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

Eduardo Bolonhez  
Pontifical Catholic University of Rio de Janeiro: Pontificia Universidade Catolica do Rio de Janeiro

Thuener Silva  
Pontifical Catholic University of Rio de Janeiro: Pontificia Universidade Catolica do Rio de Janeiro

Bruno Fanzeres dos Santos (✉ bruno.santos@puc-rio.br)  
Pontifical Catholic University of Rio de Janeiro: Pontificia Universidade Catolica do Rio de Janeiro  
https://orcid.org/0000-0003-0104-3419

Research Article

Keywords: Bitcoin, blockchain networks, allocation models

DOI: https://doi.org/10.21203/rs.3.rs-614361/v1

License: ©️ This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License
**Abstract (structured along the lines of: background, aim, method, results, conclusion –150 words):**

The Bitcoin operates in a Blockchain network under which a group of participants are responsible for adding new blocks into the chain. These participants are called miners and the ones that successfully add a block into the network receive a reward for their work. As the technology evolved over the years, this "mining" process has become more challenging with miners facing long periods without positive cash flow, while still having costs associated. This resulting business architecture has driving participants away from the technology, jeopardizing its operations, and defying its progression. In order to cope with this issue, an alternative to provide miners' financial sustainability is to join a mining pool, which main purpose is to mitigate this cash flow sparsity by sharing the (more-recurrent) rewards obtained by the group. Therefore, in this work, we propose a reward sharing methodology for mining pools based on the Nucleolus of a stochastic cooperative game. A risk-averse value functional based on the Conditional Value-at-Risk (CVaR) is used to characterize the game’s certainty equivalent.
Two numerical experiments were conducted in this work: (i) one based on a small, illustrative network; and (ii) one derived from real data of the Bitcoin-refunded Blockchain network. The focus of the experiments is on the incremental value of the proposed methodology over using intuitive allocations (uniform and based on computational power) and in what extent the relative increase in the mining likelihood by playing as a group benefits the pool stability. Finally, we discuss and numerically analyze a nested procedure based on the proposed Nucleolus-based allocation seeking for higher “fairness” in sharing the pool rewards.
A Nucleolus-Based Quota Allocation Model for the Bitcoin-Refunded Blockchain Network

Abstract The Bitcoin operates in a Blockchain network under which a group of participants are responsible for adding new blocks into the chain. These participants are called miners and the ones that successfully add a block into the network receive a reward for their work. As the technology evolved over the years, this “mining” process has become more challenging with miners facing long periods without positive cash flow, while still having costs associated. This resulting business architecture has driving participants away from the technology, jeopardizing its operations, and defying its progression. In order to cope with this issue, an alternative to provide miners’ financial sustainability is to join a mining pool, which main purpose is to mitigate this cash flow sparsity by sharing the (more-recurrent) rewards obtained by the group. Therefore, in this work, we propose a reward sharing methodology for mining pools based on the Nucleolus of a stochastic cooperative game. A risk-averse value functional based on the Conditional Value-at-Risk (CVaR) is used to characterize the game’s certainty equivalent. Two numerical experiments were conducted in this work: (i) one based on a small, illustrative network; and (ii) one derived from real data of the Bitcoin-refunded Blockchain network. The focus of the experiments is on the incremental value of the proposed methodology over using intuitive allocations (uniform and based on computational power) and in what extent the relative increase in the mining likelihood by playing as a group benefits the pool stability. Finally, we discuss and numerically analyze a nested procedure based on the proposed Nucleolus-based allocation seeking for higher “fairness” in sharing the pool rewards.

Keywords Bitcoin · Blockchain Network · Cooperative Game Theory · Nucleolus of a Stochastic Game · Risk Aversion · Stochastic Optimization
1 Introduction

Over the course of history, mankind has traded goods using many different tools. Gold, silver, copper, metal coins, paper currency, debit cards and a plethora of other examples illustrate the utensils used. In present days, the fiduciary system is predominant in our society. This system is based on the trust that a debtor will honour its obligations. Nonetheless, money transactions are the responsibility of governments and banks, with the former being able to print new money at will. Those aspects have been heavily criticized in the past years by different economists, arguing that the model is excessively centralized, inducing non-negligible levels of inflation and, in the long-term, unemployment (Friedman, 1977). Furthermore, an individual has limited knowledge over the amount of money circulating throughout the financial system, but the institutions are fully aware of their financial activities. With many scandals arising from the lack of digital privacy, leaking of personal data (Newman, 2015) and misuse of “people’s money” (Naheem, 2015), a constantly increasing share of society is advocating towards a system design that hands over control, at least partially, back in their hands (Alt and Puschmann, 2012).

Following this aim for a more decentralized system, Bitcoin/Blockchain technologies emerged in the past years aiming at (ultimately) fulfilling this desire. Fundamentally, Bitcoin is a cryptocurrency (Narayanan et al., 2016) backed by its own Blockchain network, and was firstly introduced in a 2008 seminal article by Satoshi Nakamoto (Nakamoto, 2008). Its main characteristic is the absence of financial/physical coverage and to be marginal to all currently fiduciary system. Structurally, the Bitcoin makes use of distributed existing and private machinery (which owners are typically referred to as Miners) to verify each transaction in its Blockchain network. Each transaction needs to be verified and registered (within chained blocks), and thus the Blockchain network intends to mimic a (digital) “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 this process, every miner takes part in a “guessing game” of information presented in the new block, using their computational power. For consistency and stability, the network adjusts its “guessing” difficulty with ultimate goal of establishing a fixed time of completion for each block1. The network is not hosted in any specific server around the globe, but is embedded within each machine connected to the Blockchain network, thus challenging external attacks, since a malicious agent would have to violate all computers in the network to succeed. Although each specific Blockchain ledger has its own types of validation and consolidation, they are typically structured by means of distributed voting (Mattila, 2016; Mattila et al., 2016) through a consensus on the state of the ledger (Shrivastava et al., 2020).

To effectively participate in the Proof of Work, as higher is the miner computational power, more likely it is to succeed in verifying the information contained in the next block. From a financial perspective, by “guessing” this information, the miner is thus rewarded with a combination of a fixed (deterministic) payment along with a block-dependent variable income. Therefore, from a single-miner viewpoint, the cash-flow nature of the business typically comprises several periods of income absence followed by “guessing” a block and receiving the aforementioned payment. More specifically, participating in this business inherently induces great uncertainty as a participant could spend months or even years without positive cash flows, but paying the costs necessary to keep its machinery on. As a consequence, this fat-tailed negatively-skewed cash flow structure could potentially induce miners to leave the Blockchain network, thus weakening its structure due to the lack of Bitcoin trading, leading the network to stop, collapsing the system (Derks et al., 2018). In order to cope with this issue, an alternative to provide miners’ financial sustainability is to join a mining pool (Dev, 2014). Roughly speaking, a mining pool constitutes of a coalition with multiple miners that share the revenues and costs by acting in the network, thus a particular case of a Cooperative Game (Lewenberg et al., 2015b). Its main purpose is to mitigate the sparsity of each miner cash

1 Currently, the fixed time is set to approximately 10 minutes (see, for instance, (Dwyer, 2015) for a thorough and profound discussion).
flow by sharing the (more-recurrent) rewards obtained by the pool due to the increase in computational power to verify the blocks information. As a consequence, the pool miners receive revenues from the Blockchain network more frequently and remain within the network due to the financial incentives provided by cooperation (Back et al., 2014), thus keeping the Bitcoin/Blockchain system active in a sustainable way (Eyal and Sirer, 2018; Kiayias et al., 2016).

