Incentive Assignment in Hybrid Consensus Blockchain Systems in Pervasive Edge Environments

Yaodong Huang, Yiming Zeng, Fan Ye, Yuanyuan Yang

Abstract—Edge computing is becoming pervasive in our daily lives with emerging smart devices and the development of communication technology. Smart devices with various resources make data transactions prevalent over edge environments. To ensure such transactions are unmodifiable and undeniable, blockchain technology is introduced into edge environments. In this paper, we propose a hybrid blockchain system to enhance the security for transactions and determine the incentive for miners in edge computing environments. We propose a Proof of Work (PoW) and Proof of Stake (PoS) hybrid consensus blockchain system utilizing the heterogeneity of devices to adapt to the characteristic of edge environments. We raise the incentive assignment problem for a fair incentive to PoW miners. We formulate the problem and propose an iterative and another heuristic algorithm to determine the incentive that the miner will receive for a new block. We further prove that the iterative algorithm can obtain global optimal results. Simulation and experiment results show that our proposed algorithm can give a reasonable incentive to miners under different system parameters in edge blockchain systems.

Keywords—Pervasive edge computing, Hybrid blockchain, Proof of Work, Proof of Stake, Incentive mechanism

1 Introduction

The arriving 5G networks aim at providing low-latency, high-throughput, and energy-saving computing to a massive number of devices. Thanks to the backbone technology, edge computing is becoming increasingly crucial to enhance the quality of service for thriving smart edge devices. Such devices like phones, IoT sensors, or even vehicles offer an immense amount of data, which can be shared and transferred among different clients. With the abundance of devices and data, edge computing can process data locally without the involvement of the cloud or other centralized services. New business models have emerged to provide paid information services to users for income. An example is "We media", where data producers sell content like video clips or texts to interested customers to make money.

Consider a situation where data producers have for-profit content for sharing and trading. Other users may want to access such content and pay for them. The subscriptions allow paid users to access corresponding content quickly and securely while denying unpaid users from obtaining them. Most current solutions require a trusted third party to manage such content and subscriptions. For instance, Gumroad [1] provides services for data producers to sell digital content directly to consumers. Although considerable amounts of text, audio, and video content are sold on these platforms, there are still adverse events [2], mostly related to security, trust, and privacy concerns.

In edge environments, micro-access control and micro-payment transactions provide fast identity verification and data accessing without trusted third parties. This allows data to be traded locally on a relatively small scale. There are two kinds of typical cases using data sharing and trading in such environments. The first case is the data sharing and payment processing between peer edge devices in a limited range. For example, vehicles can sell pictures and road condition information directly to other vehicles without using a cloud-based backend platform [3], or IoT devices producing real-time sensing data and can provide sensing-as-a-services for risk management [4]. The second case is digital trading between two organizations on a limited scale. The history records are crucial to help manage the trading time, payment, and inventory. Among these cases, security and trust must be improved to make sure the transactions are unmodifiable and undeniable. The private blockchain system is a solution for such micro-access control and micro-payment management, where users can directly manage subscription payments and data delivery in edge environments, helping all parties in a distributed manner.

Recently, blockchain technologies used in cryptocurrencies, like Bitcoin [5] or Ethereum, have drawn much attention from both academics and industries. The two major characteristics of blockchain are distribution and security. First, the blocks are distributively stored among all nodes in the network. Each block serves as a ledger storing some transactions. For the blockchain system, each block contains a hash value pointing to the previous block to form a chain. Using the content of blocks can trace back the historical data and reconstruct the current status. Second, the transactions and blocks in the blockchain system are easy to validate

- *Y. Huang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794.*
  E-mail: yaodong.huang@outlook.com
- *Y. Zeng, F. Ye, and Y. Yang are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794.*
  E-mail: {yiming.zeng, fan.ye, yuanyuan.yang}@stonybook.edu.

but hard to modify. The chain structure makes sure that any changes on blocks or transactions will affect an entire branch of the chain. In theory, malicious parties must control over a quarter of the total computational power [6] such that they can have a chance to manipulate the information of the blockchain. Blockchain technology improves the efficiency, security, and privacy of transactions in a distributed manner without the help of centralized trusted third parties.

