all its participants, bringing a higher value than if they worked alone (or in any sub coalition). When a coalition includes an action that all its members prefer, other than the grand coalition, it is said that the former blocks the latter, breaking the concept of the nucleolus. To propose allocations that keep the nucleolus stable is a problem on how to divide resources, or proposing the optimal sharing of them. The sharing problem is an instance of the general problem of allocating costs (or benefits) among a coalition of players that cooperate on the construction of a shared resource (Freire, 2015).

The sharing of resources obtained through the methodology proposed in this work is represented by the variable x_n, or the allocation of a player $n \in \mathcal{N}$. The allocation of wealth for each player is defined as to not allow short-sale and also that all the players belong to the same pool:

$$\sum_{n \in \mathcal{N}} x_n = 1, \tag{4-4}$$

$$x_n \geq 0, \qquad n \in \mathcal{N}. \tag{4-5}$$

Now, the nucleolus of a cooperative game is hereby defined as the set

$$\mathcal{X} = \left\{ \mathbf{x} \in [0,1]^N \;\middle|\; \sum_{n \in S} x_n R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}}) \succeq_{\Phi} R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S}), \quad \forall\, S \in \rho(\mathcal{N}). \right\} \tag{4-6}$$

Equation (4-6) represents the set of allocation shares ($\mathcal{X}$) where a player has a higher value (measured by Φ) sharing the rewards obtained by $\mathcal{N}$, than out of the great coalition, for any sub coalition of players. Using the measure of value Φ from equation (3-12) together with (4-6), the nucleolus is:

$$\Phi\left(\sum_{n \in S} x_n R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}}) \right) \geq \Phi\Big(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S}) \Big). \tag{4-7}$$

Since $R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})$ and x_n are independent and because of the positive homogeneity nature of CVaR_α discussed in subsection 3.3, equation (4-7) can be written as:

$$\sum_{n \in S} x_n \Phi\Big(R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}}) \Big) \geq \Phi\Big(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S}) \Big). \tag{4-8}$$

With the concept of x_n discussed, let λ be zero (risk neutrality) and μ the expected value of the variable income distribution. In this context, since the random variables $\tilde{\pi}_t^v$ and $\tilde{\gamma_{t,S}}$ are independent for all $t \in \{1, \ldots, T\}$, the profit value of a coalition $S \in \rho(\mathcal{N})$, along with the T blocks ahead resumes to:

$$\mathbb{E}[R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})] = \sum_{t=1}^{T} \Big[(\pi^f + \mu)q_S - C_S\Big] = \Big((\pi^f + \mu)q_S - C_S\Big)T. \tag{4-9}$$

Agents usually take risk-neutral decisions, that is, based only on the expected value of the revenue (in our model, with $\lambda = 0$). However, in this case, we argue that the nucleolus set defined in (4-6) become independent of the number of blocks ahead, hence equivalent to looking only at the next one (as shows Proposition 1, below). In this case, it would ignore the profit of a sufficient revenue stream for the coalition to remain viable and stable.

Proposition 1: For $\lambda = 0$, the nucleolus set ($\mathcal{X}$) defined in (4-6) become independent of the number of blocks ahead T.

Proof: For λ is zero, the profit value defined in subsection 3.3 is given by Equation (4-9). Therefore, for each $S \in \rho(\mathcal{N})$, the main constraint in the nucleolus set ($\mathcal{X}$) defined in (4-6) resumes to:

$$\begin{aligned}\sum_{n \in S} x_n \mathbb{E}[R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})] \geq \mathbb{E}[R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})] &\iff \sum_{n \in S} x_n \geq \frac{\mathbb{E}[R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})]}{\mathbb{E}[R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})]}, \\ &\iff \sum_{n \in S} x_n \geq \frac{\Big((\pi^f + \mu)q_S - C_S\Big)}{\Big((\pi^f + \mu)q_{\mathcal{N}} - C_{\mathcal{N}}\Big)}.\end{aligned}$$

Therefore, for a risk neutral game ($\lambda = 0$), the the nucleolus set ($\mathcal{X}$) becomes:

$$\mathcal{X} = \left\{ \mathbf{x} \in [0,1]^N \;\middle|\; \sum_{n \in S} x_n \geq \frac{\Big((\pi^f + \mu)q_S - C_S\Big)}{\Big((\pi^f + \mu)q_{\mathcal{N}} - C_{\mathcal{N}}\Big)}, \quad \forall\, S \in \rho(\mathcal{N}) \right\}. \tag{4-10}$$

4.3 The Optimization Model

Now that the profit for a miner (equation (4-9)), the probability to mine (equation (3-2)), $CVaR_\alpha$ (equation (3-11)) and the measure of value Φ (equations (3-12) and (4-8)) are defined, we propose the use of a tool that can help obtain the optimal allocations for a player n, in the form of x_n.

We need a tool to measure the difference between being in the pool and not in the pool, and make sure it returns, at least, zero (meaning that participating in the pool brings, at least, the same value as mining out of the pool). Here, for each $S \in \rho(\mathcal{N})$, we introduce the variable ϵ_S, translating exactly the numerical difference described, becoming $\sum_{n \in S} x_n \Phi(R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})) \geq \Phi(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})) - \epsilon_S$

We want the vector $\{\epsilon_S\}_{S \in \rho(\mathcal{N})}$ to be as low as possible. If ϵ_S is lower than zero for all $S \in \rho(\mathcal{N})$, it means that the left-hand side of the equation is higher than the right-hand side, and thus, there is value to participating in the pool. More precisely, the grand coalition brings more profit to the participants than mining outside of the pool. In this context, what we want then is to minimize the maximum difference between the value of the coalition and the players out of the coalition, as:

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \max_{S \in \rho(\mathcal{N}) \setminus \mathcal{N}} \left\{ \Phi\Big(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})\Big) - \sum_{n \in S} x_n \Phi\Big(R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})\Big) \right\} \right\}. \tag{4-11}$$

From a computationally perspective, equation (4-11) can be conveniently re-written as follows:

$$\min_{x_n, \epsilon} \;\; \epsilon \tag{4-12}$$

Subject to:

$$x_n \geq 0, \qquad \forall\; n \in \mathcal{N}; \tag{4-13}$$

$$\sum_{n \in \mathcal{N}} x_n = 1; \tag{4-14}$$

$$\epsilon \geq \Phi\Big(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})\Big) - \sum_{n \in S} x_n \Phi\Big(R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})\Big), \quad \forall\; S \in \rho(\mathcal{N}) \setminus \mathcal{N}. \tag{4-15}$$

As each coalition is represented in the last restriction of model (4-12)–(4-15), there will always be a difference between each sides of that equation, as already stated. For example: for 3 players, there will be 7 combinations of them, meaning that there will be 7 equations for each of those sub coalitions, and consequentially 7 values for such difference.

There is only one value of ϵ for each analysis, but since every coalition has a numerical difference between being in or out of the pool, we name such difference as ϵ_S. For a given allocation to be in the nucleolus of the game, every ϵ_S must have a value of, at maximum, zero.

Restrictions (4-13) and (4-14) guarantee that the players in the pool do not participate in short-sale and that they make up for all the pool.

We highlight that, in (4-15)–(4-12), ϵ recovers the coalition with the highest difference between being in and out of the grand coalition, thus recovering the most critical coalition with respect to core stability. Furthermore, if it is negative, than the optimal solution identified ($\mathbf{x}^*$) belongs to the core, i.e., $\mathbf{x}^* \in \mathcal{X}$. Otherwise, the core is empty, thus no allocation can be found such that being in the pool is advantageous with respect to every possible coalition $S \in \rho(\mathcal{N}) \setminus \mathcal{N}$.

Aiming to analyze the influences of optimal allocations in this work, both sides of this restriction are going to be calculated separately, and hereby be called $\mathcal{P}(\Phi)$ and $\mathcal{O}(\Phi)$. Their difference is equal to ϵ_S. $\mathcal{O}(\Phi)$ measures the value a coalition S has when it is not mining in the pool, while $\mathcal{P}(\Phi)$ measures the value S has when joining the pool.

$$\mathcal{O}(\Phi) = \Phi\Big(R_S(\tilde{\pi}_t, \tilde{\gamma}_{t,S})\Big). \tag{4-16}$$

$$\mathcal{P}(\Phi) = \Phi\left(\sum_{n \in S} x_n R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})\right). \tag{4-17}$$

With all optimal allocations x_n obtained by the results of the optimization model, those can be used to indicate the incomes for every player in the pool:

$$\mathcal{R}\Big(x_n, R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})\Big) = x_n \times R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}}), \quad \forall\, n \in \mathcal{N}. \tag{4-18}$$

.

Note that $\mathcal{R}\Big(x_n, R_{\mathcal{N}}(\tilde{\pi}_t, \tilde{\gamma}_{t,\mathcal{N}})\Big)$ is the income that a given player $n \in \mathcal{N}$ obtains by participating in the pool, with a quota of x_n.

5
Results and Discussions

Since the theory behind Bitcoin mining through its blockchain networks is discussed, and the mathematical models, as well as the concept of the nucleolus in game theory are already presented in previous chapters, we now study the applications of such knowledge. First, we analyse an illustrative example with 3 players, for 3 different time periods (or number of blocks ahead in the ledger), obtaining the optimal allocations for a range of λ from 0.0 to 1.0. The periods chosen for analysis are T=1, 6 and 144 (representing the next 10 minutes, 1 hour and 24 hours). The results are analyzed from a cumulative perspective, and a visual representation of the core is obtained. An analysis of the value of each side of equation (4-8) on the optimization model (4-12)–(4-15) indicates the reason that, for some values of λ, there is no nucleolus.

With the model validated for the base case, a real case analysis is conducted based on real data from pool miners and bitcoin from January, 2019. The model is tested for a higher number of players, where it is shown that for some coalitions in a pool, joining forces for the next T blocks brings monetary value to the pool.

In both cases (illustrative example and real case), a hypothetical gain in probability when mining in the coalition is proposed and the results are discussed.

5.1
Illustrative Example

The first studied case consists of a pool with 3 miners, and all the players outside the pool are considered as one player (hereby called “Rival”) with the aggregated sum of their hashpowers (which equals the complementary value to the pool’s total hashpower). This case is used to test the model and validate its settings. The profit from mining is obtained following equation (3-5). Also, $\tilde{\pi}_t^v$ is considered to be zero for simplicity, since the fixed reward for mining a block is significantly higher then the average of the variable reward (12.5 BTC against 0.99 BTC), and T to be 1 (single period). Further analysis of $T = 6$ and 144 are conducted in section 5.1.3.

With that, let us consider a bitcoin network where the pool members are the only participants. We initially assume that they have no mining costs, the probability of a coalition $S \in \rho(\mathcal{N})$ to mine a block is additive on the players hashpower (i.e., $f(S) = 0,\ \forall\ S \in \rho(\mathcal{N})$) and the agents are risk neutral ($\lambda = 0.0$). Table 5.1 summarizes the case.

Player	Hashpower (%)	Cost to Mine (BTC)
#1	25	0.0
#2	15	0.0
#3	60	0.0
Rival	0	0.0

Table 5.1: Hashpower of players with no risk measure, no costs and mining alone.

The optimal allocation in this case is described by $x_1 = 0.25$, $x_2 = 0.15$, $x_3 = 0.60$ and $\epsilon = 0.0$, equal to each player hashpower, in line with **Proposition 1**. More precisely, this intuitive result shows a solution to the reward problem by allocating the pool revenues according to the computational power, or the “relative hashpower” (which, in this case, is the very hashpower of each player, since there is no rival at this point). That solution comes from the setup where the players in the pool are the only ones in the network trying to mine the next block. Also, there is no strict benefit from being part of the pool, since $\epsilon = 0.0$. That means that both $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$ have the same value.

Now, let us analyse a similar setup, but considering individual costs to mine different from zero and the total cost within a coalition to be additive in the individual costs. Here, we consider such costs to be in a direct relation to the hashpower of each player (higher hashpowers are likely to pay more for energy since it means a more robust computer setup), as it is expected in reality. Though differences in the cost of energy itself (for instance, for different countries) is not yet considered. Table 5.2 summarize the case.

Player	Hashpower (%)	Cost to Mine (BTC)
#1	25	1.5
#2	15	0.2
#3	60	2.5
Rival	0	0.0

Table 5.2: Hashpower of players with no risk measure, with costs to mine, and mining alone.

The optimal allocations in this case are $x_1 = 0.196$, $x_2 = 0.202$ and $x_3 = 0.602$. Once more, the optimization model returns $\epsilon = 0.0$. This result shows that adding costs to the players shifts the solution from the intuitive allocation (rewarding through the relative power without any rival). Player 1, for instance, goes from an allocation of 0.25 to 0.196. When analysing the value of $\mathcal{O}(\Phi)$ for Player 1, without costs it is 3.125, whereas with costs, it is 1.625. For Players 2 and 3, that value is respectively 1.875 and 1.675, and 7.5 and 5.0. This shows that, even alone, a player has his value in mining lowered by adding costs (as expected). Also, since λ is zero, $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$ are only being influenced by the expected value (no risk profile has being taken into account for this value of λ). Since costs are presented, the average revenue for each player is lowered, and player 1 is "being punished" by the pool for having a high cost. Player 3, although with the highest cost to mine has also the highest hashpower, thus bringing a greater value to the pool, and maintaining his reward. The nucleolus of the game is deviated from the hashpower distribution by the presence of the mining costs.

The next case includes risk aversion profile and also the rivals. The case setup is presented in Table 5.3:

Player	Hashpower (%)	Cost to Mine (BTC)
#1	25	1.5
#2	15	0.2
#3	35	2.5
Rival	25	2.0

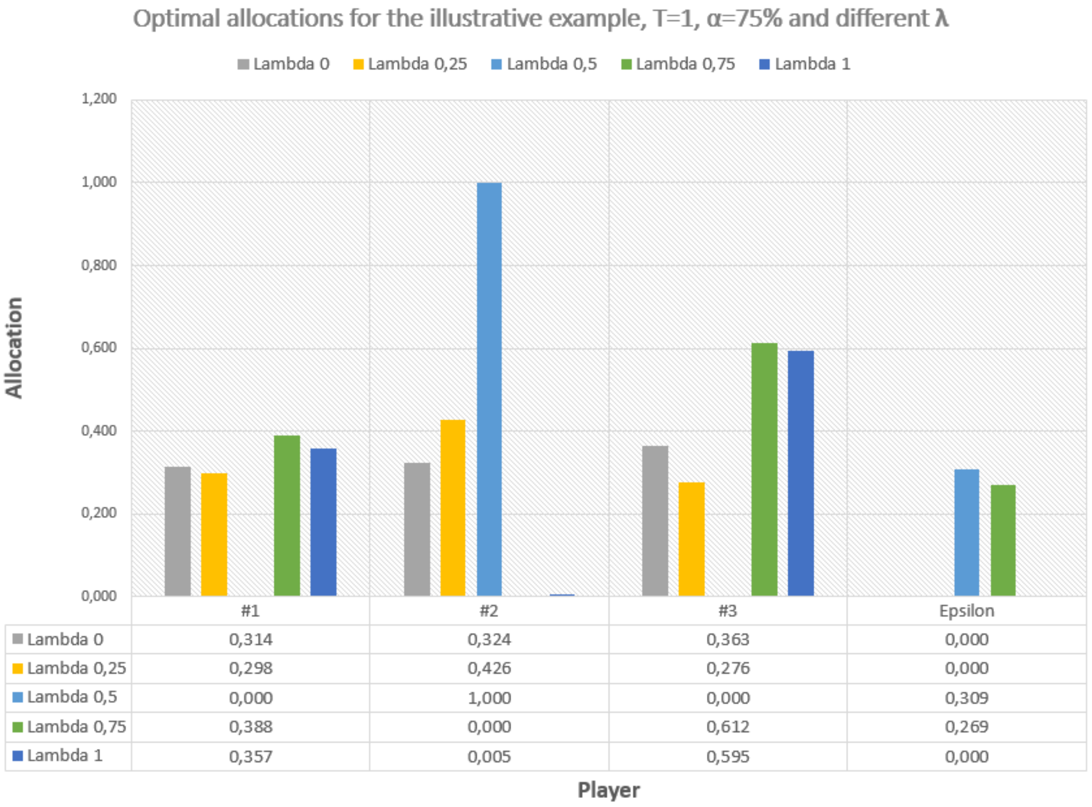
Table 5.3: Hashpower and cost of players in illustrative example.