Therefore, the objective of this work is to present a fair and efficient methodology to share the financial benefits of acting cooperatively in a mining pool within a Bitcoin-refunded Blockchain system. For this purpose, we make use of the concept of the Core of a cooperative game and design a Nucleolus-based quota allocation methodology within an stochastic game setting (Schmeidler, 1969; Shapley, 1967; Suijs and Borm, 1999). More specifically, the Nucleolus-based allocation method aims at defining a quota sharing that induces the maximum benefit of acting cooperatively compared to the highest-valued coalition possible to be set by the members of the pool. We argue that, although there is no consensus on which allocation method is better suited in general applications, in the particular context of this work, since the Nucleolus-based methodology tackles directly stability-issues of a cooperative group, it is better suited for the general goal of keeping the Blockchain system active, i.e., it is worthwhile for all miners to remain in the mining pool, creating thus a (financially) stable and economically efficient system. From a methodological perspective, on the other hand, the contextual structure requires to characterize the problem as a stochastic game due to the uncertainty in mining a block by the members of the pool. Furthermore, the fat-tailed negatively-skewed cash flow dynamics previously discussed induces the need for a game modeling based on risk-averse agents. In this work, we leverage on the fundamental properties of convex risk measures (Föllmer, 2016) and make use of the quantile-based one known as the Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev, 2002; Street, 2010) to induce risk-averse preferences within the stochastic cooperative game. The resulting risk-averse Nucleolus-based quota allocation methodology falls into the class of a non-linear model composed by a two-level system of optimization problems, thus not suitable for direct application of efficient mathematical programming techniques. Therefore, by applying a straightforward reformulation procedure, we devise an equivalent single-level two-stage optimization problem over which standard stochastic programming methods and algorithms/decompositions can be applied (Fanzeres et al., 2019, 2020; Freire et al., 2015). Two numerical experiments were conducted to illustrate the applicability and provide insights over the quota allocation methodology proposed in this work: (i) one based on a small, illustrative network; and (ii) one derived from real data of the Bitcoin-refunded Blockchain network. The focus of the experiments is on the incremental value of the proposed methodology over using intuitive allocations (uniform and based on computational power) and in what extent the relative increase in the mining likelihood by playing as a group benefits the pool stability. Finally, we discuss and numerically analyze a nested procedure based on the proposed Nucleolus-based allocation seeking for higher “fairness” in sharing the pool rewards.

This work is structured as follows: Section 2 thoroughly describes the Bitcoin and Blockchain network and outlines how mining pools bring more value to players. Section 3 presents the Nucleolus-based quota allocation methodology proposed in this work. Section 4 highlights the applicability of the method proposed in this work with two numerical experiments. Finally, Section 5 concludes the work.

2 Blockchain & Bitcoin Network

Bitcoin is a Decentralized Application operating in peer-to-peer (DApp) that runs within a network called Blockchain. It was first introduced in a seminal article by Satoshi Nakamoto (Nakamoto, 2008) in 2008, aiming at providing a “peer-to-peer version of electronic cash” allowing payments to be made without a financial institution while preventing double-spending. To achieve this goal, Nakamoto proposed the concept of a universal ledger with consensus for the information stored.
More specifically, the Bitcoin network is composed of aggregated information stored in chronological order into structures called blocks. Each block contains information such as transactions and fees. The network thus works as a ledger, and the information is imputed into each new block by a participant called miner. The miner is responsible for verifying and adding data to the chain in a process known as the Proof of Work. Each miner takes part in a computational “race” to guess the correct hash of the subsequent block. A hash algorithm turns an arbitrary amount of data into a fixed-length hash, and each block has a unique hash on its header that identifies it. Each participant then uses cutting-edge hardware that can perform several hash calculations per second, the hashrate. The relative computational power of each player in the network is called hashpower.

For consistency and stability, the Bitcoin network adjusts the difficulty to guess the next hash according to the total hash of players participating on it. In other words, as the total hashpowers fluctuates over time, so does the difficulty to mine a new block. The winning participant claims a rewards (decomposed in a fixed amount of Bitcoin plus a variable amount which depends on the specificities of the mined block) and the other players verify that the block is valid. If so, it gets added in a chronological chain subsequent to the previous block mined. It is noticeable that for Bitcoin to be transacted, it is of utmost importance for miners to compute it in the network. For users to trade Bitcoin for goods, they need to have both a private and a public key. The concept of public-key cryptography allows for this method to be perform without security issues (Diffie and Hellman, 1976; Andoni et al., 2019).

The Bitcoin price\(^2\) has largely oscillated over time (da Gama Silva et al., 2019), but such variation along with the the difficulty to mine\(^3\) and number of daily transactions\(^4\) provides important information. For instance, when prices 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 as first steps towards an equilibrium. For these reasons and to keep the network running, it is crucial to study and setting up incentives to miners not turn off their equipment or change networks even at times of adversity. It is also important 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 financial uncertainty in mining, an alternative is to join a mining pool.

2.1 Mining Pools and Reward sharing

As previously discussed, without miners, users would retain their Bitcoin in their private wallets without the possibility to commercialize them. A miner then uses its computational power to keep the network functioning, but it is an intrinsic risky and uncertain business environment since trying to guess the next hash against competitors throughout the world produces constant costs with a high uncertainty about incomes. The miner that finds the specific hash earns a fixed income\(^5\) plus a variable revenue in Bitcoins. The latter is composed by the sum of taxes from the transactions of a mined block. Once the halving process tend to reduce the fixed income to roughly zero as time goes on, the miners in the near future will earn only the variable revenue. At the beginning of Bitcoin,

\(^5\) In May 2021, of an amount of 6.25 Bitcoins, and this value is halved every 210.000 new blocks created (see Quartz – https://qz.com/681996/everything-you-need-to-know-about-the-bitcoin-halving-event/, accessed: November 26, 2019, for a better overview of this process).
there were fewer miners and the network difficulty was lower, allowing users to be able to mine from their homes with “modest” equipment. But as the technology developed and attracted new users, the difficulty rose, demanding more powerful machinery. Those lone miners not only observed their hashpower diluting to minimal values, but also an increase in mining profit volatility. More specifically, 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. On the other hand, miners with higher hashpower are more likely to succeed and be financially sustainable in the business.

An alternative to ensure more recurrent income is to participate in a pool, where a group of miners share their computational power in the mining process, increasing thus their chances of positive cash flows (Kroll et al., 2013). More precisely, when mining within a pool, due to the increase of joint computational power, the mining likelihood is higher. As a consequence, the pool gathers more block rewards over time, stabilizing the participants’ cash flow and, ultimately, enabling 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 et al., 2019) states, if a miner has the chance to select between various cryptocurrencies to lend his computational power, it is diversifying its portfolio in a similar way as proposed by the Markowitz Modern portfolio theory (Markowitz, 1952; Castro et al., 2020). Many Bitcoin Mining Pools already operate with popular reward sharing methods, such as Proportional, Pay-Per-Share, Slush Method and Pay-Per-Last-N-Shares (Rosenfeld, 2011). All of those methods have advantages and disadvantages (such as pool hopping). In this work, we propose a method that differs from the above, since it does not explicitly work with the concept of “shares” and “rounds”.

Participating in a pool is taking part in a game that can be studied by Game Theory. On the topic of Cooperative Games in bitcoin mining, (Lewenberg et al., 2015a) demonstrates the difficulty in achieving a cooperative equilibrium when considering delays in the communication between miners; (Fisch et al., 2017) proves that pools that allocate its mining rewards following geometric distribution have greater utility than those of both Proportional and Pay-Per-Last-N-Shares; (Cong et al., 2021) proposes that the equilibrium of a cooperative mining game depends on the fee charged by the pool manager, as well as the number of players in the coalition. We also highlight that interactions within a Blockchain network can be of many types, not just cooperative. For instance, (Liu et al., 2019) assess the non-cooperative nature of individual mining, constructs the extensive-form game (where each block includes a fork chain selection), the Stackelberg game (where the game includes leaders and followers of strategies) and the stochastic game (Han et al., 2011). In particular, the stochastic game is similar to the one studied in this paper.