Despite the advantages of blockchain technology in such distributed systems, edge devices are often heterogeneous over resources, especially storage and energy. Maintaining the security of blocks in blockchain systems often requires a tremendous amount of energy and storage space. Such resource requirement is beyond the capabilities of most edge devices with limited resources (e.g., phones, IoT sensors), and will make them less inclined to participate. Meanwhile, some of the devices (e.g, edge servers and vehicles) will have a larger amount of resources to conduct computing-intensive tasks, which may help enhance the security of edge blockchain systems. In our previous work [7], we have discussed the storage assignment problem to support the edge blockchain system. However, how to combine heterogeneous devices to design an effective private blockchain system that all devices can participate, assigning fair incentives and improving security remain a challenging problem.

In this paper, we introduce a consensus-hybrid Proof of Work (PoW) and Proof of Stake (PoS) blockchain system in edge environments. Although the consensus-hybrid blockchain is not a new concept, we discuss the applicabilities in edge computing environments and provide a private blockchain coordinating resource-limited and resourceful edge devices to improve efficiency and security. This paper focuses on the assignment of incentives to users using different forms of consensus for new blocks in the edge blockchain system. We propose the incentive assignment problem to determine how much incentive is given to PoW miners for mining a new block. A Stackelberg game is formulated to describe the incentive assignment problem and we propose an iterative algorithm and a heuristic algorithm to solve it. We also prove that the iterative algorithm can converge and achieve global optimal results. Small scale experiments and extensive simulations show our proposed algorithms can work well on real edge scenarios and offer an appropriate incentive to PoW miners under different settings of hybrid blockchain system parameters.

We make the following contributions in this paper.

- We introduce a PoS and PoW hybrid consensus blockchain system. We summarize the different consensus and how to apply the hybrid blockchain for the heterogeneity of edge devices. We then adapt the PoW and PoS consensus for the transactions in pervasive edge computing and encourage the participation of both resource-limited and resource-rich devices.
- We propose a novel incentive assignment mechanism to determine the incentive for a new block for miners in the edge blockchain system. We raise the problem to give corresponding PoW miners a fair incentive and formulate it into a two-stage Stackelberg game. We propose an iterative algorithm to solve the problem and offer theoretical analysis to prove it can converge to the global optimal result. We further propose a heuristic algorithm to achieve comparable results with less complexity.
- We implement the hybrid consensus blockchain on real edge devices and conduct a small-scale experiment. It shows that the proposed blockchain works well in pervasive edge environments. We further implement the incentive assignment algorithm and conduct extensive numerical evaluations. The results show that our proposed mechanism can give an appropriate incentive to miners under different system settings.

The rest of the paper is organized as follows. Section 2 discusses some related work on blockchain and edge computing. Section 3 discusses the system overview of the hybrid blockchain. Section 4 presents the design of PoW and PoS blocks and mining processes. In Section 5 we formulate the PoW mining incentive assignment problem and offer the solution. We conduct numerical simulations in Section 6. Finally, we conclude the paper and discuss future work in Section 7.

2 Related Work

Blockchain technology is a novel, distributed system now widely used in cryptocurrencies, like Bitcoin [5], Litecoin [8], and Ethereum [9]. From its original introduction by Satoshi Nakamoto in 2008, related cryptocurrencies have exploded in the following decades. Blockchain technology is based on cryptography theories that can prevent unauthorized changes in such distributed systems. The system holds when the majority of computational power owned by users is honest. A malicious party must control over half of the computational power to modify the historic data stored in the network and must control a quarter of the total computational power to pose any threat to the system over new block generation.