For the setting above, the optimization model (4-12)–(4-15) returns the following allocation results for different levels of risk aversion:

Player	λ				
	0.0	0.25	0.50	0.75	1.0
#1	0.314	0.298	0.000	0.388	0.357
#2	0.324	0.426	1.000	0.000	0.047
#3	0.363	0.276	0.000	0.612	0.595
ϵ	0.000	0.000	0.309	0.269	0.00

Table 5.4: Optimal allocations for the illustrative example, $T = 1$, α=75% and different levels of risk aversion (λ).

Results in Table 5.4 differ from the hashpower distribution, showing that the presence of the risk measure and costs associated with mining drive the solution away from an intuitive allocation. We highlight that, for $\lambda \in \{0.50, 0.75\}$, the optimization model did not identify an allocation with $\epsilon \leq 0.0$. That means that mining in the pool is actually financially worst for those two risk profiles.



	#1	#2	#3	Epsilon
■ Lambda 0	0,314	0,324	0,363	0,000
■ Lambda 0,25	0,298	0,426	0,276	0,000
■ Lambda 0,5	0,000	1,000	0,000	0,309
■ Lambda 0,75	0,388	0,000	0,612	0,269
■ Lambda 1	0,357	0,005	0,595	0,000

Figure 5.1: Column bars chart for the optimal allocations for the illustrative example, $T = 1$, α=75% and different levels of risk aversion (λ).

Figure 5.1 shows the same result as table 5.4, but represented visually in a series of column charts. For each player, the evolution in the allocation x_n is represented for $\lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. In the last column, the evolution of ϵ is presented.

In Table 5.5, values of $\mathcal{O}(\Phi)$, $\mathcal{P}(\Phi)$ and $\{\epsilon_S\}_{S \in \rho(\mathcal{N})}$ obtained for $\lambda \in \{0.50, 0.75\}$ are shown.

Player	λ	$\mathcal{O}(\Phi)$	$\mathcal{P}(\Phi)$	ϵ_S	λ	$\mathcal{O}(\Phi)$	$\mathcal{P}(\Phi)$	ϵ_S
1	0.50	0.062	0.000	0.062	0.75	-0.719	-0.720	0.001
2	0.50	0.738	0.489	0.249	0.75	0.269	0.000	0.269
1, 2	0.50	0.799	0.489	0.309	0.75	-0.451	-0.720	0.269
3	0.50	-0.312	0.000	-0.312	0.75	-1.406	-1.314	-0.272
1, 3	0.50	-0.249	0.000	-0.249	0.75	-2.125	-1.854	-0.271
2, 3	0.50	0.422	0.489	0.067	0.75	-1.139	-1.134	-0.005
1,2 and 3	0.50	0.489	0.489	0.000	0.75	-1.854	-1.854	0.000

Table 5.5: $\mathcal{O}(\Phi)$, $\mathcal{P}(\Phi)$ and ϵ_s for the illustrative example, λ 0.5, 0.75 and 1.0.

Table 5.5 shows the reason why the optimization model indicates a result that is not in the nucleolus of the game for $\lambda \in \{0.50, 0.75\}$: the highest value of ϵ_s is not zero for each of those cases. It shows that for all coalitions in such λ's, there is at least a subgroup of players where the value of mining outside the pool is higher than mining in the pool. Also, the risk aversion is at a high level such that $\mathcal{P}(\Phi)$ for pool (coalition representing all the 3 players) is negative for $\lambda = 0.75$. If playing together makes players #1, #2 and #3 lose money, there is no sharing method that will make it worth (unless $\mathcal{O}(\Phi)$ was even worse). We highlight that $\lambda = 1.0$ is an extreme case. All values of $\{\epsilon_S\}_{S \in \rho(\mathcal{N})}$ are equal to zero, as both $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$ are negative for all coalitions and the same. More precisely, here, the risk profile reaches a point where no monetary gain is computed by the measure of value. For every other λ, ϵ is zero, and so all $\{\epsilon_S\}_{S \in \rho(\mathcal{N})}$.

Not all allocation sharing belongs to the nucleolus of the game for our model. Based on the setup from table 5.3, a common and intuitive type of sharing in which all players equally receive the same allocation ($x_n = 33.33\%$) is analyzed, with an ϵ value of 0.575. This shows that, for this allocation policy, splitting the rewards equally for all three players is not inside the nucleolus of the game. Again for based on the setup from table 5.3, another type of sharing not in the nucleolus of the game is the one where players are rewarded by their relative hashpower, or: $x_n = \frac{hp_n}{\sum_{n \in \mathcal{N}} hp_n}$. The value of ϵ for this case is of 0.64, showing that players are rewarded more if they are not mining together in the pool, for this allocation.

5.1.1 Single Period Analysis

The allocations for $T = 1$ in the illustrative example have been obtained for the different setups proposed, and it is of interest to analyze the inverse accumulated probability for each player within the pool and by its own.

Based on the setup from table 5.3, and using a Monte Carlo simulation with 100,000,00 scenarios (only to be able to obtain the cumulative distribution), we obtain the next results:

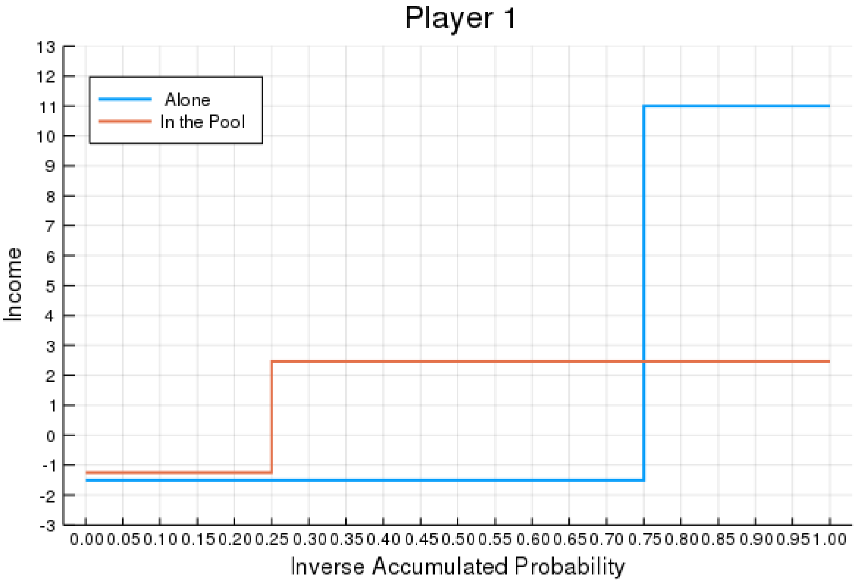


Figure 5.2: Accumulated Probability for Player 1, illustrative example, $T = 1$, $\alpha = 75\%$ and $\lambda = 0.0$.

Figure 5.2 shows an expected result: player 1 mines by itself 25% of the time (as shown with this player having negative cash flows up to 75% of the time). When he mines, his amplitude of gain is 11 BTC, which is equal to 12.5 BTC, the fixed revenue from mining a block, minus 1.5 BTC, the cost he has to mine (and also his value of $CVaR_\alpha$). Player 1 mines alone 25% of the time because that is his probability to mine. Here, with T=1 only, the probability to mine is the expected value of a Bernoulli distribution with probability q_S, which is exactly the probability the player has to mine. During the time he spends not mining a block, his constant negative cash flow has an amplitude of -1.5 BTC.

On the other hand, when mining in the pool, player 1 spends 75% of the time with positive cash flows. This result matches the probability the coalition has to mine. Now, amplitude of player 1 is $x_1(12.5-4.2) = 2.60$ BTC, in which, -4.2 BTC is due to the $CVaR_\alpha$ term. His revenue is lower than that when he succeeds mining alone, but his cash flow is positive most of the time. When having negative cash flows in the coalition, his amplitude is of $-4.2x_1 = -1.32$ BTC. In practice, we argue that a player will likely prefer to have positive cash flows more often, than to receive a higher value rarely.

The behavior observed above can be summarized describing the analytical form for the Inverse Accumulated Probability (IAP):

$$F^{-1}(A) = \begin{cases} -C_n & A < 1 - q_n = 1 - hp_n \\ \pi^f - C_n & A \geq 1 - q_n = 1 - hp_n \end{cases}$$

$$F^{-1}(P) = \begin{cases} -x_n * (C_{\mathcal{N}}) & P < 1 - q_{\mathcal{N}} = 1 - \sum_{n \in \mathcal{N}} hp_n \\ x_n * (\pi^f - C_{\mathcal{N}}) & P \geq 1 - q_{\mathcal{N}} = 1 - \sum_{n \in \mathcal{N}} hp_n \end{cases}$$

The same behavior in the IAP can be observed for players 2 and 3:

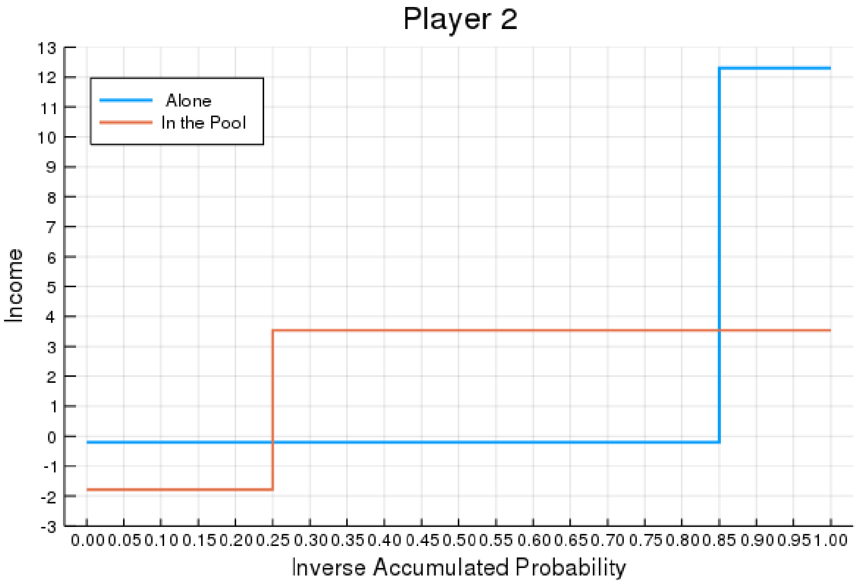


Figure 5.3: Accumulated Probability for Player 2, illustrative example, $T = 1$ and $\lambda = 0.0$.

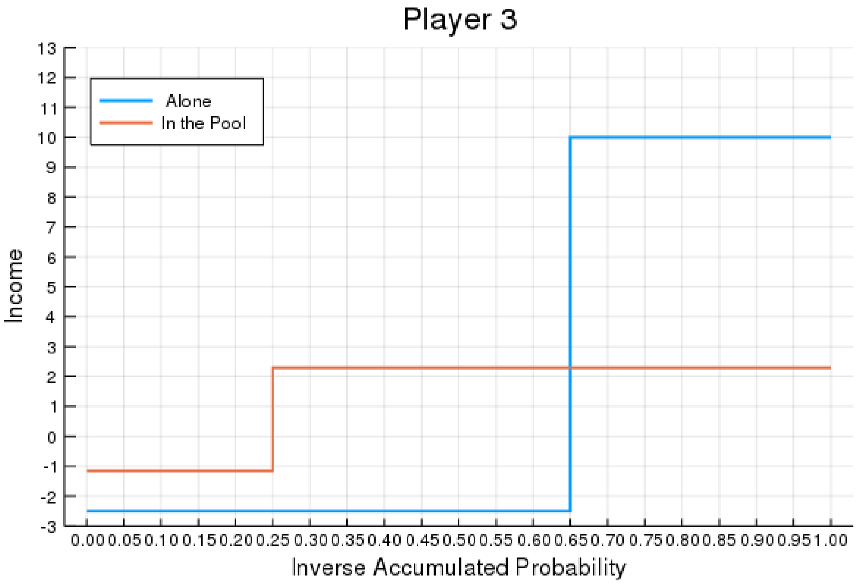


Figure 5.4: Accumulated Probability for Player 3, illustrative example, $T = 1$ and $\lambda = 0.0$.

The differences of player 2 (Figure 5.3) and player 3 (Figure 5.4) with respect to player 1 lie in the amplitude of gains (when in the pool). They share the rewards according to the allocations obtained in the optimization model, but start shifting the signal in the cash flow at the same time (having positive gains roughly 75% of the time). The amplitude of gains for mining alone are similar, though the mining costs differ for each player, and their time without positive cash flows are different from one another, since their hashpower are different.

The optimization model is now adapted to force allocations to be between 0.0 and 1.0, with steps of 0.001 for each player (x_1, x_2 and x_3), and the value of ϵ is collected as the result of the optimization model. For players 1 and 3, Figure 5.5 shows the results where $\epsilon \leq 0$:

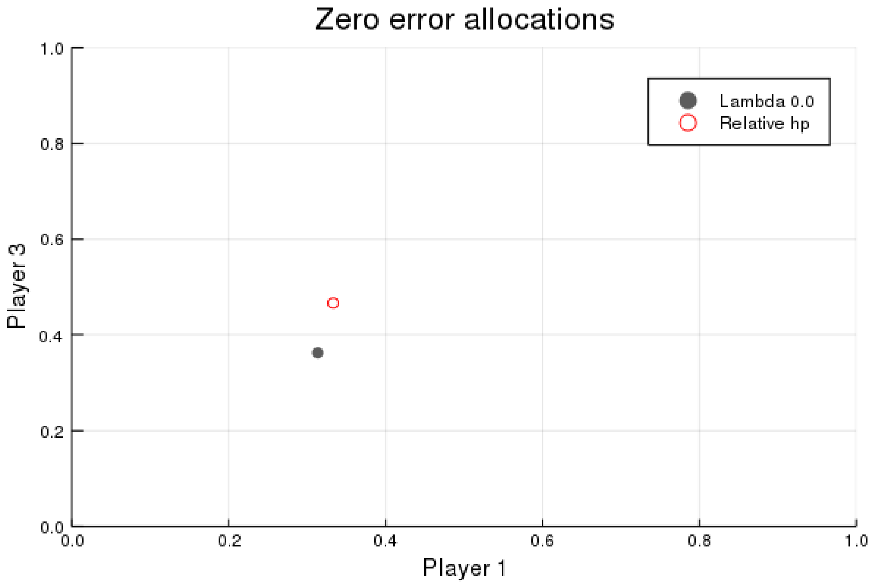


Figure 5.5: Allocations inside the core for players 1 and 3, illustrative example, $T = 1$ and $\lambda = 0.0$.

The result above shows that, for the base case with $T = 1$ and $\lambda = 0$, the allocations that maintain stability of the pool are the ones found by the optimization model, and there is only a single allocation that returns $\epsilon \leq 0$. Worth noting that the relative hashpower allocation is shown in the Figure 5.5, but does not represent an allocation in the nucleolus. Figure 5.5 is the visual representation of the core, or nucleolus of the game, as the allocations in the figure are the ones that maintain the pool stable. The "Relative hp" point is shown just as a comparison of this sharing method. The same result is obtained for $\lambda \in \{0.25, 0.5, 0.75, 0.99\}$ and presented next.