Game theoretic applications that are non-cooperative have a wide variety of applications, such as preventing a Pool Blocking Withholding attack (PBWH) (Courtois and Bahack, 2014) and selfish mining, analysis of fork chain and cartel strategies and manipulation of data verification system (Laszka et al., 2015). Akin to this work, (Dimitri, 2017) utilises a non-cooperative game to analyse the interaction between miners to determine their decision to invest or leave a certain pool. The literature indicates a number of issues in Bitcoin/Blockchain mining that can be mitigated using Game Theory, but the majority of works study non-cooperative contexts. Our work aims to propose a reward sharing method in a cooperative framework (mining pools) aiming at maintaining the miners within the pool with economic incentives not to engage in pool-hoping or other malicious attacks, because the pool itself is sharing rewards with higher value than if said player abandoned it.

3 Nucleolus-based Quota Allocation Model

In the context of this work, we assume a total of \( N^{(T)} = \{1, \ldots, N^{(T)}\} \) players participating in the Proof-of-Work of a Bitcoin network, of which \( N \subseteq N^{(T)} \) are of particular interest since
they can act cooperatively. Furthermore, we set as $\mathcal{N}(R) \subseteq \mathcal{N}(T)$ the remainder players such that $\mathcal{N}(R) = \mathcal{N}(T) \setminus \mathcal{N}$. To ease presentation and without loss of generality, the set of remainder players will be “unified” as a single player called “rivals”, for simplicity. Formally, following the standard cooperative game theory nomenclature, we hereinafter refer to $\mathcal{N}$ as the grand coalition and the set of all coalitions possible to be assembled as its powerset $\mathcal{P}(\mathcal{N})$. We also refer to $q_S$ as the probability that a coalition $S \in \mathcal{P}(\mathcal{N})$ successfully mines the next block and directly relates it to the hashpower ($h_n$) of each player $n \in S$ as follows:

$$q_S = \sum_{n \in S} h_n + f(S), \quad \forall \; S \in \mathcal{P}(\mathcal{N}), \quad (1)$$

where $f : \mathcal{P}(\mathcal{N}) \to \mathbb{R}_+$ is a set function that maps the relative increase (benefit) in the mining likelihood of coalition $S \in \mathcal{P}(\mathcal{N})$ by playing as a group.

It worth highlighting that 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 mining result independent from the previous one, with the same probability $q_S$. Therefore, in this work, for a given coalition $S \in \mathcal{P}(\mathcal{N})$, we denote by $\gamma_S \in \{0, 1\}$ a binary random variable indicating the success or failure in mining a block, modelled by a Bernoulli distribution with probability $q_S$, i.e., $\gamma_S \sim$ Bernoulli($q_S$). In the context of a sequence of $T$ blocks ahead, the joint probabilistic representation thus becomes a Binomial distribution, due to be a series of independent Bernoulli trials (Casella and Berger, 2006). For nomenclature purposes, hereinafter, we refer to the collection $\tilde{\gamma}_S \triangleq \{\gamma_{t,S}\}_{t=1}^{T}$ as the stochastic process representing the mining dynamics of a coalition $S \in \mathcal{P}(\mathcal{N})$ in the blocks $t \in \{1, \ldots, T\}$, and to $\tilde{\Pi}_S \in \{0, \ldots, T\}$ as the random variable indicating the number of blocks mined by coalition $S \in \mathcal{P}(\mathcal{N})$ within the next $T$ blocks, i.e.,

$$\tilde{\Pi}_S \equiv \sum_{t=1}^{T} \tilde{\gamma}_{t,S} \sim \text{Binomial}(T, q_S), \quad \forall \; S \in \mathcal{P}(\mathcal{N}). \quad (2)$$

Following the discussion in Section 2, the total income $(\bar{\pi} \triangleq \{\bar{\pi}_t\}_{t=1}^{T})$ from mining a block $t \in \{1, \ldots, T\}$ is given by

$$\bar{\pi}_t = \pi^{(f)} + \bar{\pi}^{(v)}_t, \quad \forall \; t \in \{1, \ldots, T\}, \quad (3)$$

reflecting the payment dynamics of the network. In (3), $\bar{\pi}^{(v)} \triangleq \{\bar{\pi}^{(v)}_t\}_{t=1}^{T}$ denotes the (uncertain) variable income from mining the blocks $t \in \{1, \ldots, T\}$ and $\pi^{(f)}$ indicates the (deterministic) income per block. On the one hand, for consistency in analysis and without loss of generality, in this work, the latter is assumed fixed and time-invariant within the maturity of analysis, since the length of blocks ahead is considered sufficiently small for the halving process to take place. More precisely, the Bitcoin network adjusts the “guessing” difficulty of the Proof of Work so that each block takes a fixed time of completion, which is currently set as (approximately) 10 minutes. Therefore, any reasonable time-span analysis is sufficiently small to maintain $\pi^{(f)}$ unchanged by the halving process or the adjusted network difficulty to express any considerable changes. On the other hand, the variable income per block is a random variable and comprises the sum of all the fees paid by each individual that has a transaction added to the block mined. Formally, the profit of a coalition $S \in \mathcal{P}(\mathcal{N})$ in the next $t \in \{1, \ldots, T\}$ blocks can be written as follows:

$$R(\bar{\pi}, \tilde{\gamma}) = \sum_{t=1}^{T} (\tilde{\pi}_t \tilde{\gamma}_{t,S} - C_S), \quad \forall \; S \in \mathcal{P}(\mathcal{N}), \quad (4)$$

where $C_S$ denotes the total cost a coalition $S \in \mathcal{P}(\mathcal{N})$ incurs in trying to guess the next block $t \in \{1, \ldots, T\}$, which might be composed by operational (e.g., electrical energy consumption),
The other hand, by correctly guessing the header of a block, the miner consequently receives the block income \( \tilde{\pi}_t \) of the nucleolus of a coalitional game. For this purpose, let \( \Phi \) be an allocation function, which represents the core of the cooperative game. The main objective of this work is to propose a quota-allocation methodology based on the concept of the nucleolus of a coalitional game. For this purpose, let \( \Phi \) to represent the core of the cooperative game. To characterize this functional, from a single-player perspective, note that the nature of the business entails a long period of absence in the mining cash flow interrupted by a relatively high (with respect to the total cost of mining) income whenever the player guess the header. As a consequence, we argue that a risk-neutral measure might be myopic to appropriately account for this fat-tailed negatively-skewed profit distribution. Therefore, in this work, we leverage on the structural properties of the quantile-based risk functional, named Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev, 2002; Staino and Russo, 2020), to devise an appropriate risk-averse certainty-equivalent measure for the game. Roughly speaking, for a given percentile \( \alpha \in (0,1) \), the CVaR can be interpreted as the average of the \( (1 - \alpha) \) least profitable scenarios, acting as a smooth metric for a worst-case value\(^6\). By virtue of this characteristic, a (convex) combination between the expected value and the CVaR of the coalition profit (4) is considered in this work aiming at balancing the expected (long-term) profit with the low-quantile impact (Fanzeres et al., 2015; Torraca and Fanzeres, 2021).

Formally, the risk-averse certainty-equivalent measure for the game and the respective characteristic (value) function \( v : \mathcal{P}(N) \to \mathbb{R} \) of a coalition \( S \in \mathcal{P}(N) \) is stated as follows

\[
v(S) = \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_S) \right) = \lambda \text{CVaR}_\alpha \left( R(\tilde{\pi}, \tilde{\gamma}_S) \right) + (1 - \lambda) \mathbb{E} \left[ R(\tilde{\pi}, \tilde{\gamma}_S) \right], \quad \forall S \in \mathcal{P}(N). \tag{5}\]

In (5), \( \lambda \in [0,1] \) outlines the general risk aversion embedded within the game. More specifically, \( \lambda = 0 \) induces a risk-neutral game and as \( \lambda \to 1 \), the CVaR component is highlighted, leading thus to a more risk averse framework.