Proof of Work (PoW) and Proof of Stake (PoS) are the most commonly used consensus in blockchain systems. PoW is used in many traditional cryptocurrencies, while PoS is accepted by some cryptocurrencies recently [10]–[12]. In PoW, participating users compete with others over a cryptography problem, usually exhaustively hash contents of the block to make sure the hash value is smaller than a threshold. This process is often time and energy-consuming, receiving much criticism. PoS, on the other hand, achieves the consensus from the historical data of users such as wealth or age, which can be publicly validated. The process reduces energy consumption for new blocks generation, which is a more environmentally friendly way. Some cryptocurrencies like Ethereum have the plan to change to PoS consensus in a near future.

On the contrary to cloud computing which moves the computing to a centralized cloud, edge computing moves the computing work to distributed nodes on the edge of the network. The computing mostly or entirely happens on nodes near to or inside the edge devices [13]. With the increasingly powerful edge smart devices and fast-growing networking technology like 5G and Wi-Fi 6, data sharing among edge devices and clouds creates many novel applications [14]–[16]. Recently, blockchain technology has been introduced for data transaction security for edge environments. Many edge applications such as IoT [17], [18], vehicle network [19], and network function virtualization [20] have applied blockchain to enhance security, privacy, scalability, and robustness. Although blockchain can bring such advantages, limitations of edge environments make it impractical to directly deploy blockchain. Wu et al. [21] discuss the task offloading of mining on mobile edge networks.

Recently, the hybrid blockchain protocol has drawn much attention. The hybrid design brings many new features to the blockchain systems and utilizes advantages from different consensus protocols. Liu et al. [22] propose a fork-free PoW protocol and combine it with PoS to make a hybrid blockchain. Santos et al. [23] discuss measurement of the complexity of different consensus protocols including PoW-PoS hybrid protocol. Gupta et al. [24] studied the time control using PoW blocks and describe an implementation process of the hybrid-blockchain. Hu et al. [25] propose a hybrid blockchain focusing on energy consumption and using software-defined networks to secure the applications for IoT devices. Harvilla et al. [26] propose a blockchain system called PAI, which mitigates the 51% attack utilizing the stakeholders of PoS minters to vote on PoW proposals, and Decred [27] also uses similar methodologies. While these hybrid blockchains receive active participation in public systems, they have certain limitations when applied to the private edge blockchain systems, including the long wait for the confirmed incentives of PoS minters. Some hybrid models combine other consensus protocols as well. Abuidris et al. [28] propose a system combining Proof of credibility and Proof of stake to secure e-voting systems.

To tackle the complicated collaboration in edge networks, the game theory is a promising technique that has been widely adopted in various networks. In [29], the authors consider a D2D communication framework in which the operator of the base station offers incentives to owners of devices to motivate content communication. In [30], a wireless sensor network consisting of many private sensor networks is considered.

3 Background

In this section, we first introduce background information about different consensuses of blockchain systems and how the hybrid consensus blockchain is used in our situations.

3.1 Consensus Mechanisms

3.1.1 Proof of Work

Proof of Work is a well-known consensus mechanism and presented in Bitcoin [5], which grants the privilege to users who solve a computationally intensive math problem. The user needs to hash certain information and a random number so the hash value meets some preset patterns. The user is called **miner** and the process is called **mining**. For instance, Bitcoin miners need to hash the timestamp, hash value of Merkle (a type of tree for transaction data) root, the previous block hash value, current target (indicate difficulty), and a nonce to get a hash value. The hash value must be smaller than a given number (target). The miner can change the hash value by picking a different nonce. The process that finding the nonce thus certain hash value is called mining. The smaller the target is, the harder the mining process will be. Currently, the target is 20 consecutive 0's in the front of the hash value (in hexadecimal form)[1]. The security of the PoW mechanism prevents changing information in the blocks by relating it to a hash value in the chain. Unless one entity controls more than a quarter to half of the total computational power in the network [31], [32], it cannot counterfeit the chain.