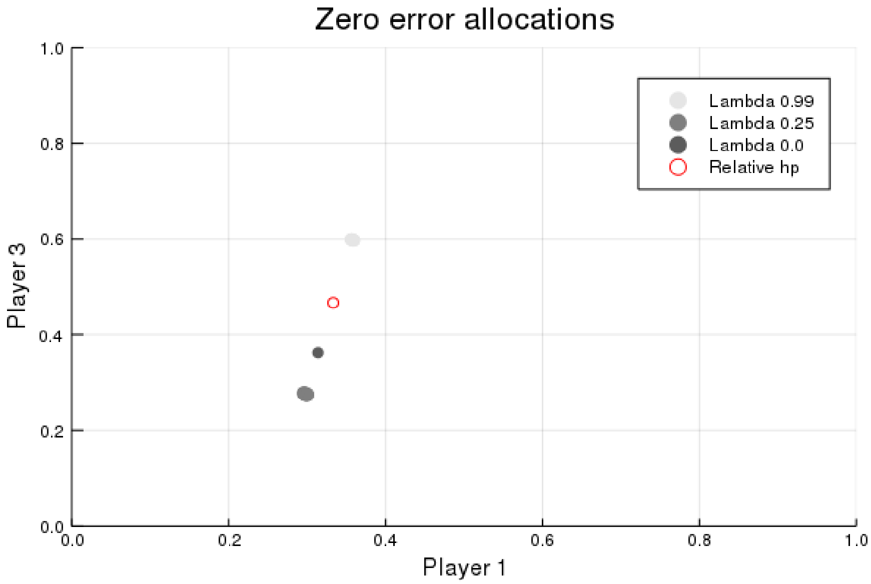


Figure 5.6: Allocations inside the core for players 1 and 3, illustrative example, $T = 1$, $\alpha = 0.25$ and different $\lambda \in \{0, 0.25, 0.50, 0.75, 1.0\}$.

Figure 5.6 shows that $\lambda = 0.5$ and 0.75 do not return solutions with ϵ equal or lower than zero (as expected by the results shown in table 5.4), but also shows that the ones that return ($\lambda \in \{0.0, 0.25, 0.99\}$) show a single allocation as the core. More specifically, taking part in the pool for $T = 1$ (or the next 10 minutes) returns the same values for $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$. The total reward obtained by the pool is the same of the sum of the player's individual reward if mining alone, just distributed according to the allocations proposed by the optimization model and distributed at different time.

The results above validate our model for $T = 1$ and $\lambda \in \{0.0, 0.25, 0.5, 0.75, 0.99\}$, with the expected value from revenues and costs obtained, along with the probabilities matching analytically with the sum of hashpower for any given S. Now, we propose an analysis for $T > 1$.

5.1.2 Gain in Probability

Equation (3-2) proposed in Section 3.1 of this work, indicates that the probability of a coalition $S \in \rho(\mathcal{N})$ to mine might not be only the sum of the computational power of each player, but also a result of a function $f(S)$ that may increase this probability due to the gathering of players in the pool. Considering said increase occurring in the form of:

$$q_S = \sum_{n \in S} hp_n + (|S| - 1) \times \beta, \tag{5-1}$$

The term $|S|$ refers to the number of players in the coalition $S \in \rho(\mathcal{N})$. For example, if there is only one player, there is no gain in the probability to mine, which becomes the hashpower of that player. If there are 2 (two) players in the coalition, there is a term β added to q_S. If there are 3 (three) players in the coalition, there is a term $2 \times \beta$ added to q_S, and so on. A beta of 0.01 indicates an increase of 1% per player in the pool. For different increments on mining probability $\beta \in \{0.01, 0.02, 0.05, 0.07, 0.1\}$ and a risk-profile level $\lambda = 0.25$ and based on the setup from table 5.3, the obtained allocations are presented in Table 5.6:

	β				
Player	0.00	0.02	0.05	0.07	0.10
x_1	0.298	0.306	0.313	0.316	0.319
x_2	0.426	0.404	0.386	0.378	0.370
x_3	0.276	0.289	0.301	0.306	0.311
ϵ	0.0	-0.289	-0.729	-1.02	-1.46

Table 5.6: Optimal allocations for the illustrative example for $\lambda = 0.25$, $T = 1$, $\alpha = 75\%$ and different increments on mining probability $\beta \in \{0.01, 0.02, 0.05, 0.07, 0.1\}$.

The main result from this setup is the value of ϵ being negative as β increases. Since it becomes each time more negative, it shows that raising the probability to more than the sum of hashpowers induces value for the pool that was not present before. For all coalitions, $\mathcal{P}(\Phi)$ is higher than $\mathcal{O}(\Phi)$. Another approach to measure such influence is to obtain the average revenue (as shown in equation (4-9)):

β	0.0	0.02	0.05	0.07	0.1
$\mathbf{E}[R_S(\tilde{\pi}_t^S, \tilde{\gamma}_{t,S})]$	5.17	5.42	5.67	6.42	7.67

Table 5.7: Average Income of S, per different β.

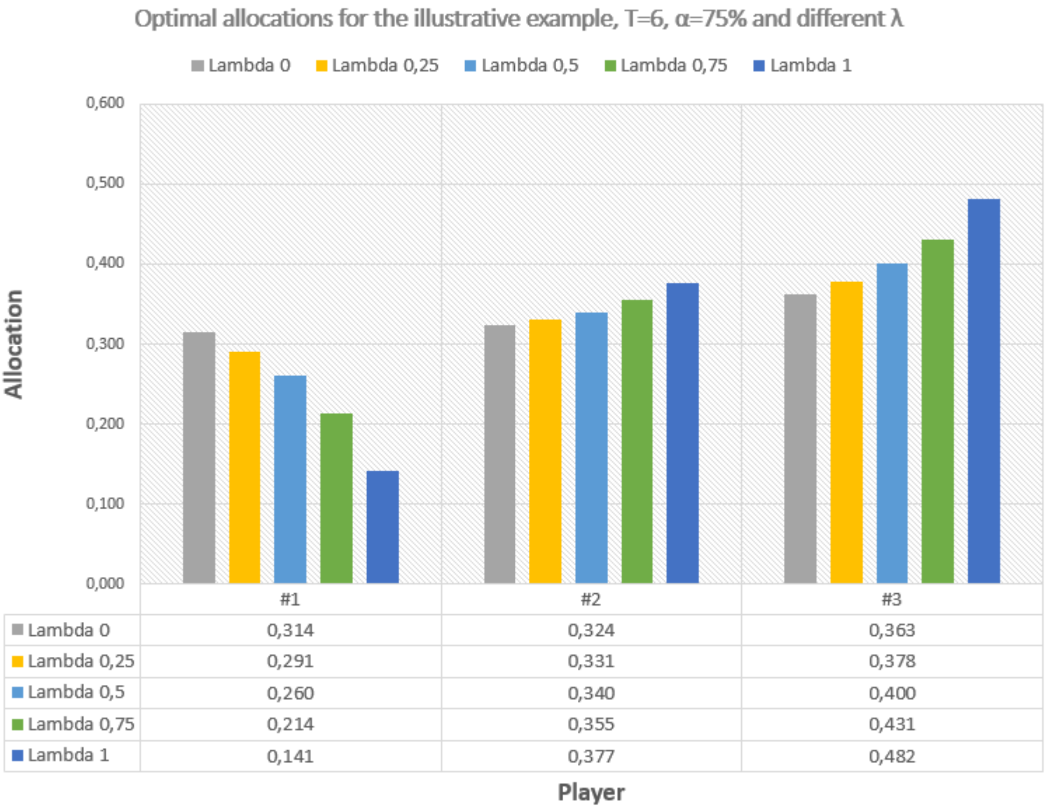
Table 5.7 shows that $\mathbf{E}[R_S(\tilde{\pi}_t^S, \tilde{\gamma}_{t,S})]$ raises with β. This is likely a result of the raise in q_s caused by adding β. So, the coalition S mines more blocks on average, thus receiving higher rewards.

5.1.3 Multi period analysis

With the illustrative case considering T=1, we now impose that T is 6 (and $\beta = 0.0$), which means the next 6 blocks (or 1 hour) are considered in the analysis, transforming the probability to mine from a Bernoulli to a Binomial distribution, as discussed in section 3.1. The optimal allocations obtained from the model, based on the setup from table 5.3 are now shown in table 5.8 and figure 5.7:

	λ				
	0.0	0.25	0.5	0.75	1.0
#1	0.314	0.291	0.260	0.214	0.141
#2	0.324	0.331	0.340	0.355	0.377
#3	0.363	0.378	0.400	0.431	0.482
ϵ	0.000	-1.64	-3.28	-4.92	-6.56

Table 5.8: Optimal allocations for the illustrative example, $T = 6$, α=75% and different λ.



	#1	#2	#3
■ Lambda 0	0,314	0,324	0,363
■ Lambda 0,25	0,291	0,331	0,378
■ Lambda 0,5	0,260	0,340	0,400
■ Lambda 0,75	0,214	0,355	0,431
■ Lambda 1	0,141	0,377	0,482

Figure 5.7: Column bars chart for the optimal allocations for the illustrative example, $T = 6$, α=75% and different levels of risk aversion (λ).

For $\lambda = 0$, in line with **Proposition 1**, the optimal allocations are the same as for T=1. Any λ different from 0.0 now differs from the same case as for T=1. This shows that, other than the introduction of risk measure tools, the multi period analysis also has great influence over the results of the optimization model. Aside from $\lambda = 0.0$, all the others show ϵ lower than zero, and raising in value with higher risk profiles. That result differs from the one in Table 5.4, showing that a multi-period setup changes the pool rewards, bringing value to the players who are part of the coalition, what did not happen for T=1 (in fact, for 2 values of λ, the pool had lower value than mining alone).

The inverse accumulated probability, with the same setup as T=1 and λ 0.0, for player 1, is presented in Figure 5.8.

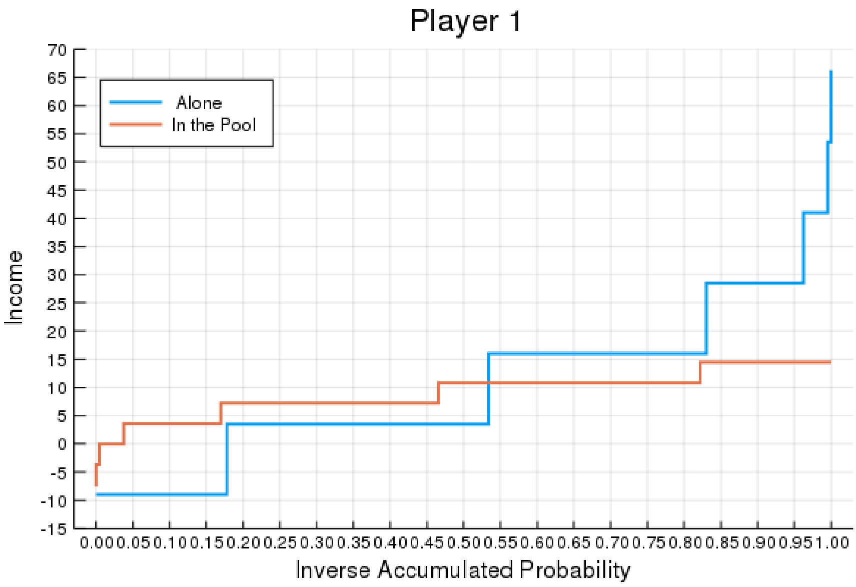


Figure 5.8: Accumulated Probability for Player 1, illustrative example, $T = 6$, $\alpha = 0.25$ and $\lambda = 0.0$.

Note the existence of a set of 6 “jumps” in the graph for each series. Those correspond to probabilities that player 1 has to mine once in the 6 next blocks, to mine twice in the next 6 blocks up to mine all next 6 blocks. The probabilities for each “jump” are defined by a Binomial distribution with q_S. For example, the chance to mine at least once (alone) is 82.20%, which matches with what is seen in the graph (the cash flow becomes positive at 1-82.20, or 17.80%). The rationale is analog to mining twice and up to six times. The probability to mine at least once in the 6 next blocks in a coalition is 99.98% (the cash flow becomes positive at 1-99.98, or 0.02%). The results are similar for players 2 (Figure 5.9) and 3 (Figure 5.10).

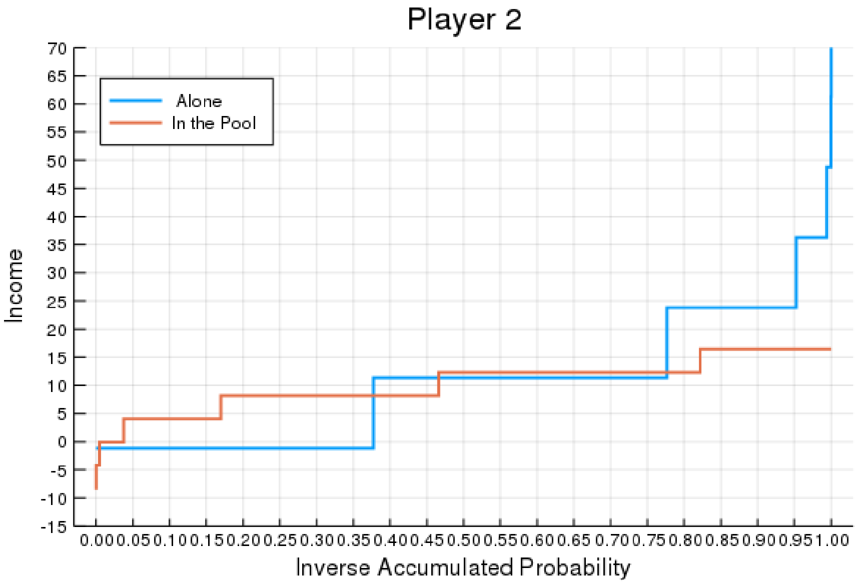


Figure 5.9: Accumulated Probability for Player 2, illustrative example, $T = 6$, $\alpha = 0.25$ and $\lambda = 0.0$.

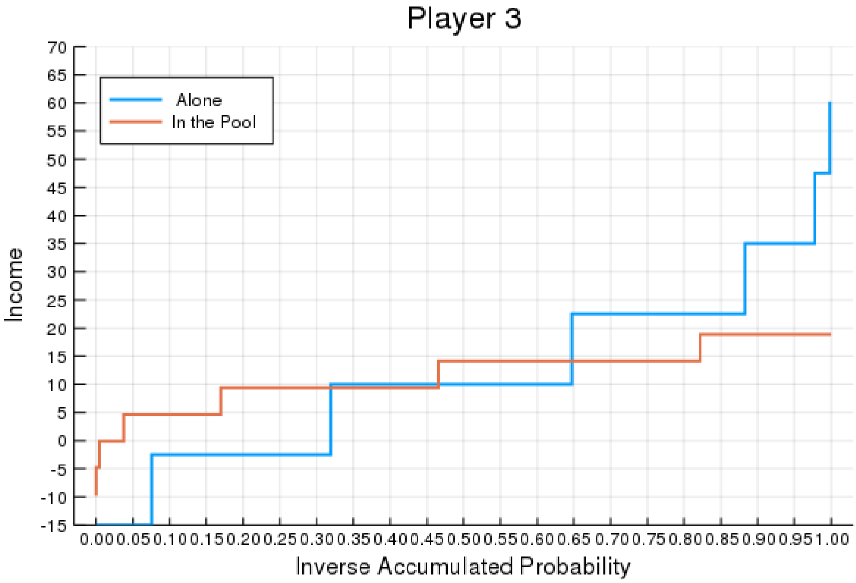
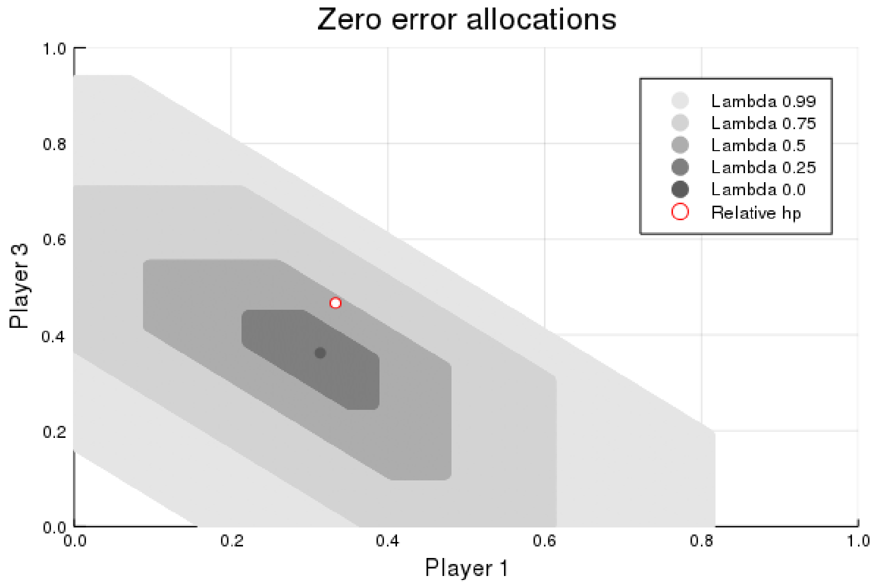


Figure 5.10: Accumulated Probability for Player 3, illustrative example, $T = 6$, $\alpha = 0.25$ and $\lambda = 0.0$.