Conceptually, a cooperative game can be characterized by the interaction between distinct players seeking for better results (usually in financial terms) by leveraging into some source of mutual synergy and can be simply defined by the pair \((N,v)\) such that \( v(\emptyset) = 0 \) (Barron, 2008; Jezic et al., 2016). One of its key facets is the investigation of fairly allocation methods to spread out the cooperation benefits among associates in order to ensure sufficient conditions for each player to keep cooperating. In technical literature, there exists several known methods for this aforementioned allocation (Barron, 2008; Freire, 2017; Street et al., 2011; Shapley, 1953; Junqueira et al., 2007). One of the most studied and widely used in industry by practitioners is based on the concept of the Core of a cooperative game. Roughly speaking, the Core is defined as the set of all feasible (financial) allocations among associates such that the resulting cooperation is beneficial for all coalition \( S \in \mathcal{P}(N) \) possible to be formed. As a consequence, an allocation within the Core of \((N,v)\) ensures that any individual player or coalition does not have the (financial) incentive to leave the cooperative group (grand coalition).

Formally, let \( x \triangleq \{ x_n \}_{n \in N} \) be an allocation vector. For expository purposes, we consider as an allocation a percentage of the grand coalition profit, as a quota allocation. Following equation (5),

\(^6\) See, for instance, (Street, 2010; Fanzeres et al., 2014; Moreira et al., 2018, 2019) for an extensive discussion on its properties and interpretations.
the Core of \((\mathcal{N}, v)\) is hereby defined as the set
\[
\mathcal{X} = \left\{ x \in \Delta^{\lvert \mathcal{N} \rvert} \left| \sum_{n \in S} x_n R(\tilde{\pi}, \tilde{\gamma}_S) \geq \Phi R(\tilde{\pi}, \tilde{\gamma}_N), \ \forall S \subseteq \mathcal{P}(\mathcal{N}) \right. \right\},
\]
where \(\Delta^{\lvert \mathcal{N} \rvert}\) is the \(\lvert \mathcal{N} \rvert\)-dimensional probability simplex, i.e.,
\[
\Delta^{\lvert \mathcal{N} \rvert} = \left\{ x \in [0, 1]^{\lvert \mathcal{N}\rvert} \left| \sum_{n \in \mathcal{N}} x_n = 1 \right. \right\},
\]
e nsuring thus an efficient quota allocation. On the one hand, we highlight that the game is not structurally balanced, thus the non-emptiness of the Core \((\mathcal{X})\) is not guaranteed, following the Bondareva-Shapley Theorem (Shapley, 1967). On the other hand, for some instances, \(\mathcal{X}\) might also not be a singleton.

In this context, we make use of a follow-on concept of the Nucleolus of the cooperative game \((\mathcal{N}, v)\) to identify an efficient quota allocation within the Core (Schmeidler, 1969; Benedek et al., 2020). The Nucleolus of \((\mathcal{N}, v)\) can be defined as an allocation within the Core that induces the maximum (financial) benefit of acting along with the grand coalition with respect to the highest-valued coalition. Formally, for a given quota allocation \(x\), let \(\epsilon_S(x)\) to define the excess value of \(S \subseteq \mathcal{P}(\mathcal{N})\) with respect to the grand coalition \(\mathcal{N}\), i.e.,
\[
\epsilon_S(x) = \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_S) \right) - \sum_{n \in S} x_n \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_N) \right), \quad \forall S \subseteq \mathcal{P}(\mathcal{N}) \quad (7)
\]

Therefore, the proposed Nucleolus-based quota allocation model is defined as the following two-level system of optimization problems
\[
x^* \in \arg \min_{x \in \Delta^{\lvert \mathcal{N} \rvert}} \left\{ \max_{S \subseteq \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}} \{ \epsilon_S(x) \} \right\}, \quad (8)
\]
Structurally, the objective of (8) is to identify a quota allocation \(x \in \Delta^{\lvert \mathcal{N} \rvert}\) such that the excess \(\epsilon_S(x)\) of the highest-valued coalition \(S \subseteq \mathcal{P}(\mathcal{N})\) is minimized. Note that, if \(\max_{S \subseteq \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}} \{ \epsilon_S(x^*) \} \leq 0\), the financial benefits of acting cooperatively within the pool of miners is higher than all coalitions possible to be formed, thus, there are not enough sufficient financial incentives for the grand coalition to be ceased. As a consequence, each player within the pool of miners is not only highly rewarded due to cooperation, keeping active the fundamental nature of the blockchain network, but also receives a more stable cash flow, avoiding the aforementioned fat-tailed negatively-skewed profit distribution.

From a computational perspective, the two-level system of optimization problems in (8) that defines the Nucleolus-based quota allocation is a non-linear model, thus not suitable for standard optimization methods and algorithms, or commercial solvers. Nevertheless, problem (8) can be conveniently re-written as the following linear programming problem:
\[
(x^*, \epsilon^*) \in \arg \min_{x, \epsilon} \epsilon \quad (9)
\]
subject to:
\[
x_n \geq 0, \quad \forall n \in \mathcal{N}; \quad (10)
\]
\[
\sum_{n \in \mathcal{N}} x_n = 1; \quad (11)
\]
\[
\epsilon \geq \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_S) \right) - \sum_{n \in S} x_n \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_N) \right), \quad \forall S \subseteq \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}. \quad (12)
\]
We highlight that, in (9)–(12), \(\epsilon^*\) recovers highest-valued coalition, i.e., \(\epsilon^* = \max_{S \subseteq \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}} \{ \epsilon_S(x^*) \}\), thus measuring the most critical coalition with respect to the pool stability. Furthermore, if \(\epsilon^* \leq 0\), we ensure cooperative efficiency within the pool of miners.
0, than the Nucleolus-based quota allocation identified ($x^*$) belongs to the Core, i.e., $x^* \in \mathcal{X}$. Otherwise, the Core is empty, thus no allocation can be found such that being in the pool is financially advantageous with respect to some coalition $S \in \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}$ possible to be assembled.

Aiming at analyzing the influence of the optimal quota allocations in this work, for nomenclature purposes, we refer to $\mathcal{P}_S(\Phi)$ as the value of coalition $S \in \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}$ within the mining pool and as $O_S(\Phi)$ the value of $S \in \mathcal{P}(\mathcal{N}) \setminus \mathcal{N}$ acting independently (outside the pool). More precisely,

$$O_S(\Phi) = \Phi\left(R(\bar{\pi}, \bar{\gamma}_N)\right);$$

$$\mathcal{P}_S(\Phi) = \sum_{n \in S} x_n \Phi\left(R(\bar{\pi}, \bar{\gamma}_N)\right).$$

In the next section, we present a set of numerical experiments aiming at studying and evaluating the proposed quota allocation methodology described in this section.

4 Case Studies

In order to illustrate the applicability of the proposed quota allocation method, in this section, we analyze two numerical experiments. We begin (in Section 4.1) with an illustrative example of a 4-player network ($|N^{(T)}| = 4$) of which three players can act cooperatively ($|N| = 3$) and then move to a realistic case in Section 4.2 based on real data of the Bitcoin-refunded Blockchain network of January 2019. We focus on the incremental value over using intuitive allocations (uniform and based on hashpower) and in what extend the relative increase in the mining likelihood by playing as a group influence the pool stability. Lastly, we discuss different methods to allocate rewards under different Core-based metrics, comparing with the proposed Nucleolus-based one. It worth highlighting that, in all numerical experiments conducted in this section, we assume that the collection of variable incomes $\bar{\pi}^{(v)} \triangleq \left\{\bar{\pi}_t^{(v)}\right\}_{t=1}^T$ are block-wise independent and which one follows a log-Normal distribution with parameters estimated using historical data of bitcoin prices during the first semester of 2018.

4.1 4-Player System: Illustrative Example

In this illustrative example, we assume, for expository purposes, that $f(S) = 0, \quad \forall S \in \mathcal{P}(\mathcal{N})$ in equation (1), i.e., null increase in the mining likelihood by acting cooperatively, and set $\alpha = 75\%$.

Table 1 presents the hashpower distribution of each player within the network along with the individual total mining cost per block considered in this case. We assume, for the sake of simplicity, that a coalition total cost is linear in the cost of each player: $C_S = \sum_{n \in S} C_{(n)}$.