3.1.2 Proof of Stake

Proof of Stake consensus mechanism aims at reducing the amount of power consumption. The total amount of energy consumed per year for Bitcoin mining is 78 TWh (7.8×10^{10}kWh)[2]. PoS, on the other hand, is an energy-saving method to reach the consensus to generate new blocks. Users who create PoS blocks are called **minters**, and the process is called **minting**. Unlike PoW mining where miners competing to solve a cryptography problem, the specific minter of the next block is randomly chosen based on the history related factors (e.g, wealth, age, storage). These factors are often assigned as tokens. Recently, PoS gains much attention as a low energy cost alternative over block consensus. Many cryptocurrencies appear based on this concept, e.g., Nxt [10] and Peercoin [11], and Ethereum has plans to move to PoS in the future [33].

Although PoS has advantages over PoW on energy consumption, it has certain drawbacks that prevent it from being widely used. First, due to the low complexity of computation work, working on different chains is less of a computational burden. This may create more branches and some users can work on multiple branches to make more profit [34]. Second, since minters with a larger number of tokens have a higher chance of minting the next block, richer minters will become richer, and poorer will stay poor. This also makes it more vulnerable to the 51% attack in which an entity obtains 51% of the tokens. Note that a node can be both miners and minters, but it should have two different accounts for different consensus. The accounts will not interfere with each other.

3.2 Hybrid Blockchain Design

Hybrid consensus blockchain systems have been discussed in previous work [22], [23]. In these blockchain systems, the chain is made up of two different kinds of blocks, PoS and PoW blocks. The blocks are created by users using different consensus. The blockchain is briefly described in Fig. 1.

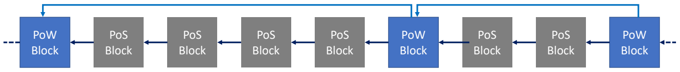


Fig. 1. A brief description of hybrid blockchain. PoS blocks are the majority in the blockchain. PoW blocks are inserted in between PoS blocks.

[1] Data obtained from the hash of recent Bitcoin blocks on https://blockchain.info in March 2021.

[2] Data obtained from https://digiconomist.net/bitcoin-energy-consumption in March 2021.

As we can see from Fig. 1, there are more PoS blocks than PoW blocks. PoS blocks are generated (minted) by the users (minters) with limited resources, which are the majority of users in edge environments. The PoS blocks process and store most of the transactions. These transactions are quickly processed, and a part of the transaction fees are given to the block minters.

PoW blocks are generated (mined) by users (miners) with more resources. The mining process is basically the same as the traditional PoW. The difference in our scenarios is that, instead of the longest chain, they choose the chain that contains all legal transactions and has the most PoW blocks to append the new block. Since the PoS chain may create many branches, PoW blocks can freeze a certain branch (by appending the new block on a certain branch). Other users will continue to work on the frozen branch. Freezing a certain branch will reduce the number and length of branches. Meanwhile, transactions need to be validated by PoW miners. A part of the transaction fees is also given to miners. We discuss the detailed mining process in Section 4.4.

PoW blocks are inserted into the chain in a balanced frequency. A low fraction of PoW blocks will increase the processing time for transactions. A high fraction of PoW blocks will take too much computational power that exceeds the capacities of users. Thus, a balanced frequency of PoW blocks is needed, and an expected ratio between PoS and PoW block, denoted as r, is crucial for the stableness of the blockchain.

4 Blockchain Design

In this section, we discuss the block structures of PoS and PoW blocks used in our system, and introduce the minting and mining process in edge environments.

4.1 Proof of Stake Blocks

The structure of PoS blocks is illustrated in Fig. 2.