The results show that, when considering the next T=6 blocks, instead of the next T=1 block, a player has a higher chance to mine at least once when considering more than a block ahead. This means a gain is obtained when looking at the multi-period case. There is another gain that can be visually observable when analysing the results for different λ in the visual representation of the core:



:Core

Figure 5.11: Allocations inside the core for players 1 and 3, illustrative example, $T = 6$, $\alpha = 75\%$ and different λ.

Raising λ brings more different allocations to the nucleolus of the game. When raising λ, the tail of the (1-α) worst cases also increases, and that puts more weight to the risk measure tool $CVaR_{\alpha}$, producing results where players tend to aggregate towards a common objective to avoid losing their money. This was not presented in T=1, since now the analysis indicates that the combinatorial nature of the multi period game brings more value to the pool so that, for the next T blocks, it is better for the players to join forces in a pool. Also, a λ close to 1.0 means a risk profile that only takes into consideration the losses. Nevertheless, economic agents usually mix both parameters (return and risk) in a close to 50% weight for each. Also worth noting that the point of Relative hp is not in the nucleolus of the game for λ 0.0 and 0.25, further reinforcing the need of this study, since sharing by computation power, which would be intuitive to do, does not always keep the coalition stable.

The same results can be seen for combinations of players 1 & 2 and 2 & 3:

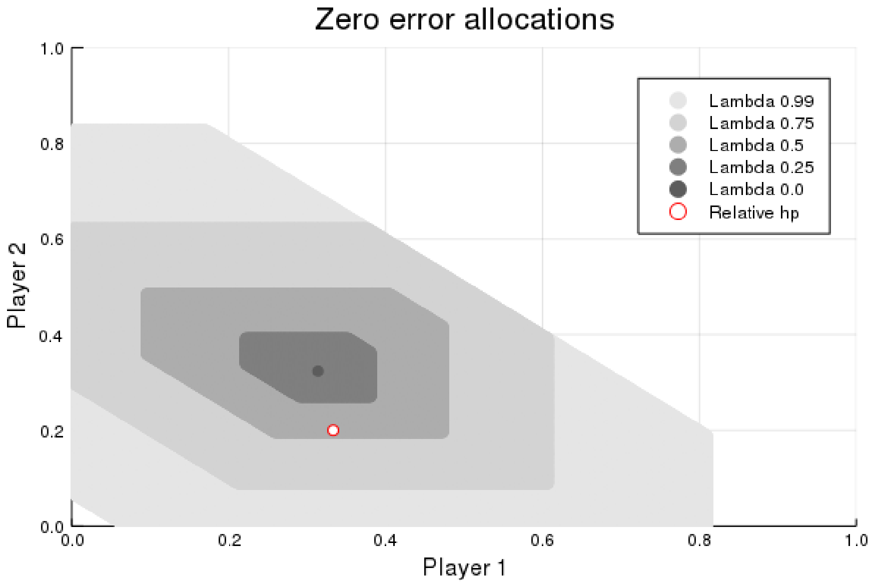


Figure 5.12: Allocations inside the core for players 1 and 2, illustrative example, $T = 6$, $\alpha = 75\%$ and different λ.

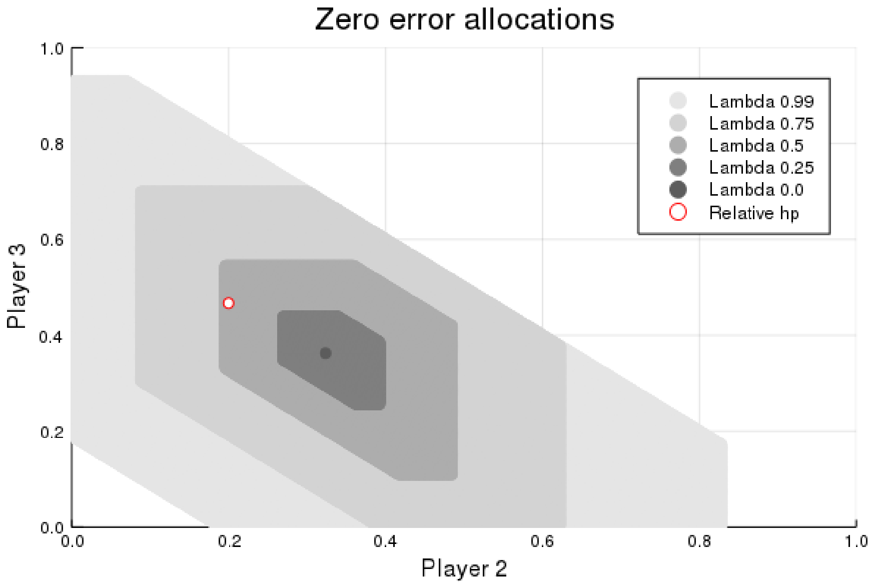


Figure 5.13: Allocations inside the core for players 2 and 3, base case, $T = 6$, $\alpha = 75\%$ and different λ.

The cloud presented in the figures above does not differentiate ϵ_S in terms of its value. For instance, if we want to highlight the allocations that contain the minimal value of ϵ for $\lambda = 0.5$, the result is:

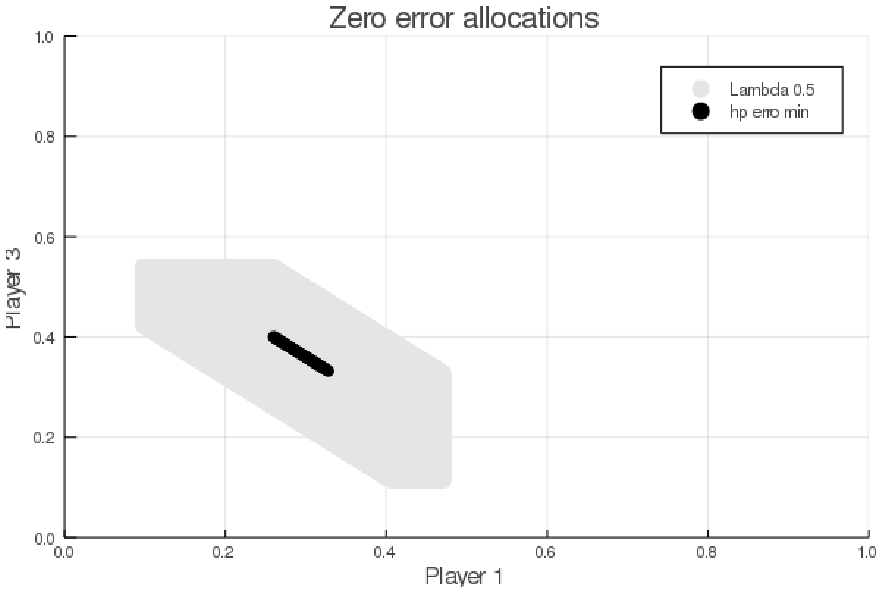


Figure 5.14: Allocations inside the core for players 1 and 3, illustrative example, $T = 6$, $\alpha = 75\%$, $\lambda = 0.5$ and allocations with minimal ϵ.

Figure 5.14 indicates that the allocations with lower value of ϵ lie at the center of the polygon for $\lambda = 0.5$. For the other values of λ studied in this work, the results are:

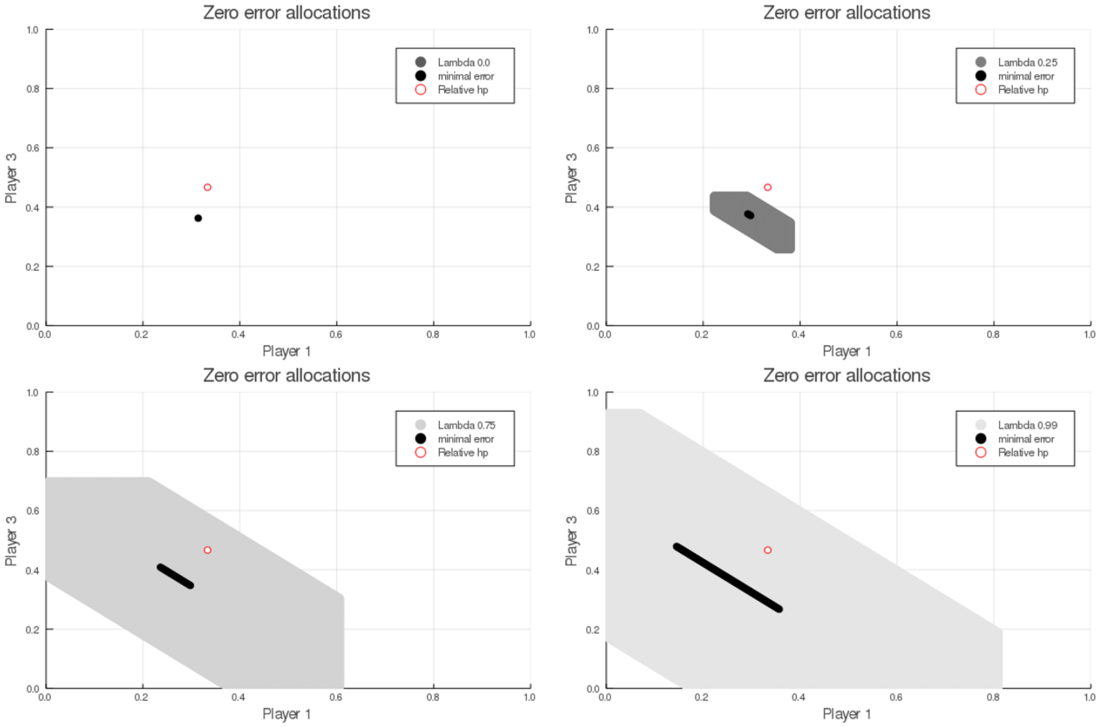


Figure 5.15: Allocations inside the core for players 1 and 3, illustrative example, $T = 6$, $\alpha = 75\%$, different λ and allocations with minimal ϵ.

For each of the cases above, figure 5.15 shows that the allocations with the lowest ϵ are contained at the center of the cloud of allocations for each λ..

We now analyse how the same figures of the nucleolus behave for β different of zero ($\beta = 0.1$):

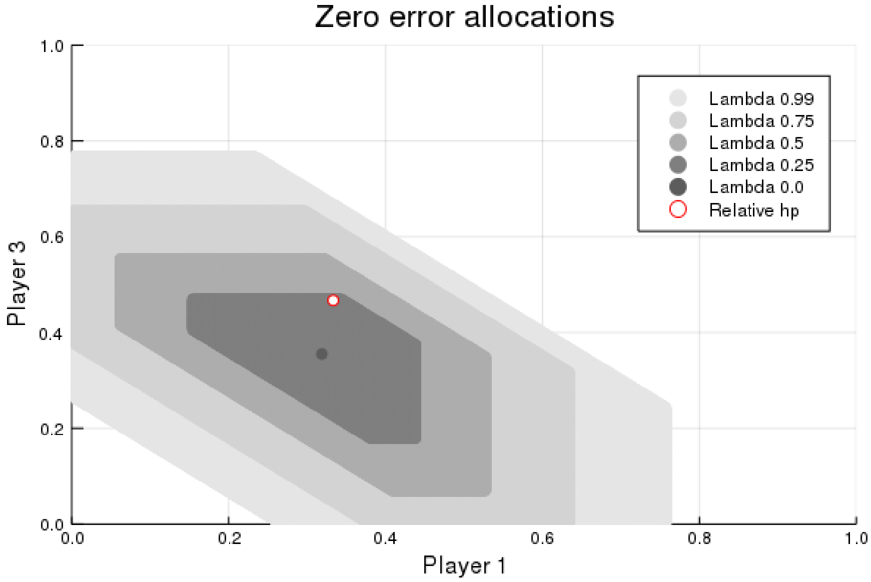


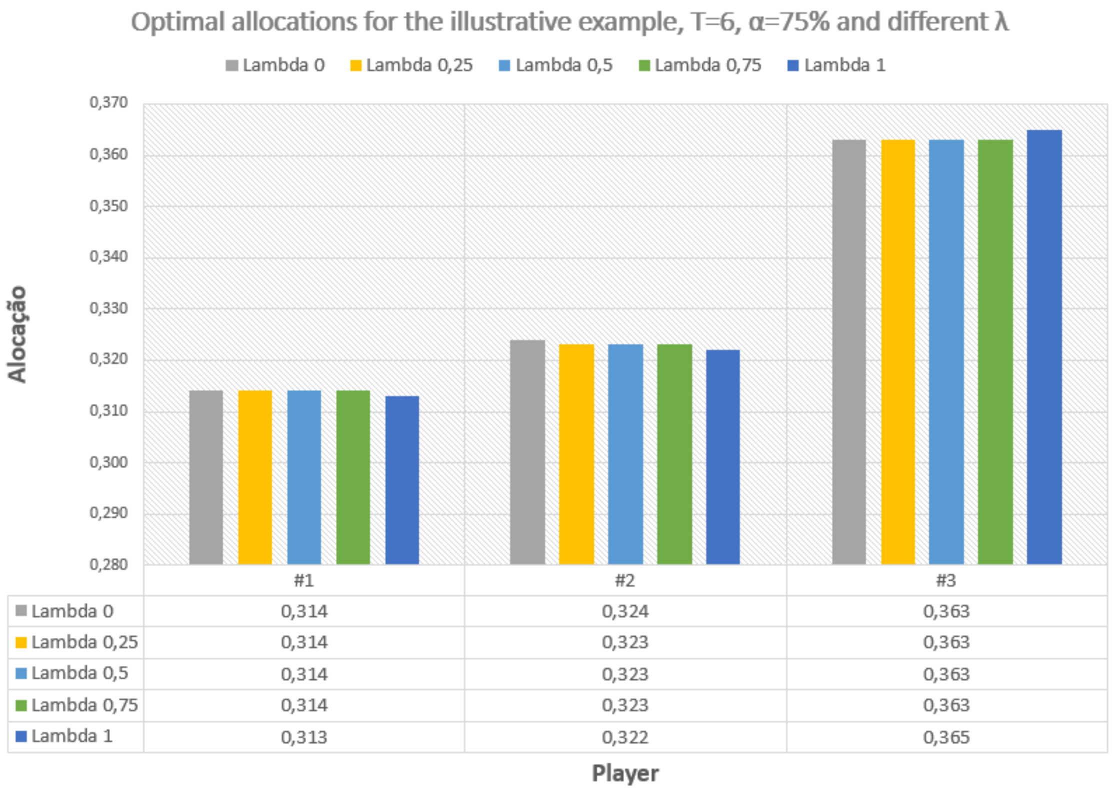
Figure 5.16: Allocations inside the core for players 1 and 3, illustrative example, $T = 6$, $\alpha = 75\%$, different λ and β=0.1.

Figure 5.16 shows that raising β shrinks the area of the nucleolus for higher λ, and the opposite for lower λ. This is likely due to the adjustment that β provides; it is now not necessary to go through higher risks to obtain higher values from joining the pool.

For $T = 144$ (24 hours, or a day), results are similar to $T = 6$, also indicating the gain when joining a pool looking at multi period mining. For the sake of simplicity, only the inverse accumulated probabilities figure for player 1 and the nucleolus set representation for players 1 and 2 are presented, as well as the allocations identified by the optimization model and the results shown in table 5.9 and figure 5.17:

	λ				
	0.0	0.25	0.5	0.75	1.0
#1	0.314	0.314	0.314	0.314	0.313
#2	0.323	0.323	0.323	0.323	0.322
#3	0.362	0.363	0.363	0.363	0.365
ϵ	0.00	-9.58	-19.2	-28.7	-38.3

Table 5.9: Optimal allocations for the illustrative example, $T = 144$, α=75% and different λ.