<table>
<thead>
<tr>
<th>Player</th>
<th>Hashpower (%)</th>
<th>Cost (BTC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>25.0</td>
<td>1.5</td>
</tr>
<tr>
<td>#2</td>
<td>15.0</td>
<td>0.2</td>
</tr>
<tr>
<td>#3</td>
<td>35.0</td>
<td>2.5</td>
</tr>
<tr>
<td>Rival</td>
<td>25.0</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1  Illustrative example – Hashpower distribution and individual total mining cost per block.

To begin with, we highlight that, under the particular case of a single block-ahead analysis ($T = 1$), risk-neutral game ($\lambda = 0$), absence of mining costs for all coalition possible to be set, i.e.,
\( C_S = 0, \forall S \in \mathcal{P}(N) \) and Player \( #3 \) is merged with the Rival player (i.e., a 3-Player network with zero mining cost and \((h_1, h_2, h_3) = (0.25, 0.15, 0.60)\)), the Nucleolus-based quota allocation is precisely equal to the hashpower distribution: \((x_1, x_2, x_3) = (0.25, 0.15, 0.60)\), as expected. However, as a Rival to the mining pool is introduced and the cost of mining is non-negligible (following Table 1), the quota allocation distribution deviates from the expected. Table 2 displays, for a single-block ahead \((T = 1)\), the Nucleolus-based allocation for different risk-aversion levels along with the excess of the highest-value coalition \((\epsilon^*)\).

<table>
<thead>
<tr>
<th>Player</th>
<th>(\lambda)</th>
<th>0.00</th>
<th>0.25</th>
<th>0.50</th>
<th>0.75</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.314</td>
<td>0.298</td>
<td>0.000</td>
<td>0.388</td>
<td>0.357</td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>0.324</td>
<td>0.426</td>
<td>1.000</td>
<td>0.000</td>
<td>0.047</td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>0.363</td>
<td>0.276</td>
<td>0.000</td>
<td>0.612</td>
<td>0.595</td>
<td></td>
</tr>
<tr>
<td>(\epsilon^*)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.309</td>
<td>0.269</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Illustrative example – Nucleolus-based allocation for different risk-aversion levels \((\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\})\) and the excess of the highest-value coalition \((\epsilon^*)\) for a single-block ahead analysis \(T = 1\).

Firstly, note that, in the risk-neutral setting \((\lambda = 0.0)\), the quota allocation is (roughly) uniform among the three players and, as we increase the risk-aversion level \((\lambda \rightarrow 1)\), it diverges significantly both from the hashpower- and uniform-based allocations. On the one hand, for low risk-aversion levels \((\lambda = 0.25)\), the Nucleolus-based solution tend to reward more Player \#2, which is the most cheapest with lower computational capacity among all three, but for an extreme risk-aversion setting \((\lambda = 0.99)\), the reward distribution tends to benefit more the Player \#3 due to its high hashpower and mining cost. On the other hand, we highlight that for \(\lambda \in \{0.50, 0.75\}\) there exists a coalition \(S \in \mathcal{P}(N)\) that is more profitable to jointly mine outside the pool than in full collaboration within the mining pool, since \(\epsilon^* > 0\) (see column five and six, last line). In other words, that means that mining in the pool is actually financially worst for those two risk profiles for some coalition \(S \in \mathcal{P}(N)\).

<table>
<thead>
<tr>
<th>Coalition</th>
<th>(\lambda)</th>
<th>(O_S(\Phi))</th>
<th>(P_S(\Phi))</th>
<th>(\epsilon_S)</th>
<th>(\lambda)</th>
<th>(O_S(\Phi))</th>
<th>(P_S(\Phi))</th>
<th>(\epsilon_S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(#1)</td>
<td>0.50</td>
<td>0.062</td>
<td>0.000</td>
<td>0.062</td>
<td>0.75</td>
<td>-0.719</td>
<td>-0.720</td>
<td>0.001</td>
</tr>
<tr>
<td>(#2)</td>
<td>0.50</td>
<td>0.738</td>
<td>0.489</td>
<td>0.249</td>
<td>0.75</td>
<td>0.269</td>
<td>0.000</td>
<td>0.269</td>
</tr>
<tr>
<td>(#1, #2)</td>
<td>0.50</td>
<td>0.799</td>
<td>0.489</td>
<td>0.309</td>
<td>0.75</td>
<td>-0.451</td>
<td>-0.720</td>
<td>0.269</td>
</tr>
<tr>
<td>(#3)</td>
<td>0.50</td>
<td>-0.312</td>
<td>0.000</td>
<td>-0.312</td>
<td>0.75</td>
<td>-1.406</td>
<td>-1.314</td>
<td>-0.272</td>
</tr>
<tr>
<td>(#1, #3)</td>
<td>0.50</td>
<td>-0.249</td>
<td>0.000</td>
<td>-0.249</td>
<td>0.75</td>
<td>-2.125</td>
<td>-1.854</td>
<td>-0.271</td>
</tr>
<tr>
<td>(#2, #3)</td>
<td>0.50</td>
<td>0.422</td>
<td>0.489</td>
<td>0.067</td>
<td>0.75</td>
<td>-1.139</td>
<td>-1.134</td>
<td>-0.005</td>
</tr>
<tr>
<td>(#1, #2, #3)</td>
<td>0.50</td>
<td>0.489</td>
<td>0.489</td>
<td>0.000</td>
<td>0.75</td>
<td>-1.854</td>
<td>-1.854</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 3 Illustrative example – Coalition value within \((P_S(\Phi))\) and outside \((O_S(\Phi))\) the mining pool for \(\lambda \in \{0.50, 0.75\}\).

Table 3 illustrates each coalition value within \((P_S(\Phi))\) and outside \((O_S(\Phi))\) the mining pool for \(\lambda \in \{0.50, 0.75\}\). Note that Player \#2 is the one that introduces the highest value to the cooperative pool, thus also being the most critical agent for the pool stability. In fact, on the one hand, for \(\lambda = 0.50\) (columns 2 up to 5 in Table 3) the value of Player \#2 by mining on its own is roughly
as higher as the most-valued coalition $S \in \mathcal{P}(N)$, namely coalition \{#1, #2\}, and roughly 43% higher than the grand coalition. On the other hand, for $\lambda = 0.75$ (columns 6 up to 9 in Table 3), Player #2 is the only coalition to hold a positive value, including the grand coalition. It is important to highlight that the most “powerful player”, i.e., the one with the highest hashpower (Player #3), is in fact the agent that leads to the worst values due to its intrinsic mining cost. As a consequence, even by diluting its mining cost within the grand coalition and significantly increasing the pool probability to mine the next block is not sufficient to financially overcome the highest-valued coalition (coalition \{#1, #2\} for $\lambda = 0.50$ and coalition \{#2\} for $\lambda = 0.75$).

We follow on the analysis of this illustrative example by considering a multi-block setting with $T = 6$, i.e., the next 6 blocks (or one-hour ahead), with the same setup as in Table 1. Note that, in this case, the probability to mine shifts from a Bernoulli to a Binomial distribution, thus evaluating the influence of a combinatorial multi-period mining. Table 4 shows the nucleolus-based allocation for different risk-aversion levels ($\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}$) and the excess of the highest-value coalition ($\epsilon^*$) in this multi-period analysis.

<table>
<thead>
<tr>
<th>Player</th>
<th>$\lambda$</th>
<th>0.00</th>
<th>0.25</th>
<th>0.50</th>
<th>0.75</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td></td>
<td>0.314</td>
<td>0.291</td>
<td>0.260</td>
<td>0.214</td>
<td>0.141</td>
</tr>
<tr>
<td>#2</td>
<td></td>
<td>0.324</td>
<td>0.331</td>
<td>0.340</td>
<td>0.355</td>
<td>0.377</td>
</tr>
<tr>
<td>#3</td>
<td></td>
<td>0.363</td>
<td>0.378</td>
<td>0.400</td>
<td>0.431</td>
<td>0.482</td>
</tr>
<tr>
<td>$\epsilon^*$</td>
<td></td>
<td>0.000</td>
<td>-1.64</td>
<td>3.28</td>
<td>4.92</td>
<td>6.56</td>
</tr>
</tbody>
</table>

Table 4 Illustrative example – Nucleolus-based allocation for different risk-aversion levels ($\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}$) and the excess of the highest-value coalition ($\epsilon^*$) for a multi-period analysis $T = 6$.