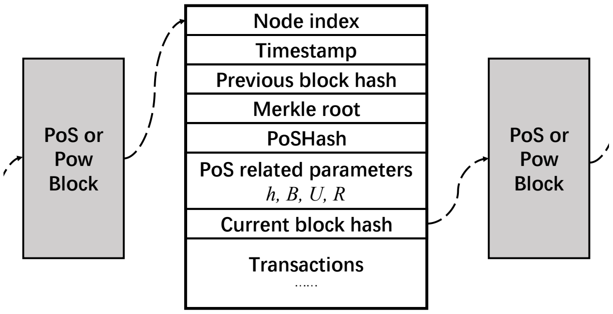


Fig. 2. The structure of a PoS block

The purpose of PoS in the blockchain is to reduce power consumption and encourage users (as minters) with limited resources to participate and process all transactions in edge environments. The PoS blocks record the information for the minting process that other users can validate. A block consists of a header and contents. The header records basic information of the block. Contents record transactions that happen after the previous block generates from a transaction pool. In the header, the timestamp, index, and previous block hash are similar to those in normal blockchain systems. Transactions are encoded in a binary tree of hash values called Merkle tree. The root of the tree, called Merkle root is recorded in the header of the PoS block. PoSHash and PoS related parameters are used to validate whether the block is legal. PoSHash is used to generate and validate corresponding parameters. Expected time amendment B can adjust the expected time between two consecutive PoS blocks. Target value R and hit value h are for other users to verify that the block comes from the minter that has the privilege. Settings of these PoS related parameters are introduced in Section 4.4. Finally, the minter will hash such information in the header and store the hash value into the current block hash entry. The hash value does not need to satisfy certain patterns.

4.2 Proof of Work Blocks

The structure of PoW blocks is illustrated in Fig. 3.

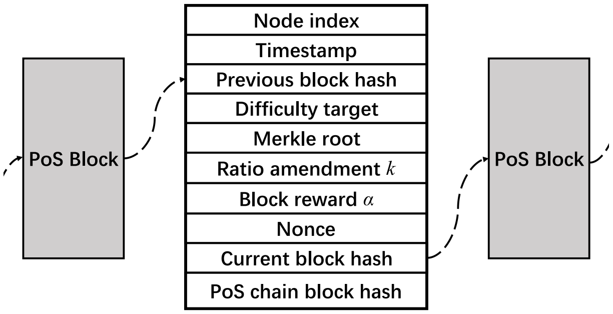


Fig. 3. The structure of a PoW block

PoW block is introduced to enhance the security of the blockchain. Since the hash value must satisfy certain patterns, miners must enumerate a nonce number to make the hash value of the header satisfy such a pattern. Thus, counterfeiting a block costs large time and power consumption. To record such information, the design of the PoW blocks has some major differences from the PoS block. First, it has the difficult target part recoding the pattern of the hash value, which indicates how hard mining the block will be. Second, nonce n must be recorded for other users to validate whether the hash value is correct. Third, ratio amendment k is presented which adjusts the number of PoS blocks between two PoW blocks and makes sure the PoW block generation time is proper. Fourth, the Merkle root design is different from that of PoS blocks, which is not for the transaction in the block but rather the Merkle root in PoS blocks after the previous PoW block. The PoW block, in our design, does not contain any transactions. All transactions are processed in PoS blocks.

The purpose of Merkle root is to validate whether the transaction in a block is changed. A slight change in a transaction will make Merkle root very different, thus will also affect the hash value of a block. In our design, transactions are processed in PoS blocks, and each block has a Merkle root entry. Miners later will mine PoW blocks and want to get part of the transaction fees from these transactions. It will validate Merkle roots in these PoS blocks. The miner will build a new binary from the PoS Merkle root and the new tree root will be stored in the PoW block Merkle tree entry. It can ensure that the transactions are hard to counterfeit, and for the benefit of PoW miners, it will choose honest transactions since it can get the most transaction fees.

Note that the absence of transactions in the PoW block can increase transaction safety. Since PoW blocks do not contain transactions, a counterfeit transaction must be encoded in the PoS block of a malicious minter first. It then needs to be accepted later by PoW miners to be encoded in the Merkle tree of a PoW block. It is too hard to coordinate all parties unless it has more than half of the computational power to counterfeit the PoW chain.

4.3 Minting Process

PoS blocks are generated by the minting process. Minters compare the contribution to the system (i.e., mint more blocks) and randomly select one minter to be granted the privilege to mint the next block. In edge environments, the contribution of users is crucial for block generation and transaction processing. Our design goal is to make sure that a minter that contributed more will have more advantage to mint the next block while still preserve the probability for new minters to participate.