	#1	#2	#3
Lambda 0	0,314	0,324	0,363
Lambda 0,25	0,314	0,323	0,363
Lambda 0,5	0,314	0,323	0,363
Lambda 0,75	0,314	0,323	0,363
Lambda 1	0,313	0,322	0,365

Figure 5.17: Column bars chart for the optimal allocations for the illustrative example, $T = 144$, α=75% and different levels of risk aversion (λ).

What is interesting to note is, when a larger period of time is taken into consideration, allocations do not change with the raise in risk profile, but ϵ does. Taking higher risks still benefits more the coalition, but the sharing quota of each player is unchanged.

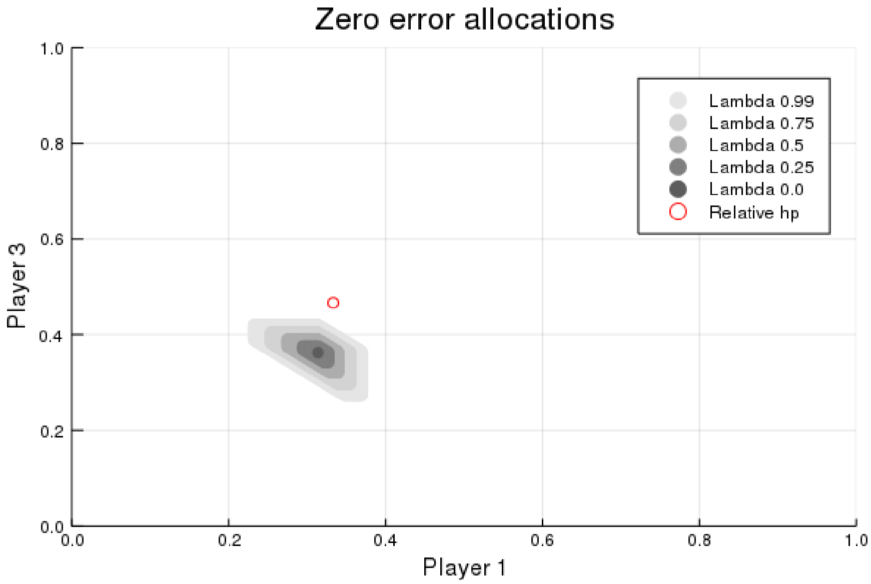


Figure 5.18: Allocations inside the core for players 1 and 3, illustrative example, $T = 144$, $\alpha = 0.25$ and different λ.

An important result arises from Figure 5.18: with the increase in T, the “cloud” of allocations belonging to the nucleolus of each λ decreases, which means that the model may converge to a set of single allocations that keep the players together in the pool (as seen in Table 5.9). The allocation sharing by computational power (the “Relative hp” dot in Figure 5.18) is not in the nucleolus of the game for any λ. It can be concluded that, for a high enough number of blocks ahead, there is no risk profile that embraces the sharing of rewards via intuitive computational sharing as a method in the core of the pool, like it was presented with T up to 6. Greater T brings value to the players due to the combinatorial characteristic of the multi-period setup, but tends to a single reward quota for the players.

5.2
Definitions and motivation: Real Case

Now that the method for obtaining optimal allocations for a pool in the Bitcoin network has been proposed and tested at a small scale, we expand the analysis to consider a more realistic context. Figure 5.19 depicts the pool distribution of the bitcoin network from January 29th, 2019:

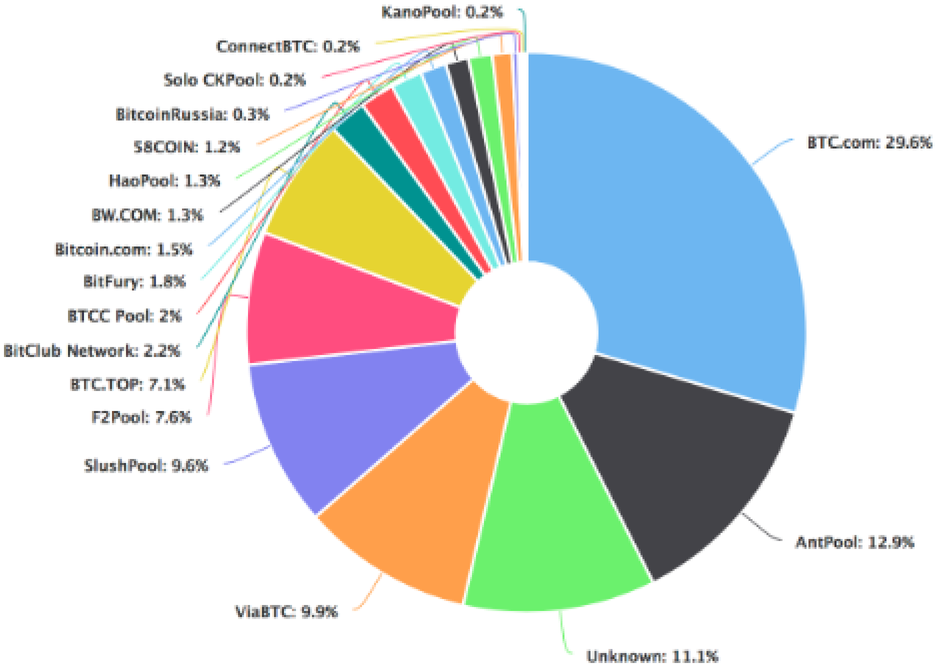


Figure 5.19: Hashpower distribution in the bitcoin network, January 29th, 2019 (BitcoinMiningPools).

KanoPool is a player with 0.2% of the hashrate in the network. On the same day, according to blockchain.info [1], the difficult D of the network was D = 5,814,661,935,891.00, and the total hashrate of the network was h = 39,310,594.00. Using equation (3-1), the time that KanoPool would take to correctly guess a hash is $\tau = 3.17 * 10^{15}$s, or approximately 4 days.

That result does not consider the fact that KanoPool races against other players in the network. Before he guesses his first header' hash, 576 other blocks had been mined during those 4 days, and there is a good chance that someone else finds the correct hash before he mines his expected block. Taking, in theory, another 4 days for him to have another shot with a correct hash.

If KanoPool join forces with all the other players up to BItClub Network, the total hashpower of the group would be of 12,2%, bringing τ to 87 minutes, or approximately 1,45 hours to correctly guess the next hash. This new pool would be much more likely to mine a new block, or at least, would take less time to guess the hash correctly, thus being able to compete with the larger pools.

[1] https://www.blockchain.com/charts/difficulty, accessed may 5th, 2020

5.3 Real case study proposed

This section analyzes the behavior of a pool that KanoPool participates, using the optimization model proposed in this work. A coalition with all the players up to F2Pool is proposed (containing 13 players), and the 14^{th} player, or the rival, being a "fictional" player that contains the cumulative hashpower from all the other players in Figure 5.19. Data obtained from (CryptoCompare), (Wheretomine) (BitcoinMiningPools) and (MiningCostsbyCountry), together with the Dollars to Bitcoin price in January 29th, 2019 (3452,00) help build the case and are demonstrated in Table 5.10:

Player	Country	Hashpower (%)	Cost to mine (U$S)	Cost to Mine (BTC)
KanoPool	United States	0,2	10720,40	0.00621
ConnectBTC	China	0,2	3644,93	0.00211
Solo CKPOOL	Iceland*	0,2	16857,09	0.00977
Bitcoin Russia	Russia	0,3	7032,58	0.00611
58COIN	China	1,2	3644,93	0.00127
HaoPool	China	1,3	3644,93	0.00137
BW.com	Finland**	1,3	16677.76	0.0628
Bitcoin.com	United States	1,5	10720.40	0.0466
Bitfury	Georgia	1,8	6861.05	0.0358
BTCC Pool	Finland	2,0	16677.76	0.096
Bitclub Network	Iceland	2,2	16857.09	0.107
BTC.TOP	China	7,1	3644.93	0.0750
F2Pool	United States	7,6	10720.40	0.236
Rivals	***	73,1	-	-

Table 5.10: Real case parameters.

– * - (CryptoCompare) indicates this player operating in Asia and Europe. Iceland has been chosen to represent it;

– ** - Player with no information available of country of operation. Finland choosen as a representation;

– *** - Rivals costs are not necessary, since we use only his hashpower in the model.

The setup of the network is better observed visually, showing the difference between the hashpower of players and their cost to mine, in figure

5.20. For instance, F2Pool has the highest hashpower of the pool, but also the highest cost to mine (in BTC):

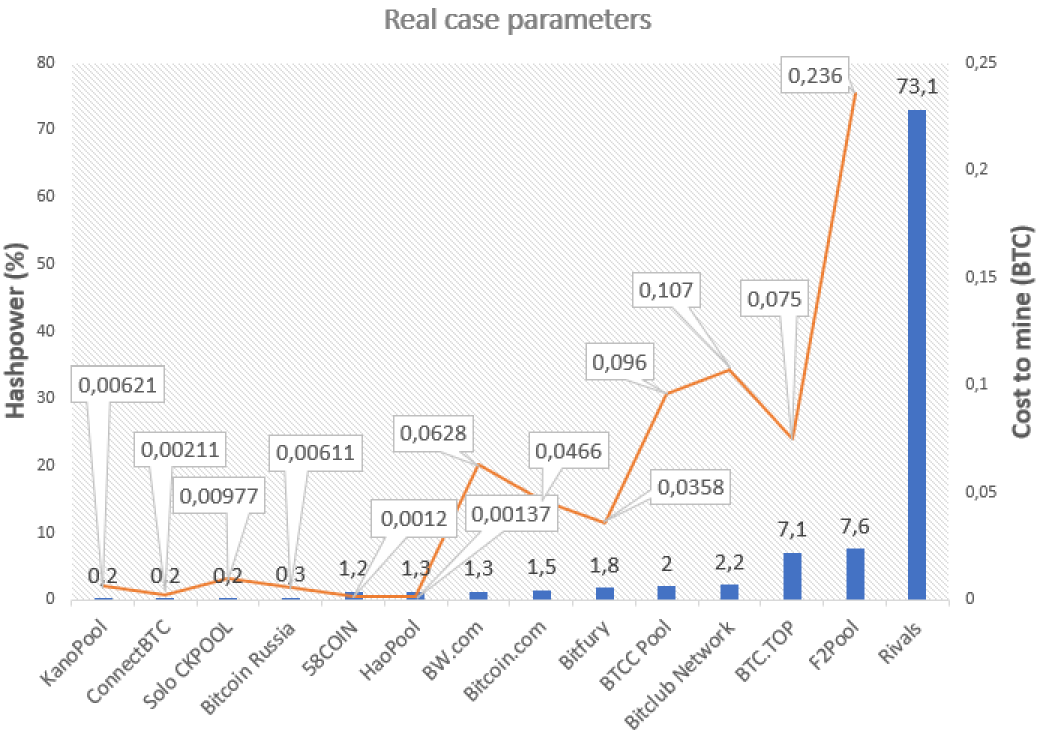


Figure 5.20: Setup of the bitcoin network for the real case. The blue bars are the hashpower of each player, and the orange line is the cost to mine (in BTC).

Reference (MiningCostsbyCountry) indicates costs to mine that are not related to the computational power each player has, or the “amount of machines” one player owns. In order to normalize this, the cost that is given by the study is multiplied by the hashpower of the player in said country, thus producing a result that takes into account his computational power.

It is worth noting that due to computational limitations, a model with all 14 players exceeds the memory needed to obtain results. As a solution to this, players 1 - 4 (KanoPool, ConnectBTC, Solo CKPOOL and Bitcoin Russia) are considered as one, with their hashpowers and costs summed (since they own almost identical hashpower (%) and cost to mine (BTC)).

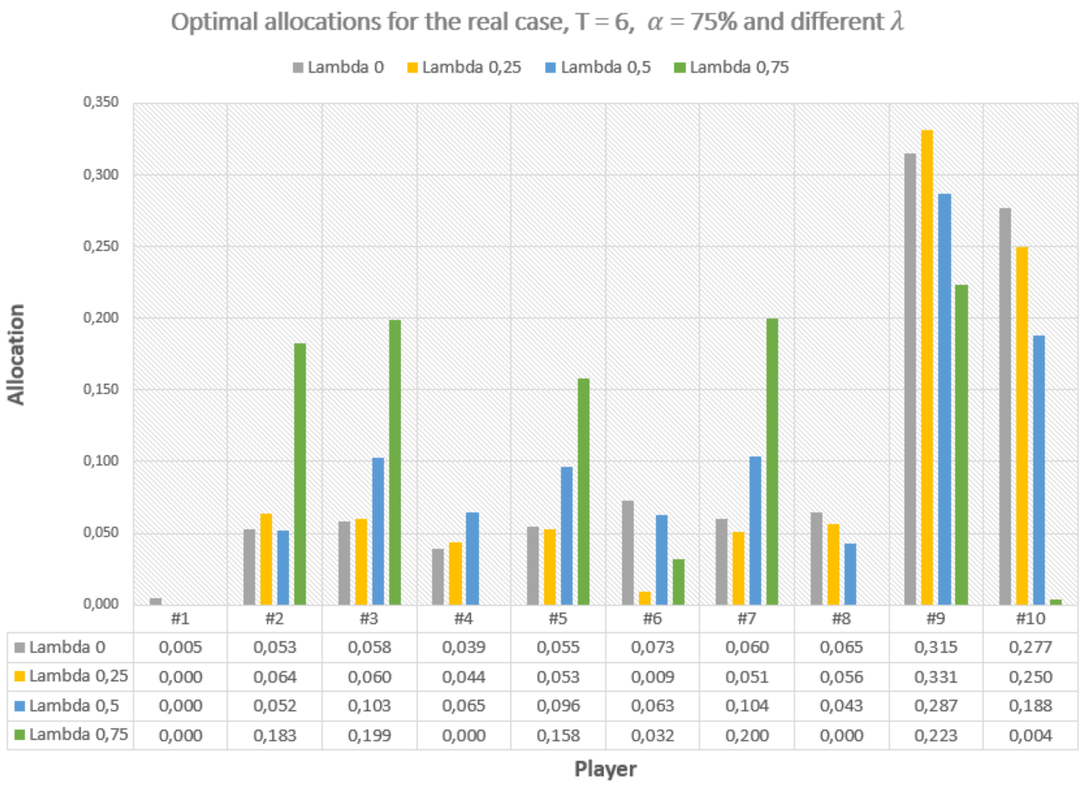
The optimization model returns the results in Table 5.11 for this setup and T=6. The results are similar in structure to those obtained in Section 5.1.3, now with more players to be observed. This time, since we already start with T greater than 1, all different λ (except 1.0) produce results that return ϵ zero or lower, which means that there is a nucleolus in the game set for each case, as expected after the illustrative example analysis.

	λ					
	Pool	0.0	0.25	0.5	0.75	1.0
#1	* Pools 1-4	0.005	0.000	0.000	0.000	0.000
#2	58COIN	0.053	0.064	0.052	0.183	1.000
#3	HaoPool	0.058	0.060	0.103	0.199	0.000
#4	BW.com	0.039	0.044	0.065	0.000	0.000
#5	Bitcoin.com	0.055	0.053	0.096	0.158	0.000
#6	Bitfury	0.073	0.090	0.063	0.032	0.000
#7	BTCC Pool	0.060	0.051	0.104	0.200	0.000
#8	Bitclub Network	0.065	0.056	0.043	0.000	0.000
#9	BTC.TOP	0.315	0.331	0.287	0.223	0.000
#10	F2Pool	0.277	0.250	0.188	0.004	0.000
ϵ	–	0.000	-0.053	-0.030	-0.006	0.017

Table 5.11: Optimal allocations for the real case, $T = 6$, α=75% and different λ.

– * Pools 1-4 reference the aggregated players KanoPool, ConnectBTC, Solo CKPOOl and Bitcoin Russia.

Figure 5.21 shows the evolution of the allocations shown in table 5.11. It can be seen that, the more risk averse the profile, the less allocation is reserved for player #10 (since that player has the highest cost to mine). Also, the results for $\lambda = 1.0$ are not shown since for that risk profile there is no nucleolus ($\epsilon \geq 0.0$).