Generally speaking, note that, similar to the single-block ahead case (Table 2), the allocation in the multi-period, risk-neutral setting still follows the uniform-based quota distribution as shown in the second column of Table 4. Furthermore, the allocations for the risk-averse setting (columns 3 up to 6 of Table 4), also moves away from the “intuitive” uniform- or hashpower-based allocations. Finally, as $\lambda \to 1$, the grand coalition value increases non-negligible with respect to the highest-valued coalition as $\epsilon^*$ becomes more negative (last line of Table 4). This last result can be seen as a direct influence from the shift of Bernoulli to Binomial distribution in the probability to mine.

In order to enhance this analysis, Figure 1 and Figure 2 depicts the cumulative probability distribution function of Player #1’s profit by acting without and cooperatively within the pool for the single-block ($T = 1$) and multi-block ($T = 6$) settings, respectively, under the risk-neutral ($\lambda = 0.00$) quota allocation.

Figure 1 displays that Player #1 mines by itself only 25% of the time and 75% of the time while is acting cooperatively within the pool, highlighting the aforementioned fat-tailed negatively-skewed cash flow structure (Derks et al., 2018). In fact, even for the single-block ($T = 1$) setting, the benefits of cooperation from the viewpoint of financial sustainability is evident since, although the player collects a smaller contribution while in the pool, it is likely that it will experience more blocks-ahead with a positive cash flow. This effect is magnified in the six-blocks ahead setting (Figure 2). The Bernoulli trial for repeated draws shifts to a Binomial distribution, and it shows a set of 6 “jumps”. Those correspond to the probabilities that Player #1 mines at least 1 time, 2 times and so on until all 6 blocks. Therefore, by acting cooperatively within the pool in this illustrative example, Player #1 has a positive net income almost surely, while outside the pool, there is a

7 For expository purposes, we illustrate only the Player #1’s profit under the risk-neutral ($\lambda = 0.00$) setting, but highlight that for both Player #2 and Player #3 a similar pattern is observed, as well as for different levels of risk-aversion.
considerable chance (roughly 20%) that the player will not mine any block, thus considerably losing money in the mining process.

Next, we present a visual representation of the Core of the game\(^8\) for the different risk-aversion levels \(\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}\) considered in this example, both for the single-block (Figure 3) and multi-block (Figure 4) settings. We also include the hashpower-based allocation (referred to as Relative hp) for comparability purposes. With respect to the single-block ahead setting (Figure 3), note that the Core set is a singleton regardless of the risk-aversion level. More specifically, in this setting, due to its “one-shot” nature, the total reward obtained by the mining pool is the same as the sum of each player individual profit as if they were mining by themselves, thus only a single quota allocation solution is feasible such that satisfies all the Core constraints. We emphasize that, for \(\lambda \in \{0.50, 0.75\}\), the set is empty, as previously discussed, following the analysis of Table 2 and Table 3.

Nevertheless, in the multi-block setting (Figure 4), the Core set become distinguishable larger and increases with the risk-aversion level. In fact, we argue this pattern is observed since, as we increase the risk-aversion level, the impact of the \((1 - \alpha)\) worst case profit values (measured by the CVaR) is prominent, thus inducing to more conditions under which the miners are better off as collaborating within the pool. It is also worth noting that the hashpower-based allocation (point of Relative hp) is outside the Core for \(\lambda \in \{0.00, 0.25\}\), which reinforces the need for an effective quota allocation methodology (as the one proposed in this work), since the intuitive solution of sharing by computation power may induce a coalition instability.

\(^8\) Without loss of generality, the representation is presented as a projection in the Player #1 × Player #3 quota allocation space.
Fig. 2  Illustrative example – Cumulative probability distribution function of Player #1’s profit by acting without and cooperatively within the pool for the multi-block ($T = 6$), risk-neutral ($\lambda = 0.00$) setting.

Finally, we stress the multi-block setting and study the impact of a relatively high number of blocks ahead in the quota allocation of this illustrative example. We assume a 24 hours operation ($T = 144$ blocks) and analyze the optimal allocations for different risk profiles (Table 5) and the visual representation of the Core of the game (Figure 5). It worth highlighting that, by increasing the amount of blocks ahead, the Nucleolus-based solution tend to identify allocations close to the uniform-based one (see columns 2 up to 6 of Table 5 with comparison to Table 2 and Table 4). Furthermore, in this setting, the value of the mining pool significantly increases with respect to the highest-valued coalition, indicating that acting as a cooperative pool mitigates the miners risks, measured by the certainty-equivalent measure (5) considered in this work.

<table>
<thead>
<tr>
<th>Player</th>
<th>$\lambda$</th>
<th>0.00</th>
<th>0.25</th>
<th>0.50</th>
<th>0.75</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.314</td>
<td>0.314</td>
<td>0.314</td>
<td>0.314</td>
<td>0.313</td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>0.323</td>
<td>0.323</td>
<td>0.323</td>
<td>0.323</td>
<td>0.322</td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>0.362</td>
<td>0.363</td>
<td>0.363</td>
<td>0.363</td>
<td>0.365</td>
<td></td>
</tr>
<tr>
<td>$\epsilon^*$</td>
<td></td>
<td>0.00</td>
<td>-9.58</td>
<td>-19.2</td>
<td>-28.7</td>
<td>-38.3</td>
</tr>
</tbody>
</table>

Table 5  Illustrative example – Nucleolus-based allocation for different risk-aversion levels ($\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}$) and the excess of the highest-value coalition ($\epsilon^*$) for the stressed multi-block analysis $T = 144$. 

Furthermore, along with the increase in the number of blocks ahead, the Core set still expands with the risk-aversion level, but at a much lower enlargement rate. In fact, based on this illustrative example, we identified that the Core tends to converge (in size) to a singleton as shown in Figure 5. We also observed that, the allocation sharing by computational power (Relative hp in Figure 5) does not belong to the Core of the game for any risk-aversion level. Therefore, we argue that, for a significantly high number of blocks ahead, the reward allocation based on the intuitive computational sharing is not embraced by any risk-aversion profile, thus inducing pool (financial) instability regardless of $\lambda$. Finally, as discussed for the six-blocks ahead setting, increasing the number of blocks ahead creates value to the miners due to the combinatorial characteristic of the multi-block setup, but tends to reduce the amount of stable allocations (those within the Core) to a single one.

### 4.2 Real Case Analysis

In this section, we evaluate and analyze the benefits of the proposed quota allocation methodology in a larger mining pool based on a realistic context. We make use of the distribution of the Bitcoin-refunded Blockchain Network dated from January 29th, 2019\(^9\). Structurally, we assume a mining

\(^9\) All data used to set this realistic case were gathered in the following online references:
We also assume a Dollars to Bitcoin conversion of 3452.00 US$/BTC following the price in January 29th, 2019. We highlight that the online reference no. 4 indicates costs to mine that are not directly related to the computational
pool containing a total of 10 miners (|N| = 10) which were chosen as those that have the lowest hashpower in the network. The idea is to evaluate the benefits of cooperation among the least powerful (in a computational sense) miners and highlight the financial sustainability of these “small players” within the network. Finally, we assume a single rival as a “fictional” player that contains the cumulative hashpower from all the other players in the network. Figure 6 shows the hashpower distribution of each one of the players that can act cooperatively, as well as their total mining cost per block. It is worth noting that, for expository reasons, players #1 – #4 (i.e., the lowest in computational power) are considered as a single player (hereinafter simply called Player #1) with their hashpower and mining costs summed up (since they own almost identical hashpower (%) and cost to mine (BTC)). To ease presentation, we refer to each pool in Figure 6 in an order of hashpower, i.e., Player #2 refers to Pool #5, Player #3 refers to Pool #6, and so on. Following settings in Section 4.1, we also assume α = 75%, a multi-block setting with T = 6 (i.e., an one-hour ahead analysis), and considered that a coalition total mining cost is linear in the cost of each player: $C_S = \sum_{n \in S} {C_{\{n\}}}$, \forall S \in \mathcal{P}(N)$. Furthermore, unless otherwise stated (e.g., in Section 4.2.1), we also assume that $f(S) = 0$, \forall S \in \mathcal{P}(N)$ in equation (1).