In our previous work [7], we describe a minting process designed for edge environments. Each node i will have a hit value h_i, which is directly calculated from PoSHash and its account number. The target value varies from minter to minter since each minter will have different account numbers. Basically,

$$POSHash(d+1,i) = \text{Hash}[POSHash(d) + Account_i],$$
$$h_i = POSHash(d+1,i) \mod L,$$

where L is the largest possible hit value. $POSHash(d)$ is the POSHash in the previous block, and $POSHash(d+1,i)$ will be the next PoSHash. Each minter will also have a target value R_i based on the stake of the minter. R_i is defined as

$$R_i = U_i t B,$$

where U_i is the token of minter i, t is the time passes from the previous block, and B is the value to control the PoS block generation rate. Minter i will be granted the privilege to mint a new block when it is the first minter that satisfies

$$h_i \leqslant R_i.$$

Since R_i will grow as time passes, eventually R_i will be larger than h_i. Minter with larger U_i will have a higher possibility to be granted the privilege since R_i will grow faster. According to [7], when the expected time for generating a new PoS block is t_0, expectation time amendment B equals $\frac{L}{(N+1)t_0 U}$, where U is the average token of all minters.

4.4 Mining Process

PoW blocks are generated by the mining process. Miners compete with each other to first obtain the correct nonce to make the hash value satisfy certain patterns. The nonce finding process is similar to the traditional blockchain where miners enumerate different numbers of the nonce to get a hash value to satisfy a certain pattern. Despite the nonce finding process, our proposed mining process focuses more on the hybrid blockchain system, which needs to consider the PoS blocks and transactions. Since miners consume more computational power and help validate transactions, they are expected to get more incentives. Meanwhile, a certain number of PoS blocks between PoW blocks is crucial for a stable hybrid blockchain. Thus, the system needs to adjust the hardness and the ratio amendment k.

A miner will get an incentive as α coins when it mines a new PoW block. The determination of the expected coin number α is a game between all miners. Ratio amendment k is a crucial factor in the game and the coordination with the PoS process. Ratio amendment k is the ratio between the PoW block generation period T and expected incentive α for PoW blocks, i.e., $k = \frac{T}{\alpha}$. A longer period between two PoW blocks indicates more PoS blocks in between, i.e., more blocks and transactions need to be validated, and increases computational power cost for miners. If the cost for validation exceeds the revenue a miner can get, no user will participate in mining PoW blocks. Miners will require more coins to compensate if the PoW block generation period is longer. Ratio amendment k makes sure the block generation period and expected incentives are reasonable. We discuss the game and ratio amendment in Section 5.

Note that changing the hardness target can also change the generation period T, i.e., increasing hardness to make miners use a long time to get the right nonce. However, increasing hardness will increase power consumption, which is not abundant in edge environments. Adjusting hardness is reasonable when the total computational power in the system changes, which will affect the efficiency of PoW. We will discuss the hardness target changing in our future work.

4.5 Incentive Assignment

Incentive, such as coins, is given to minters and miners to incentivize the participation. As we discuss above, compared to the minting process, the mining process costs much more resources and increases security. To get compensation for their work and encourage more users to participate in mining PoW blocks, a fair incentive must be given to the miners of PoW blocks. Thus, we design a PoW mining incentive assignment mechanism for PoW-PoS hybrid blockchain. The incentive assignment mechanism determines how many coins a miner gets when it mines a new PoW block. The mechanism considers the participation of the PoW miners as well as the ratio between PoW and PoS blocks. We formulate the problem as a two-stage Stackelberg Game and propose an iterative algorithm to solve it. The detailed formulation and solutions are presented in Section 5.

The incentive of minters also needs consideration. If a minter proposes a block, the block will record the number of coins the minter can get. In this paper, we set that for each block, the minter will get one coin and corresponding transaction fees. However, due to the security issues of the PoS blockchain, the miners will not be able to spend the coin immediately. After the corresponding chain is frozen by the PoW blocks, and the longest chain with most PoW blocks is determined (6 PoW blocks, as used in many PoW blockchain systems), the coin of the PoS block is confirmed, and the minter can use the coin for transactions.