	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
■ Lambda 0	0,005	0,053	0,058	0,039	0,055	0,073	0,060	0,065	0,315	0,277
■ Lambda 0,25	0,000	0,064	0,060	0,044	0,053	0,009	0,051	0,056	0,331	0,250
■ Lambda 0,5	0,000	0,052	0,103	0,065	0,096	0,063	0,104	0,043	0,287	0,188
■ Lambda 0,75	0,000	0,183	0,199	0,000	0,158	0,032	0,200	0,000	0,223	0,004

Figure 5.21: Column bars chart for the optimal allocations for the real case, T = 6, α=75% and different levels of risk aversion (λ).

Let us now analyse λ 0.75 in regards to $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$. Table 5.12 shows that players #1, #4, #7 and #8 lose money when mining outside the pool (represented by their negative sign). While in the pool, their value is greater than out of the pool, indicating that there is value in being part of pool, created by the combinatorial characteristic of the Binomial distribution in the multi-period analysis. Player #7 goes from a negative to a positive value. Also, all ϵ_S are strictly lower than zero ($\mathcal{P}(\Phi)$ strictly higher than $\mathcal{O}(\Phi)$).

Player	$\mathcal{O}(\Phi)$	$\mathcal{P}(\Phi)$	ϵ_S
#1	-0.430	0.000	-0.430
#2	0.149	0.730	-0.581
#3	0.161	0.793	-0.632
#4	-0.133	0.000	-0.133
#5	0.002	0.629	-0.627
#6	0.123	0.129	-0.006
#7	-0.205	0.795	-1.000
#8	-0.232	0.000	-0.232
#9	0.881	0.888	-0.006
#10	0.009	0.015	-0.006

Table 5.12: $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$, $T = 6$, α=75% and λ=0.75.

The illustrative example studied in 5.1.3, specifically in figures 5.11 and 5.18, shows that all allocations for the nucleolus within a λ are contained in the area of λ above, with the addition of more points. For example: the area that contains the allocations within the nucleolus for λ=0.25 all are with the nucleolus for λ=0.5, 0.75 and 1.0. But allocations in the nucleolus for λ=0.5, 0.75 and 1.0 are not in the nucleolus for λ=0.25.

So, for this real case analysis, results in Table 5.11 are tested, trying to obtain the same results. Table 5.13 shows that, for a given λ, the optimal allocation returns $\epsilon \leq 0$ when tested for a higher λ. This shows the same behavior of the illustrative example, except for λ=0.0 and 1.0 (where the result is the same of Table 5.11, likely due to the solution to this risk profile not being in the nucleolus). Also, there is no strict generation of value for the allocation of $\lambda = 0.0$ when tested for other λ, with ϵ is close to zero for the results. Probably the result is really 0.0, with the difference due to computational issues when rounding numbers (including allocations, revenues and $CVaR_\alpha$).

	λ				
Obtained/Tested	0.0	0.25	0.5	0.75	1.0
0.0	0.000	0.005	0.009	0.001	0.017
0.25	0.787	-0.053	-0.0297	-0.006	0.017
0.5	2.42	1.04	-0.03	-0.006	0.017
0.75	7.97	5.13	2.37	-0.006	0.017
1.0	14.6	10.8	7.04	3.24	0.017

Table 5.13: ϵ obtained when testing x_n from different λ.

Now, we aim to test the results for the two intuitive sharing methods already exemplified before in this work: *relative hashpower* and *same allocation*. With a similar rationale than the one used in table 5.13, each allocation is tested for a value of λ, and the results are as:

λ	0.0	0.25	0.5	0.75	1.0
ϵ (relative hp)	1.18	0.661	0.231	0.127	0.022
ϵ (same allocation)	6.07	4.08	2.09	0.484	0.032

Table 5.14: ϵ obtained when testing different λ with x_n being the relative hp of the pool and same allocation.

As Table 5.14 shows, both intuitive sharing methods are not in the nucleolus of the game for any risk profile, which means that sharing the revenues according to the computational power of each player directly does not guarantee nucleolus to the coalition, as sharing rewards equally to every player also does not guarantee nucleolus to the coalition.

5.3.1 Gain in Probability - Multiperiod

Similarly to the analysis in section 5.1.2, the same test is conducted to the multi period analysis, aiming to observe the influence of a gain in probability to the pool:

	β				
Player	0.00	0.01	0.02	0.05	0.07
#1	0.000	0.042	0.058	0.073	0.080
#2	0.064	0.059	0.074	0.080	0.085
#3	0.060	0.063	0.078	0.082	0.087
#4	0.044	0.041	0.063	0.075	0.081
#5	0.053	0.057	0.076	0.080	0.086
#6	0.090	0.074	0.092	0.087	0.091
#7	0.051	0.056	0.082	0.082	0.087
#8	0.055	0.060	0.086	0.084	0.088
#9	0.331	0.297	0.255	0.185	0.162
#10	0.250	0.251	0.136	0.172	0.153
ϵ	-0.053	-0.325	-0.984	-1.390	-3.316

Table 5.15: Allocation results for different β, T=6 and $\lambda = 0.25$.

Table 5.15 indicates a shift in allocations when raising the probability of a given coalition, to mine. For instance, Player 1 has no allocation when β is zero, but sees a share of 8% of the pool's reward, when the probability to mine is raised by β=7%. On the other hand, Player 10 sees a decrease in its share. With higher probabilities to mine, the expected return is altered, so are $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$. Nonetheless, it is not only the players with lower hash power that see an increase in their profits, since the average return of the pool is raised with the pool mining more often, as table 5.16 shows:

	β				
Player	0.00	0.01	0.02	0.05	0.07
$\mathbf{E}[R_S(\tilde{\pi}_t^S, \tilde{\gamma}_{t,S})]$	15.5	22.7	29.4	49.7	63.2

Table 5.16: Expected returns for different β, T=6 and $\lambda = 0.25$.

5.3.2 Future without fixed income

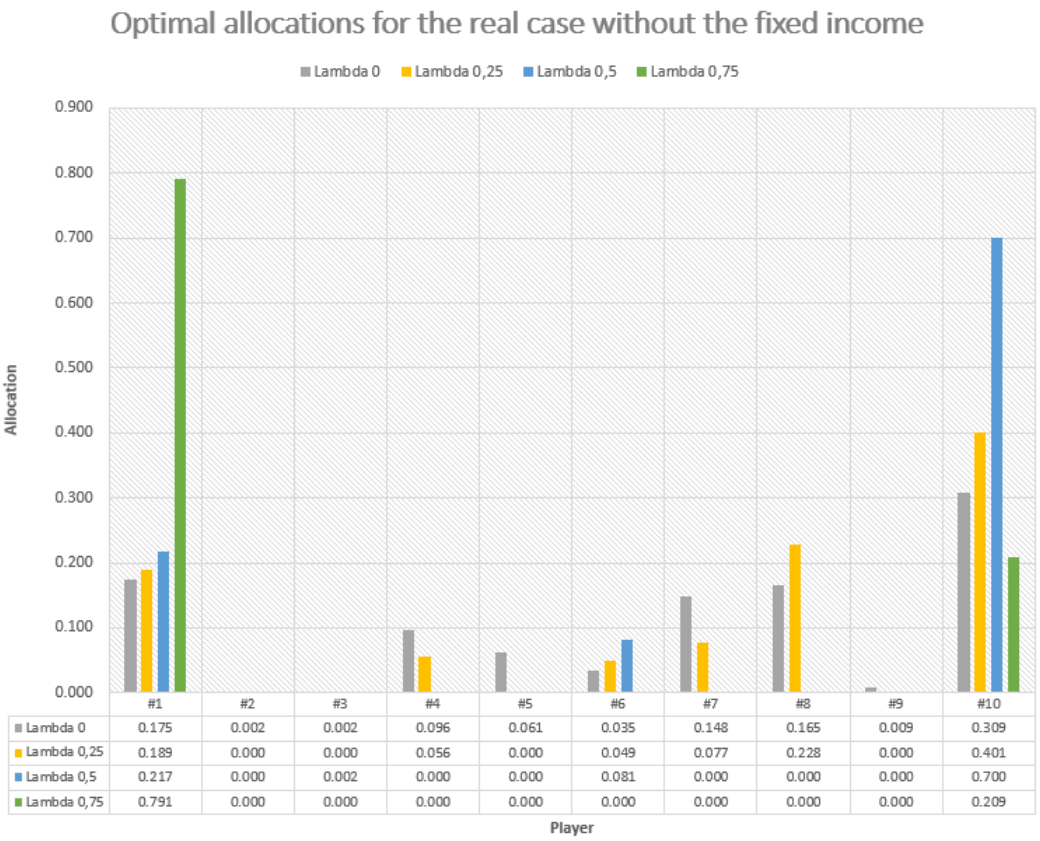
Bitcoin was proposed by Satochi Nakamoto to have a fixed amount of currency being available for transaction. By the process of halving, already discussed in this work, every 210.000 blocks added to the ledger, the fixed income is split in half. In the long run, this means that there will be a moment where players will only rely in the variable income from mining blocks.

With the data provided by Blockchain.info[1] for the variable income during the first semester of 2018, we used a log-normal distribution to model this uncertain factor. The estimated parameters by the Maximum Likelyhood method found were: $\mu = -0.048$ and $\sigma = 0.274$, with an expected value for the bitcoin of 0.98 BTC. This way, a context with computational powers and costs to mine similar to table 5.10, with fixed income being zero, is tested. The results are exhibited in table 5.17 and figure 5.22:

[1] https://www.blockchain.com/charts/transaction-fees, accessed may 5th, 2020

	λ				
Player	0.0	0.25	0.5	0.75	1.0
#1	0.175	0.189	0.217	0.791	0.000
#2	0.002	0.000	0.000	0.000	1.000
#3	0.002	0.000	0.002	0.000	0.000
#4	0.096	0.056	0.000	0.000	0.000
#5	0.061	0.000	0.000	0.000	0.000
#6	0.035	0.049	0.081	0.000	0.000
#7	0.148	0.077	0.000	0.000	0.000
#8	0.165	0.228	0.000	0.000	0.000
#9	0.009	0.000	0.000	0.000	0.000
#10	0.309	0.401	0.700	0.209	0.000
ϵ	0.000	-0.023	-0.040	-0.058	0.017

Table 5.17: Optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income.



	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
■ Lambda 0	0.175	0.002	0.002	0.096	0.061	0.035	0.148	0.165	0.009	0.309
■ Lambda 0,25	0.189	0.000	0.000	0.056	0.000	0.049	0.077	0.228	0.000	0.401
■ Lambda 0,5	0.217	0.000	0.002	0.000	0.000	0.081	0.000	0.000	0.000	0.700
■ Lambda 0,75	0.791	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.209

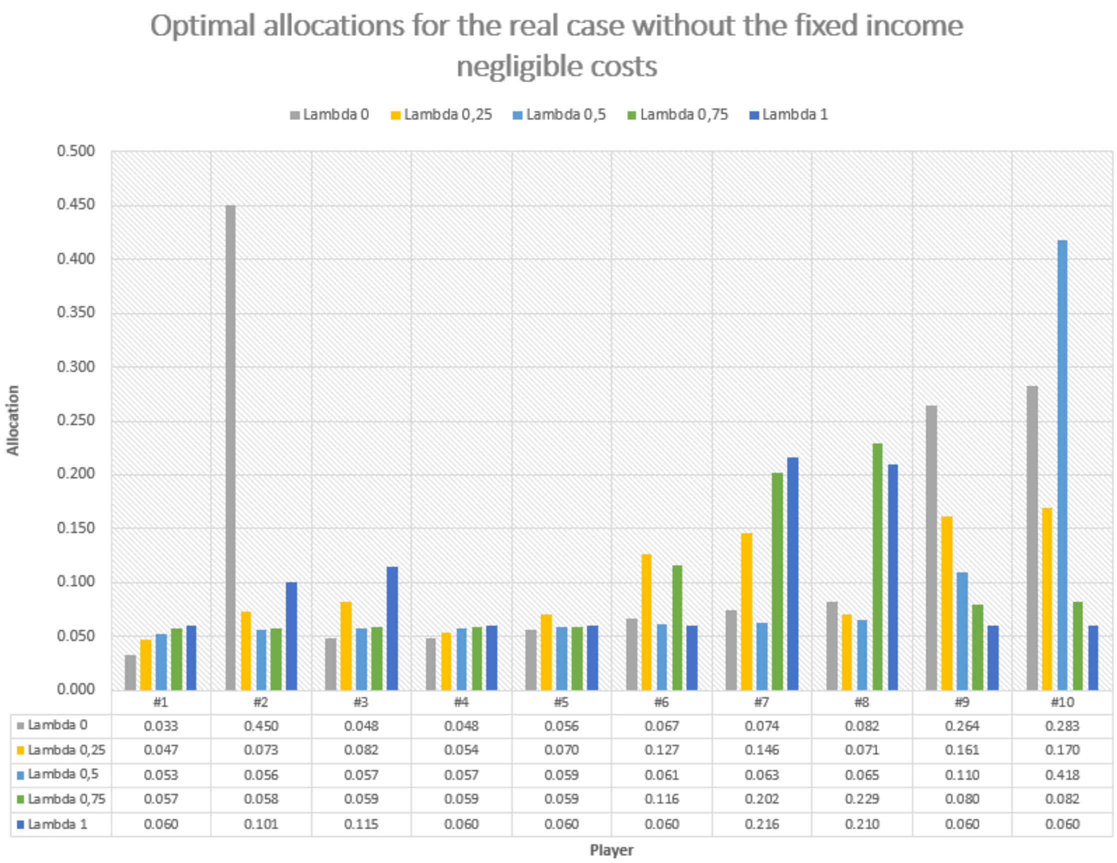
Figure 5.22: Column bars chart for the optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income.

Table 5.17 and figure 5.22 show that there is no strict generation of value in being in the pool, considering the presented situation (no fixed incomes and variable incomes only, and costs as in table 5.10) for $\lambda = 0.0$ and $\lambda = 1.0$. For the other values of λ, there is a small benefit to being in the pool, but the optimal allocations do not match the pattern seen so far, with many players having zero gains or cases with only two players having allocations greater than zero. That is not a desired result, since most of the pool does not have any financial gain to mine, thus being likely to leave the network.

Now, we propose a different approach to analyse this context. Let us consider that, until the fixed revenues become zero, computers advance in the natural way of decreasing prices for better performance and that the costs of energy tend to go down, with new setups for mining bitcoin becoming more efficient. This way, lowering the overall costs to mine by an extreme approach, until zero. For zero fixed income and the variable income as expressed in this section, we now obtain:

	λ				
Player	0.0	0.25	0.5	0.75	1.0
#1	0.033	0.047	0.053	0.057	0.060
#2	0.045	0.073	0.056	0.058	0.101
#3	0.048	0.082	0.057	0.059	0.115
#4	0.048	0.054	0.057	0.059	0.060
#5	0.056	0.070	0.059	0.059	0.060
#6	0.067	0.127	0.061	0.116	0.060
#7	0.074	0.146	0.063	0.202	0.216
#8	0.082	0.071	0.065	0.229	0.210
#9	0.264	0.161	0.110	0.080	0.060
#10	0.283	0.170	0.418	0.082	0.060
ϵ	0.000	-0.073	-0.145	-0.218	-0.291

Table 5.18: Optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income and with negligible costs.