Table 6 presents the Nucleolus-based allocation for the real case assuming different risk-aversion levels $\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}$ and the associated excess with respect to the highest-valued coalition ($\epsilon^*$). Firstly, note that under the risk-neutral setting ($\lambda = 0.00$), the Nucleolus-based solution allocates the value of mining within the pool (roughly) evenly between Players #2 – #8 and Players #9 – #10 (due to the similarity within these groups – see Figure 6), and barely allocate Player #1, the least powerful among the network. We further highlight that the highest share is allocated to Player #9 due to its favorable balance between computational power and total mining power that each player has, or the “amount of machines” one player owns. Therefore, the cost that is used in this study is multiplied by the hashpower of the player, thus producing a cost estimate that takes into account its computational power.
cost. Secondly, as risk aversion is included in the analysis (for \( \lambda \in \{0.25, 0.50, 0.75\} \)), we observe a relative gain in value with respect to the highest-valued coalition. Generally speaking, note that Player #9 still has the highest allocation share among all players within the pool, but Player #10 observes a significant decrease as the risk-aversion level increases. This pattern is due to the high total mining cost of Player #10, despite its high computational power. We also observe null quota allocations (e.g., for Player #1 in all \( \lambda \in \{0.25, 0.50, 0.75\} \) and Player #4 and Player #8 for \( \lambda = 0.75 \)). This result can be interpreted as a way out given by the pool for negative-valued players in return of their computational power. To further analyse this result, Table 7 presents the individual value within (\( P_S(\Phi) \)) and outside (\( O_S(\Phi) \)) the mining pool for \( \lambda = 0.75 \). Note that the Players \{#1, #4, #8\} have negative value, thus (from an individual perspective) it is financially sustainable for being on the pool rather than by their own. Finally, we observe that for the extreme risk-averse case (\( \lambda = 0.99 \)) the Core of the game is empty.

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
<th>( \epsilon^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.005</td>
<td>0.053</td>
<td>0.058</td>
<td>0.039</td>
<td>0.055</td>
<td>0.073</td>
<td>0.060</td>
<td>0.065</td>
<td>0.315</td>
<td>0.277</td>
<td>0.000</td>
</tr>
<tr>
<td>0.25</td>
<td>0.000</td>
<td>0.064</td>
<td>0.060</td>
<td>0.044</td>
<td>0.053</td>
<td>0.090</td>
<td>0.051</td>
<td>0.056</td>
<td>0.331</td>
<td>0.250</td>
<td>-0.053</td>
</tr>
<tr>
<td>0.50</td>
<td>0.000</td>
<td>0.052</td>
<td>0.103</td>
<td>0.065</td>
<td>0.096</td>
<td>0.063</td>
<td>0.104</td>
<td>0.043</td>
<td>0.287</td>
<td>0.188</td>
<td>-0.030</td>
</tr>
<tr>
<td>0.75</td>
<td>0.000</td>
<td>0.183</td>
<td>0.199</td>
<td>0.000</td>
<td>0.158</td>
<td>0.032</td>
<td>0.200</td>
<td>0.000</td>
<td>0.223</td>
<td>0.004</td>
<td>-0.006</td>
</tr>
<tr>
<td>0.99</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.017</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 Real Case – Nucleolus-based allocation for different risk-aversion levels (\( \lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\} \)) and the excess of the highest-value coalition (\( \epsilon^* \)).
Fig. 6 Real Case – Setup of the bitcoin-refunded Blockchain network. The blue bars indicates each player hashpower and the orange line is the cost to mine (in BTC) per block.

<table>
<thead>
<tr>
<th>Players</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_S(\Phi)$</td>
<td>-0.430</td>
<td>0.149</td>
<td>0.161</td>
<td>-0.133</td>
<td>0.002</td>
<td>0.123</td>
<td>-0.205</td>
<td>-0.232</td>
<td>0.881</td>
<td>0.009</td>
</tr>
<tr>
<td>$P_S(\Phi)$</td>
<td>0.000</td>
<td>0.730</td>
<td>0.793</td>
<td>0.000</td>
<td>0.629</td>
<td>0.129</td>
<td>0.795</td>
<td>0.000</td>
<td>0.888</td>
<td>0.015</td>
</tr>
<tr>
<td>$\epsilon_S$</td>
<td>-0.430</td>
<td>-0.581</td>
<td>-0.632</td>
<td>-0.133</td>
<td>-0.662</td>
<td>-0.006</td>
<td>-1.000</td>
<td>-0.232</td>
<td>-0.006</td>
<td>-0.006</td>
</tr>
</tbody>
</table>

Table 7 Real Case – Individual value within ($P_S(\Phi)$) and outside ($O_S(\Phi)$) the mining pool for $\lambda = 0.75$.

Lastly, for this real case, we also analyze the impact of the two intuitive sharing benchmarks (namely the hashpower- and uniform-based allocation) on the poll (financial) stability. Table 8 presents, for different risk-aversion levels ($\lambda \in \{0.00, 0.25, 0.50, 0.75, 0.99\}$), the excess of the highest-valued coalition given the hashpower- and uniform-based allocation (for expository purposes, referred to as $x^{(hp)}$ and $x^{(un)}$). As Table 8 illustrates, both intuitive sharing methods are not in the Core of the game for any risk profile, which highlights that sharing the pool profits following the individual computational power and evenly among each player does not guarantee financial stability.

4.2.1 Sensitivity Analysis in Mining Likelihood

In order to evaluate the relative increase in mining likelihood by mining as a group, in this section, we assume that the probability that a coalition $S \in 2^V$ mines a block is given by

$$q_S = \sum_{n \in S} h_n + (|S| - 1)\beta, \quad \forall S \in 2^V,$$  \hspace{0.5cm} (15)
Table 8 Real Case – Excess of the highest-valued coalition given the hashpower- \((\epsilon^*(x^{(hp)}))\) and uniform-based \((\epsilon^*(x^{(un)}))\) allocation.

<table>
<thead>
<tr>
<th>Excess</th>
<th>(\lambda)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>(\epsilon^*(x^{(hp)}))</td>
<td>1.18</td>
</tr>
<tr>
<td>(\epsilon^*(x^{(un)}))</td>
<td>6.07</td>
</tr>
</tbody>
</table>

where \(\beta\) indicates the marginal increase in probability by including a new player in the mining pool. Table 9 presents, for values of \(\beta \in \{0.00, 0.01, 0.02, 0.05, 0.07\}\), the correspondent excess of the highest-valued coalition \((\epsilon^*)\) under a risk-averse setting with \(\lambda = 0.25\) and the expected profit of the grand coalition.

Table 9 Real Case – Excess of the highest-valued coalition \((\epsilon^*)\) under a risk-averse setting with \(\lambda = 0.25\) and the expected profit of the grand coalition for values of \(\beta \in \{0.00, 0.01, 0.02, 0.05, 0.07\}\).

<table>
<thead>
<tr>
<th>(\beta)</th>
<th>0.00</th>
<th>0.01</th>
<th>0.02</th>
<th>0.05</th>
<th>0.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\epsilon^*)</td>
<td>-0.053</td>
<td>-0.325</td>
<td>-0.984</td>
<td>-1.390</td>
<td>-3.316</td>
</tr>
<tr>
<td>(E[R(\pi, \gamma)])</td>
<td>15.5</td>
<td>22.7</td>
<td>29.4</td>
<td>49.7</td>
<td>63.2</td>
</tr>
</tbody>
</table>

As expected, following the parameterization considered in (15), by increasing the mining likelihood, the value of the pool (with respect to the highest-valued coalition) also increases, although in a non-linear pace. In fact, moving from \(\beta = 0.01\) to \(\beta = 0.07\) (a 7x increase), the excess \(\epsilon^*\) decreases in the magnitude of 10 times. Moreover, the expected profit within the pool increases in the order of 4 times.