4.6 Security Discussion

PoS blockchain has many advantages, especially regarding energy consumptions. However, attacks are possible on the PoS blockchain system. Introducing the PoW process can improve the security of the blockchain system running on the edge environments. Here, we discuss some common attacks that the PoS-based system may suffer [35].

4.6.1 Nothing at stake attack

Since PoS consensus does not require much computational resource, a minter can work on multiple branches of the chain thus getting more incentives. It will even encourage minters to work on multiple branches, which will slow the process and create many forks without too much overhead of minters [34]. The hybrid consensus design can mitigate this by inserting PoW blocks into the chain. The PoW consensus in our scenarios encourages miners to select the branch which contains the most honest transactions and append PoW blocks on it. Minters follow the chain having more PoW blocks. PoW process does not have such attacks due to the high computational overhead in the process. Thus, the branches without the participation of miners will be deprecated quickly.

4.6.2 Long range attack

Due to less computational resources needed for the PoS process, the attacker can make up an entire chain branch consisting of fake transactions from a block far away in history. It can get the advantage of a minter that spends its property in the past and wipe out the transaction records in the new chain. Inserting PoW blocks, which can be regarded as checkpoints, can mitigate the impact of the attack. Since PoW blocks are hard to counterfeit, the frequent appearance of PoW blocks in the chain makes it nearly impossible for a minter to create fake PoW blocks in history and make up a longer chain while competing with all other honest PoW miners.

5 Incentive Assignment Formulation and Solution

In this section, we propose a two-stage dynamic Stackelberg game to model the incentive assignment among PoW block miners. The model is based on [36]. We then discuss the equilibrium of the model and show the algorithm to the solutions. The notations used in this section are shown in Table 1.

5.1 PoW Miners Incentive Assignment Formulation

5.1.1 Miners

As we mentioned before, PoW mining needs a lot of computational power. A simple computational power measurement is how many times of hashing a machine can do in a period. For PoW block miners, they devote a part of their computational power to the mining process. The remaining power can be used for other purposes. Each miner wants to gain as much profit as possible. For a miner i, denote the revenue (coin) of mining a PoW block as α, and the profit is defined as follows.

TABLE 1
Notations used in the game formulation

$i \in \mathcal{I}$	Miner ID and the miner set
x_i	The amount of computational power miner i devoted
s_i	The total amount of computational power of miner i
j_i	The total amount of computational power devoted other than miner i
α	The number of coins as incentive
β_i	The unit revenue than miner i can get for purpose other than mining
k	The ratio amendment of PoW process
Z	The hardness factor of PoW process
T	The expected time between two PoW blocks
r	The expected ratio between number of PoS and PoW blocks
L	The largest possible number for h_i
h_i	A hit of node i, $h_i \sim \mathcal{U}(0, M)$
N	The number of PoS minters in the network
B	The expectation time amendment, the value to adjust the time between two PoS blocks
U	The average token number of all minters

$$M_i(\alpha, x_i) = \alpha \mathbf{P}(x_i) + \beta_i(s_i - x_i) - c \tag{1}$$

In (1), x_i is the amount of computational power that miner i is willing to devote for the Proof of Work mining process. s_i is the total computational power for miner i. β_i is the non-mining revenue factor. $\beta_i(s_i - x_i)$ means the revenue that it can gain using the remainder of the computational power. c represents the cost for miners in a specific round. $\mathbf{P}(x_i)$ is the probability for miner i to mine the block, In the mining process,

$$\mathbf{P}(x_i) = \frac{x_i}{x_i + j_i}.$$

Here, $j_i = \sum_i \mathbf{x_{-i}}$, $\mathbf{x_{-i}} = \{x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n\}$, i.e., $\mathbf{x_{-i}} = \mathbf{x} \backslash x_i$.

Note that for the fairness over mining, α is the same for all miners, meaning that no matter which miner mines the block, it will get α coins; β_i can be different for each miner i, since each miner may use the remaining computational power for different purposes.