	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
■ Lambda 0	0.033	0.450	0.048	0.048	0.056	0.067	0.074	0.082	0.264	0.283
■ Lambda 0,25	0.047	0.073	0.082	0.054	0.070	0.127	0.146	0.071	0.161	0.170
■ Lambda 0,5	0.053	0.056	0.057	0.057	0.059	0.061	0.063	0.065	0.110	0.418
■ Lambda 0,75	0.057	0.058	0.059	0.059	0.059	0.116	0.202	0.229	0.080	0.082
■ Lambda 1	0.060	0.101	0.115	0.060	0.060	0.060	0.216	0.210	0.060	0.060

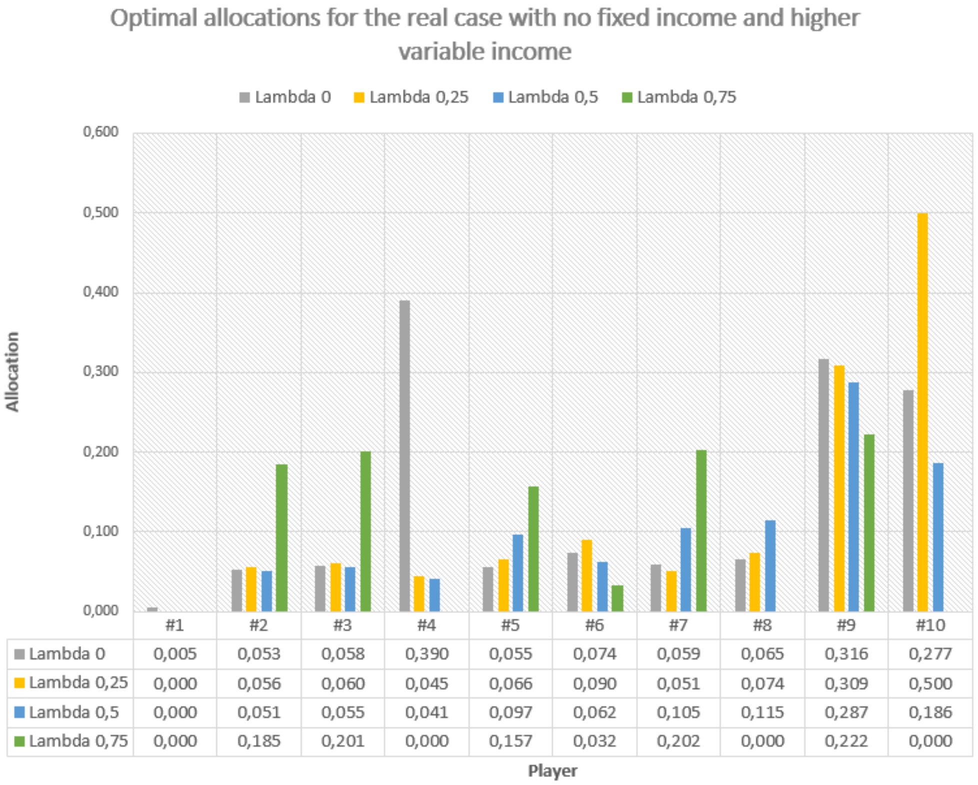
Figure 5.23: Column bars chart for the optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income and with negligible costs.

Table 5.18 and figure 5.23 show a reward sharing more akin to the pattern seen in this work, unlike table 5.17. All players have gains, and ϵ is even lower, showing that there is more value to being part of the pool. This is a behavior explained by the absence of significant mining costs. Even for $\lambda = 1.0$, there is a substantial value in being part of the pool.

Besides lowering the costs, another likely prediction with the end of fixed incomes, is that miners will ask for higher taxes to add transactions to new blocks. Our next analysis brings the costs back to mining, while considering the expected value of the Log-normal distribution to be the same as today's fixed income (12.5 BTC), changing the variable income distribution parameters to: $\mu = 2.488$ and $\sigma = 0.274$. The results are then presented in table 5.19 and figure 5.24:

Player	λ				
	0.0	0.25	0.5	0.75	1.0
#1	0.005	0.000	0.000	0.000	0.000
#2	0.053	0.056	0.051	0.185	1.000
#3	0.058	0.060	0.055	0.201	0.000
#4	0.039	0.045	0.041	0.000	0.000
#5	0.055	0.066	0.097	0.157	0.000
#6	0.074	0.090	0.062	0.032	0.000
#7	0.059	0.051	0.105	0.202	0.000
#8	0.065	0.074	0.115	0.000	0.000
#9	0.316	0.309	0.287	0.222	0.000
#10	0.277	0.250	0.186	0.000	0.000
ϵ	0.000	-0.049	-0.027	-0.005	0.017

Table 5.19: Optimal allocations for the real case, T = 6, α=75% and different λ, without the fixed income and with higher variable incomes.



	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
■ Lambda 0	0,005	0,053	0,058	0,390	0,055	0,074	0,059	0,065	0,316	0,277
■ Lambda 0,25	0,000	0,056	0,060	0,045	0,066	0,090	0,051	0,074	0,309	0,500
■ Lambda 0,5	0,000	0,051	0,055	0,041	0,097	0,062	0,105	0,115	0,287	0,186
■ Lambda 0,75	0,000	0,185	0,201	0,000	0,157	0,032	0,202	0,000	0,222	0,000

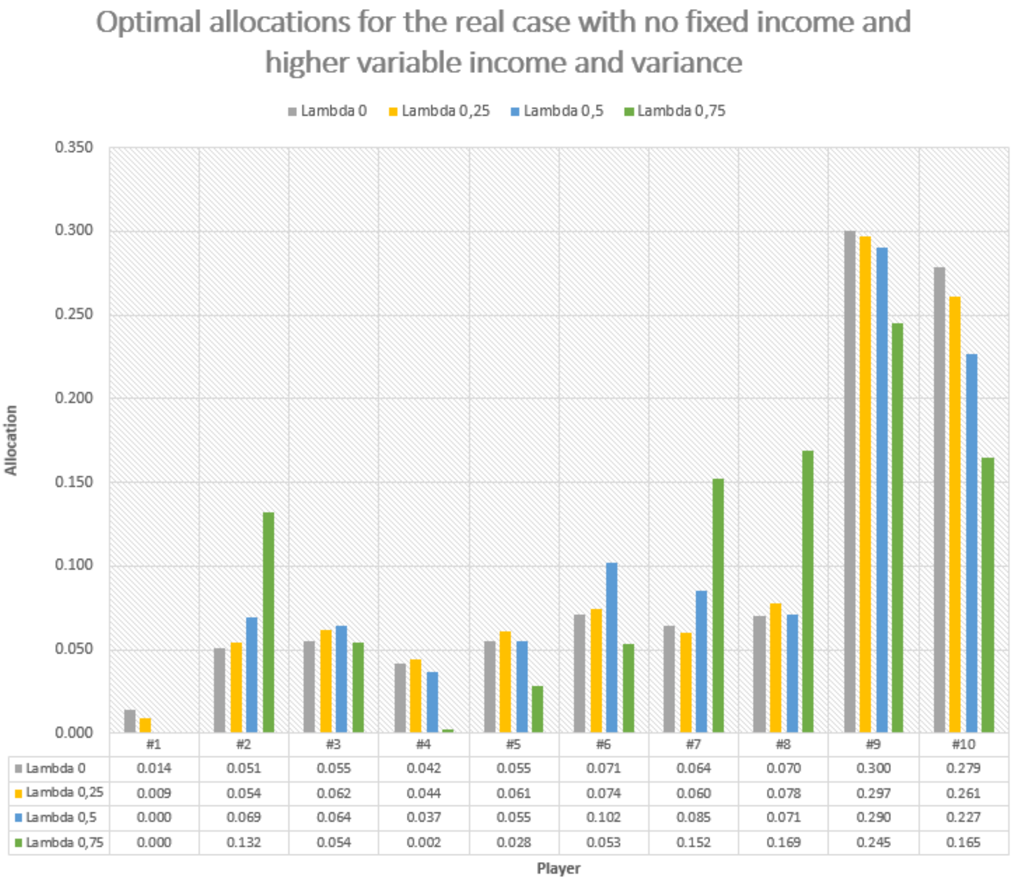
Figure 5.24: Column bars chart for the optimal allocations for the real case, T = 6, α=75% and different λ, without the fixed income and with higher variable incomes.

Table 5.19 and figure 5.24 show a result similar to table 5.11 in terms of allocations and ϵ. Since the variable income has an expected value close to that of the fixed revenue per block, but being stochastic, it is, on average, the same. This is an important result: even when, in the future, the fixed income for mining bitcoins lowers to zero, it will still be possible to share rewards in a pool in the same way as today, by having an increase in the variable incomes. This result also shows that our model shares the reward accordingly to the profits of the pool, not changing it with the source of income.

Lastly, we study a case in the future where the variable income has reached a similar magnitude to today's fixed income, and the variance of the log-normal distribution is the triple of today's. With this setup, we propose a market where this fee to add a transaction to the next block is strongly volatile. The expected value of a reward is still 12.5 BTC for the variable income. The results are:

	λ				
Player	0.0	0.25	0.5	0.75	1.0
#1	0.014	0.009	0.000	0.000	0.000
#2	0.051	0.054	0.069	0.132	1.000
#3	0.055	0.062	0.064	0.054	0.000
#4	0.042	0.044	0.037	0.002	0.000
#5	0.055	0.061	0.055	0.028	0.000
#6	0.071	0.074	0.102	0.053	0.000
#7	0.064	0.060	0.085	0.152	0.000
#8	0.070	0.078	0.071	0.169	0.000
#9	0.300	0.297	0.290	0.245	0.000
#10	0.279	0.261	0.227	0.165	0.000
ϵ	0.000	-0.073	-0.144	-0.063	0.017

Table 5.20: Optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income, with higher variable incomes and higher variance.



	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
■ Lambda 0	0.014	0.051	0.055	0.042	0.055	0.071	0.064	0.070	0.300	0.279
■ Lambda 0,25	0.009	0.054	0.062	0.044	0.061	0.074	0.060	0.078	0.297	0.261
■ Lambda 0,5	0.000	0.069	0.064	0.037	0.055	0.102	0.085	0.071	0.290	0.227
■ Lambda 0,75	0.000	0.132	0.054	0.002	0.028	0.053	0.152	0.169	0.245	0.165

Figure 5.25: Column bars chart for the optimal allocations for the real case, $T = 6$, α=75% and different λ, without the fixed income, with higher variable incomes and higher variance.

Table 5.20 and figure 5.25 show values of ϵ lower than the other setups in this section, but different allocation rewards between players. When raising the variance, coalitions out of the pool are more affected by lower rewards than the ones in the pool, since those have the pool to support them. So, ϵ is higher then in the Real Case example because $\mathcal{O}(\Phi)$ is lower, and not because $\mathcal{P}(\Phi)$ is raising. In fact, there is no negative value for all $\mathcal{P}(\Phi)$.

5.3.3 Optimal allocations behavior with different objective functions

Through this work, we aim to obtain the optimal allocations for a pool of miners in a *Bitcoin* network with an optimization model that minimizes the difference between being in and out of the pool, or as Equation (4-11) states, to minimize the maximum difference between the value of being in the pool and out of the pool.

However, there are more possible goals miners might want to reach when trying to obtain a "fair" allocation of mining rewards other than measuring the difference said above. Specifically, a pool might want to reward its players by their weighted average ($\frac{hp_n}{C_n}$, or the percentage of hashpower a player in the pool has in relation to his cost to mine) or by their contribution to the pool. For instance, in the real case analyzed, BTC.TOP & F2Poll have 222.72% and 245.45% more hashpower than Bitclub Network, the "best" miner besides them. The pool might want to maximize the rewards given to those two players, in order to maintain them in the pool and help others sustain a more frequent gain in the network.

With this mindset, we propose an analysis of the behavior of allocations for each alternative objective functions (OF):

1. OF 1 - *Infinite norm* - Equation (4-11)
2. OF 2 - *Average weighting* - $\max \sum_{n \in \mathcal{N}} \frac{hp_n}{C_n}$
3. OF 3 -*Thankful contributors* - $\max x_9 + x_{10}$

Table 5.21 shows the optimal allocations for each objective function proposed and the higher difference of $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$. Additionally, in column 3 we set for OF1 $\epsilon = 0.0$ and present the respected optimal allocations.

Player	OF 1*	OF 1	OF 2*	OF 3*
#1	0.000	0.00	0.00	0.00
#2	0.052	0.047	0.098	0.047
#3	0.103	0.051	0.106	0.051
#4	0.065	0.014	0.014	0.014
#5	0.096	0.036	0.036	0.036
#6	0.063	0.058	0.058	0.058
#7	0.104	0.021	0.021	0.021
#8	0.043	0.023	0.023	0.023
#9	0.287	0.283	0.457	0.562
#10	0.188	0.462	0.183	0.183
$\mathcal{O}(\Phi)$ - $\mathcal{P}(\Phi)$	-0.03	0.00	0.00	0.00

Table 5.21: Optimal allocations for the real case, $T = 6$, α=75%, $\lambda = 0.5$ and different OFs.

– OF1* refers to the optimal result the model returns for OF 1, already presented at the beginning of this subsection.

Firstly, we highlight that by using the objective functions 2 & 3, the respective ϵ is zero. Also, it can be noted that OF 3 aims to maximize the allocation of players #9 and #10, that are the ones in the pool with the highest hashpower. Thus, the result of such optimization translates to the "power" such players have in the pool (74.5%). Interestingly enough, that is the same sum obtained by those players in OF 1 when ϵ is set at zero (column 3). Because allocations for players #1 to #8 are the same for OFs 1 and 3, minimizing the maximum difference between $\mathcal{O}(\Phi)$ and $\mathcal{P}(\Phi)$ and maximizing players with maximum hashpower produce a similar result, only that player #9 obtains a higher share in OF 3, likely due to him having lower costs than player #10. OF 2 shows interesting results. Some players produce the same results as OFs 1 and 3 (players #4 to #8), but players #2 and #3 now have a higher share. Since those players have lower costs to mine, maximizing the average between hashpower and costs make them more valuable, and thus they bring more to the pool.

Now, we let ϵ vary from 0 (zero) to the result of ϵ in Table 5.11 for λ=0.5. In other words, we define the limits: the lowest value of ϵ that the optimization model returns for OF 1, and zero. If, lets say, ϵ is -0.5, the optimization model is forced for ϵ to be between -0.5 and 0.0, with steps of 0.005 (with the grand total of 100 steps). Those are forced into the optimization model for each OF, and the allocations obtained are plotted in a accumulated bar style in figures 5.26, 5.27 and 5.28.

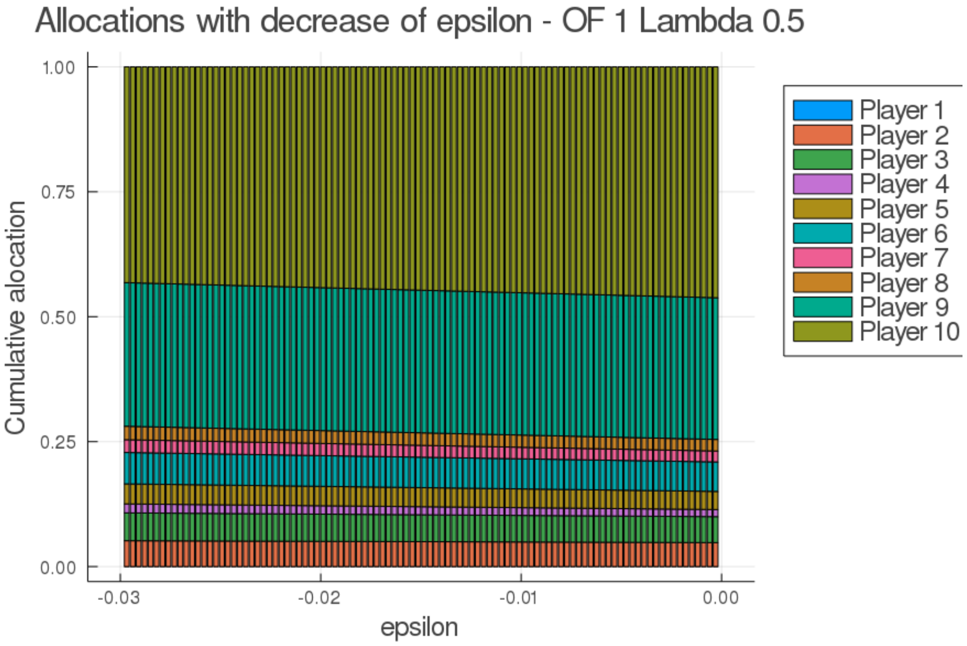


Figure 5.26: Allocations for Objective Function (OF) 1.