4.2.2 Analysis of Different Core-based Quota Allocations

The main objective of this work is to propose a Nucleolus-based quota allocation methodology to share the profits of a mining pool in a Bitcoin-refunded Blockchain network. Roughly speaking, this methodology makes use of the concept of the Core of a game and seeks to identify a quota allocation that maximize the advantage by acting cooperatively over the highest-valued coalition possible to be set. However, although this allocation methodology ensures maximum stability and is generally associated with the notion of “fairness”, we recognized that multiple different Core-based allocation methods can be used. In fact, some may argue the concept of “fairness” is subjective and a given set of players might deem that the best way to share the income from the pool is by using some particular criterion. In this regard, we intend to study two Core-based allocation alternatives in comparison to the proposed Nucleolus-based one of which a different objective function is considered in (9)–(12), namely

1. OF 1 – Nucleolus-based Allocation - Problem (8);
2. OF 2 – Average Weighting: \(\max_{x \in X} \left\{ \sum_{n \in N} \left( \frac{h_n x_n}{C_n} \right) \right\};\)
3. OF 3 – Thankful Contributors: \(\max_{x \in X} \{x_{#9} + x_{#10}\};\)

More specifically, one the one hand, the concept behind the allocation method OF 2 is to weight the computational power (hashpower) over mining cost of each player in the mining pool, following
a cost/benefit analysis. On the other hand, in OF 3, the two-most most powerful players (in a computational sense) – Player #9 and Player #10 – are “favored” in the allocation process for their significant addition and contribution to the pool. Table 10 presents the quota allocations for the different Core-based methods considered in this section and the respective excess of the highest-value coalition ($\hat{\epsilon}$). For expository reasons, we focus on a risk-aversion level of $\lambda = 0.5$. Firstly, considering the sharing method OF 2, both Player #2 and Player #9 increase significantly its allocated quota, due to their favorable cost/benefit ratio; and, on the opposite direction, Player #4, Player #5, Player #7, and Player #8 reduce their share for similar reasons. Furthermore, since OF 3 focus on maximizing the reward to the most powerful players (Player #9 and Player #10), the total allocation in both players sums up to roughly 75% of the pool profits. Interesting to note, nevertheless, that, although Player #10 is directly accounted for in the objective function, its quota allocation remains the same in all three sharing methods due to a relatively high mining cost. Finally, we highlight that, unlike the Nucleolus-based allocation (OF 1), both Average Weighting (OF 2) and Thankful Contributors (OF 3) allocations induce a null excess (i.e., $\hat{\epsilon} = 0$), which might lead to a higher pool instability.

<table>
<thead>
<tr>
<th></th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
<th>$\epsilon^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF 1</td>
<td>0.000</td>
<td>0.052</td>
<td>0.103</td>
<td>0.065</td>
<td>0.096</td>
<td>0.063</td>
<td>0.104</td>
<td>0.043</td>
<td>0.287</td>
<td>0.188</td>
<td>-0.030</td>
</tr>
<tr>
<td>OF 2</td>
<td>0.000</td>
<td>0.098</td>
<td>0.106</td>
<td>0.014</td>
<td>0.036</td>
<td>0.058</td>
<td>0.021</td>
<td>0.023</td>
<td>0.457</td>
<td>0.183</td>
<td>0.000</td>
</tr>
<tr>
<td>OF 3</td>
<td>0.000</td>
<td>0.047</td>
<td>0.051</td>
<td>0.014</td>
<td>0.036</td>
<td>0.058</td>
<td>0.021</td>
<td>0.023</td>
<td>0.562</td>
<td>0.183</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 10 Real Case – Quota allocations for the different Core-based methods considered in this section and the respective excess of the highest-value coalition ($\epsilon^*$) for a risk-aversion level of $\lambda = 0.5$.

An alternative to tackle this latter point is to combine both the Nucleolus-based allocation methodology with one of the two alternatives methods, Average Weighting (OF 2) or Thankful Contributors (OF 3). More precisely, the following two-step nested procedure can be applied.

1. Solve the linear programming problem (9)–(12) and store $\epsilon^*$;
2. Replace the objective function (9) by one of the two alternatives (i.e., $\sum_{n \in N} \left( \frac{h_{9,n}}{x_{9,n}} \right)$ or $x_{#9} + x_{#10}$), set in equation (12) that

$$\epsilon^* \geq \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_S) \right) - \sum_{n \in S} x_n \Phi \left( R(\tilde{\pi}, \tilde{\gamma}_N) \right), \quad \forall S \in \mathcal{P}(N) \setminus N,$$

solve the resulting linear programming problem and collect the optimal allocation $x^*$.

Table 11 presents the quota allocation based on the two-step nested procedure discussed for both methods Average Weighting (OF 2*) or Thankful Contributors (OF 3*) and the respective (a priori fixed) excess of the highest-value coalition ($\epsilon^*$) for a risk-aversion level of $\lambda = 0.5$. Note that the resulting quota allocation based on the two-step nested procedure differs from the standard Nucleolus-based allocation presented in Table 10 and follows a similar disposal as its equivalent in OF 2 and OF 3, although with an excess strictly negative and minimum, leading thus to a better pool stability. We argue, therefore, that this methodology provides a set of allocations the both minimize the excess of the highest-valued coalition ($\epsilon^*$) whilst ensuring a distribution akin to a pool’s philosophy.
Players

<table>
<thead>
<tr>
<th></th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF 2*</td>
<td>0.000</td>
<td>0.094</td>
<td>0.102</td>
<td>0.017</td>
<td>0.040</td>
<td>0.062</td>
<td>0.025</td>
<td>0.026</td>
<td>0.442</td>
<td>-0.030</td>
</tr>
<tr>
<td>OF 3*</td>
<td>0.000</td>
<td>0.051</td>
<td>0.055</td>
<td>0.017</td>
<td>0.040</td>
<td>0.062</td>
<td>0.025</td>
<td>0.026</td>
<td>0.531</td>
<td>0.187</td>
</tr>
</tbody>
</table>

Table 11 Real Case – Quota allocations based on the two-step nested procedure for both methods Average Weighting (OF 2*) or Thankful Contributors (OF 3*) and the respective (a priori fixed) excess of the highest-value coalition ($\epsilon^*$) for a risk-aversion level of $\lambda = 0.5$.

5 Conclusions

In this work, we propose an Nucleolus-based quota allocation model to share the rewards of a mining pool in a Bitcoin-refunded Blockchain network. This cooperative game-theoretic model takes into account the stochastic nature of the business and considers as game value function a risk-averse functional as the convex combination between the expected reward in mining with the risk associated in such activity, the later characterized by the Coherent Risk Measure known as the Conditional Value at Risk.

Two numerical experiments were conducted to illustrate the applicability of the proposed quota allocation methodology: (i) one based on a small, illustrative 4-player network; and (ii) one derived from real data of the Bitcoin network. With respect to the former, we numerically show that the value of acting cooperatively in the pool increases as more blocks-ahead is considered, due to the intrinsic combinatorial nature of the problem, thus revealing that a mining pool might be stable in the long-run. Furthermore, we empirically observe that the intuitive allocation sharing methods (uniformly and by computational power) are not usually within the Nucleolus of the game, justifying thus the need for studies to obtain a quota sharing methodology to distribute the rewards so that the pool considers fair and leads players to remain mining. With respect to the latter (real-case) experiment, we evaluated the relative increase in the mining likelihood by playing as a group, and discuss and numerically analyze a nested procedure based on the Nucleolus-based allocation seeking for higher “fairness” in the sharing of the pool rewards.

Ongoing research includes considering a block-dependent representation of the mining probability as well as a game value functional that handles uncertainty within a robust optimization framework. We also highlight that it is of importance to take into consideration in the model a delay in communication between the players in order to better characterize the physical aspects of the Bitcoin network.

References

Quota-Allocation for Bitcoin-Refunded Blockchain Network