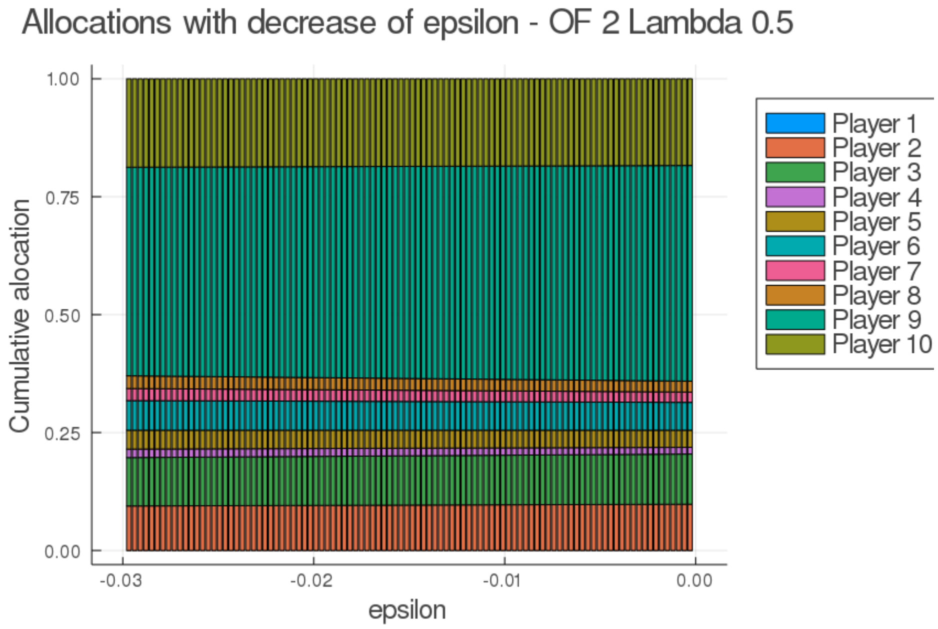


Figure 5.27: Allocations for Objective Function (OF) 2.

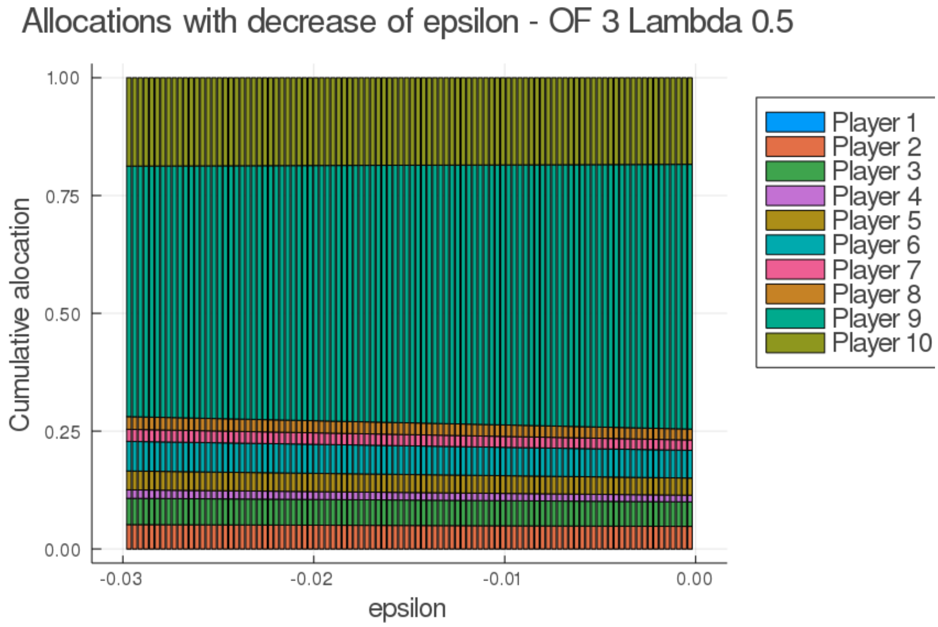


Figure 5.28: Allocations for Objective Function (OF) 3.

The analysis of Figures 5.26, 5.27 and 5.28 shows variations in allocations for all 3 OFs, but consistent with results in table 5.21. For instance, despite raising with the increase of ϵ, player #10 has a higher share than player #9 for OF 1, whilst having the opposite for OFs 2 & 3. Players with lower hashrates have variations so small, that they are not perceivable in the figures.

Recall from table 5.21, the optimal allocations for OF 2 & OF 3 recovers an ϵ equal to zero. This result indicates that this allocation is, although theoretically in the nucleolus, is fragile as there is no strict benefit. Therefore, the last analysis of objective functions consists in comparing the allocations with ϵ being forced to zero, and ϵ being forced to be the same as OF 1*, but for OF 2 and OF 3:

	OF1*	OF 2		OF 3	
Player	ϵ min	ϵ	ϵ min	ϵ	ϵ min
#1	0.00	0.00	0.00	0.00	0.00
#2	0.052	0.098	0.094	0.047	0.051
#3	0.103	0.106	0.102	0.051	0.055
#4	0.065	0.014	0.017	0.014	0.017
#5	0.096	0.036	0.040	0.036	0.040
#6	0.063	0.058	0.062	0.058	0.062
#7	0.104	0.021	0.025	0.021	0.025
#8	0.043	0.023	0.026	0.023	0.026
#9	0.287	0.457	0.442	0.562	0.531
#10	0.188	0.183	0.187	0.183	0.187
ϵ	-0.030	0.00	-0.030	0.00	-0.030

Table 5.22: Optimal allocations for the real case, $T = 6$, α=75%, ϵ 0.0, $\lambda =$ 0.5 and min, and OFs 2 and 3.

Table 5.22 shows that the allocations obtained by the optimization model return similar results with ϵ being forced to 0.0 or to the minimum value obtained by OF 1, for each of the other objective functions proposed. Still, those are different than the optimal allocations for OF 1.

A case here can be made: when trying to obtain the optimal allocations directly through OF 2 and OF 3, ϵ is not < 0.0. But after forcing it to the lowest ϵ obtained by OF 1, it does return allocations that combine both goals: to minimize ϵ (and thus, generating the highest value to the pool) and to share the rewards accordingly to a more “fair” method. That is a suggestion that this work proposes; to obtain the optimal allocations in 2 steps:

1. minimize ϵ through OF 1;

2. find new allocations forcing this ϵ with a new OF.

That way, a pool is trying to generate as much value as possible, while being able to share the rewards by the most appropriate ensuring “fairness” in the sharing of rewards.

6
Conclusion and future work

In this work, we construct models for the revenue and profit of a miner in the bitcoin network, whether alone or in a coalition. In addition, an optimization model to find the allocations of the players using a nucleolus-based quota allocation model has also been devised and implemented.

The profit model we propose takes into account the success a miner has in mining the next blocks, with the probability to mine being represented by a Bernoulli (for the next block) distribution, or a Binomial (for any number of blocks greater than 1) distribution. An allocation sharing method based on Cooperative Game Theory is used (nucleolus-based), and a characteristic (or value) function is proposed, combining the expected return in mining for bitcoin, and the risk associated with such activity, in the form of a Coherent Risk Measure (the Conditional Value at Risk).

The analysis of the illustrative example with 3 players shows that there is no strict benefit for them to be part of the pool for a single block. Also, the presence of costs to mine and the inclusion of a risk aversion profile deviates the solution from the intuitive allocation (sharing by computational power or equally sharing). There is strict benefit for the next block only if a gain in probability by being in the pool, is considered. Said gain is represented by a function proportional to the number of players in a coalition (when the coalition has more than 1 member).

While mining alone can provide higher incomes for a player, accumulated probabilities result indicates that said player is likely to spend most of its mining work with negative cash-flows, since his costs are always present. On the other hand, mining in a pool assures him positive cash flows more often, offering a less risky way to operate. When mining in a pool, the coalition has a higher hashpower than each player alone, and so the average time to correctly guess the hash of a new block becomes lower, raising the chances to mine the next block.

When looking at a multi-period setup, the combinatorial nature of mining brings value to the coalition more than working out of the pool, providing incomes that not only make players want to remain together, but gives them higher expected rewards in total. Also, the introduction of a risk measure is shown to change the optimal allocations of a pool, depending on the weight that it receives.

A visual representation of the core of the game was presented, showing all the possible allocations with 3 players. For the single period analysis, the core set is a singleton, while for the multi-period analysis there is a cloud representing many allocations in the nucleolus of the game. The longer the period considered, smaller the "area" representing the core of the game. Interesting to note, that the nucleolus raises from a single point, and then condensates again, as allocations converge. By increasing the number of blocks analyzed, we observed that intuitive sharing is not in the nucleolus of the game for any risk profile considered.

The real case analysis further stresses the optimization model and presents results that are in line with those of the illustrative example. We considered a future without the fixed income from mining and alternatives to maintain players with positive cash flows in this scenario. It can be shown that it is possible for the bitcoin network to remain operational with the miners still working mainly by raising the variable income and lowering mining costs.

Lastly, as each pool might consider sharing its rewards differently, a study with 3 proposed different object functions is conducted. Those objective functions contain the method in which each group of players might prefer to share the rewards, and so treats the "fairness" in quota allocations in Cooperative Game Theory for this work. We propose a strategy to allocate the rewards in 2 steps: first, minimize the difference between being in an out of the pool, and then use this difference to find the optimal allocations with a different objective function.

Suggestions for further works aims to raise the number of players included in the pool, as well as obtaining more robust representations of the variable incomes and the costs associated with mining Bitcoin. A time-dependent representation of the mining probability for each block is also to be considered. Also it is important to analyze the reason for the behavior of the allocations with different representations of risk profile.

Bibliography

[Artzner, 1998] ARTZNER, P., D. F. E. J.; HEATH, D.. **Coherent measures of risk**. 1998.

[Back, 2016] BACK A, CORALLO M, D. L. F. M. M. G. M. A. E. A.. **Enabling blockchain innovations with pegged sidechains**.

[Barron, 2008] BARRON, E. N.. **Game Theory: An Introduction: 1st (First) Edition**. Wiley, 2008.

[BitcoinMiningPools] **Bitcoin mining pools**. https://www.buybitcoinworldwide.com/mining/pools/. Accessed: 2019-11-27.

[Chatzigiannis, 2019] CHATZIGIANNIS, P., B. F. G. I.; LI, J.. **Diversification across mining pools: Optimal mining strategies under pow**. arXiv preprint, 2019.

[CryptoCompare] **Cryptocompare**. https://www.cryptocompare.com/mining/pools/. Accessed: 2019-11-27.

[Dev, 2014] DEV, J. A.. **Bitcoin mining acceleration and perfomance quantification**. 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), 2014.

[Diffie, 1976] DIFFIE W., H. M.. **New directions in cryptography**. IEEE Trans Inf Theory, 1976.

[Dwyer, 2015] DWYER, G.. **The economics of bitcoin and similar private digital currencies**. EJournal of Financial Stability, (17).

[Eichengreen, 2019] EICHENGREEN, B. J.. **Globalizing capital: a history of the international monetary system**. Princeton University Press, 2019.

[Equilibrium points in n-person games, 1950] NASH, J. F.; OTHERS. **Equilibrium points in n-person games**. Proceedings of the national academy of sciences, 36(1):48–49, 1950.

[Eval, 2013] EYAL, I.; SIRER, E.. **Majority is not enough: Bitcoin mining is vulnerable***. 2013.

[Fanzeres, 2015] FANZERES, B., S. A.; BARROSO, L.. **Contracting strategies for renewable generators: A hybrid stochastic and robust optimization approach**. IEEE Transactions on Power Systems, 30(4):1825–1837, 2015.

[Faria, 2009] FARIA, E., B. L. A. K. R. G. S.; PERERIA, M. V.. **Allocation of firm-energy rights among hydro plants: An aumann–shapley approach**. IEEE Transactions on Power Systems, 24(2):541–551, 2009.

[Freire, 2015] FREIRE, L.; STREET, A.; LIMA, D. A. ; BARROSO, L. A.. **A hybrid milp and benders decomposition approach to find the nucleolus quota allocation for a renewable energy portfolio**. IEEE Transactions on Power Systems, 30(6):3265–3275, 2015.

[Freire, 2017] FREIRE, L.. **On the comparison of computationally efficient quota-sharing methodologies for large-scale renewable generation portfolios**, 2017.

[Friedman, 1977] N., F.. **Nobel lecture: Inflation and unemployment**. The Journal of Political Economy, 85(3m):451–472, 1977.

[Jezic, 2016] JEZIC, G., C.-B. Y. J. H. R. J. J. L. C.. **Agent and Multi-Agent Systems: Technology and Applications**. Princeton University Press, 2016.

[Junqueira, 2007] JUNQUEIRA, M., D. C. L. C. B. L. A. O. G. C. T.; L. M. PEREIRA, M. V.. **An aumann-shapley approach to allocate transmission service cost among network users in electricity markets**. IEEE Transactions on Power Systems, 22(4):1532–1546, 2007.

[Kiayias, 2016] KIAYIAS, A.; KOUTSOUPIAS, E.; KYROPOULOU, M. ; Y., T.. **Blockchain mining games**. EC, p. 24–28, 2016.

[Kohlberg, 1971] KOHLBERG, E.. **On the nucleolus of a characteristic function game**. SIAM Journal of Applied Math, 20(1):1443–1471, 1971.

[Kroll, 2013] KROLL, J.; DAVEY, I. ; FELTEN, E.. **The economics of bitcoin mining or, bitcoin in the presence of adversaries**. 2013.

[Lewenberg, 2015] LEWENBERG, Y.; BACHRACH, Y.; SOMPOLINKSY, Y.; ZOHAR, A. ; ROSENSCHEIN, J.. **Bitcoin mining pools: A cooperative theoretic analysis**. Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, 2015.

[Manea, 2017] MANEA, M.. **Cooperative games**. Joule, 2, 2017.

[Markowitz, 1952] MARKOWITZ, H.. **Portfolio selection**. The Journal of Finance, 7.

[Mattila, 2016] MATTILA J, SEPPÄLÄ T, N. C. S. R. T. M. B. A. E. A.. **The blockchain phenomenon-the disruptive potential of distributed-consensus architectures**.

[Merlinda, 2019] ANDONI, M., R. V. F. D. A. S. G. D. J. D. M. P. P. A.. **Blockchain in the energy sector: A systematic review of challenges and opportunities**. Renewable and Sustainable Energy Reviews, 2019.

[MiningCostsbyCountry] **Estimated electricity cost of mining one bitcoin by country**. `https://powercompare.co.uk/bitcoin-electricity-cost/`. Accessed: 2019-11-27.

[Muftic, 2016] MUFTIC, S.. **Overview and analysis of the concept and applications of virtual currencies**.

[Naheem, 2015] BAGEEN, M.. **Hsbc swiss bank accounts-aml compliance and money lauderting implications**. Journal of Financial Regulation and Compliance, 23(3):285–297, 2015.

[Nakamoto, 2008] NAKAMOTO, S.. **Peer-topeer eletronic cash system**. 2008.

[Naucler, 1950] MATTILA J, SEPPÄLÄ T, N. C. S. R. T. M. B. A. E. A.. **Industrial blockchain platforms: An exercise in use case development in the energy industry.**

[Newman, 2015] NEWMAN, A.. **What the "right to be forgotten" means or privacy in a digital age**. Science, 30:507–508, 2015.

[Non-cooperative Games, 1951] NASH, J.. **Non-cooperative games**. Annals of Mathematics, 1951.

[PwC, 2015] UTILITIES, P. G. P. .. **Blockchain - an opportunity for energy producers and consumers?** IEA, World Energy Outlook, 2015.

[Rockafellar, 2002] ROCKAFELLAR, R. T.; URYASEV, S.. **Conditional value-at-risk for general loss distributions**. Journal of Banking & Finance, (26):1443–1471, 2002.

[Rosenfeld, 2011] ROSENFELD, M.. **Analysis of bitcoin pooled mining reward systems**. IEEE Transactions on Power Systems, 2011.

[Shapley, 1953] SHAPLEY, L.. **A value for n-person games in contributions to theory of games**. Annals of Mathematical Studies, 28(1):307–317, 1953.

[Street, 2009] STREET, A.. **On the conditional value-at-risk probability-dependent utility function**. 2009.

[The Bargaining, 1950] NASH, J.. **The bargaining**. Econometrica, 1950.

[Two-person, 1953] NASH, J.. **Two-person cooperative games**. Econometrica, 1953.

[Wheretomine] **Where to mine - mining pools overview**. `https://wheretomine.io/pools/`. Accessed: 2019-11-27.

[countdown] **Bitcoin block reward halving countdown**. `https://www.bitcoinblockhalf.com/`. Accessed: 2019-11-26.