The problem formulation for miner i to get the most profit is as follows.

$$\max \qquad M_i(\alpha, x_i) \tag{2}$$

$$\text{s.t.} \qquad 0 \leqslant x_i \leqslant s_i. \tag{3}$$

The objective function (2) is the profit for miner i as we address above. Constraint (3) makes sure miner i cannot use more computational power than its capacity.

5.1.2 Virtual system

Since the blockchain system is distributed, there is no central authority that controls the system settings. Every miner will compete over others to get incentives. It is actually a game among miners. For simplicity, we define a virtual system that serves as a leader in the Stackelberg game. The virtual system is not a real entity, instead, it should be regarded as a protocol that every miner agrees with. The equilibrium obtained from the game is a consensus through different miners, and it gives a solution of incentive coins not too high but still profitable for miners.

The goal of the virtual system is to minimize the coins for each block, i.e., everyone prevents other miners to get too many coins for a block. The problem formulation is as follows.

$$\min \quad \alpha \tag{4}$$

$$\text{s.t.} \quad \sum_i x_i \geqslant \frac{Z}{T}, \tag{5}$$

$$T \geqslant r\frac{L}{(N+1)BU}. \tag{6}$$

Objective function (4) is the number of coins for a block α. Constraint (5) makes sure that the total computational power of all miners contributed can satisfy the mining settings of specific hardness and expected period. Z stands for hardness. $T = \alpha k$ indicates the block generation period as we discuss in Section 4.4. Constraint (6) makes sure that the expected PoW period is larger than r times of the expected PoS generation blocks, which we discuss in Section 3.2.

5.2 Game Model and Equilibrium Analysis

We now discuss the two-stage Stackelberg game. The game is defined as follows.

- *Followers*: PoW block miners.
- *Leader*: the virtual system.
- *Strategies*: the virtual system determines the number of coins α and miners determine the amount of computational power x_i to devote.
- *Payoff*: minimize the coin α and maximize the total profit for miners $M_i(x_i)$.

We now analyze the equilibrium derivation of the problem.

Definition 1. Stackelberg Equilibrium*: The outcome $\{x^*, \alpha^*\}$ is the Nash equilibrium of the game G_B, if the following conditions are concurrently satisfied for every miner $i \in \mathcal{I}$ and the virtual system:*

$$M(\alpha, x_i^*) \geqslant M(\alpha, x_i), \forall i \in \mathcal{I},$$
$$\alpha^* \leqslant \alpha.$$

This problem is challenging because the number of coins and the amount of computational power each miner devotes are coupled together. The processes of the interaction between the virtual system and miners are dynamic. To analyze the problem, we separate the process of the game into two different stages. In Stage I, the virtual system presents the expected coin α to miners. In Stage II, miners receive the expected coin α and adjust the computational power to participate. The results are returned to the virtual system. The virtual system and miners adjust the value until equilibrium is reached. This game jointly solves the two problems.

5.2.1 Stage II: Individual miner problem (IMP)

We first address the case in Stage II. The objective for resellers is to maximize their total profit. After getting the expected number of coins α of the leader, the determination of miners is decided as the response for participation. We analyze the existence and uniqueness of the Nash equilibrium in the IMP.

Definition 2. *A computation resource assignment vector $\mathbf{x}^* = (x_1^*, \cdots, x_n^*)$ is the Nash equilibrium of the IMP, if, for each miner $i \in \mathcal{I}$, $M_i(\alpha, x_i, j_i^*) \leqslant M_i(\alpha, x_i^*, j_i^*)$, where $j_i^* = \sum_{l \neq i | l \in \mathcal{I}} x_l^*$.*

We prove the existence of the equilibrium of IMP in Theorem 1.

Theorem 1. *A Nash equilibrium exists and is unique in the game IMP.*

Proof. We investigate followers (miners) first. The strategy space of the miner i is defined as $[0, s_i]$ from the constraint (3), which is non-empty, convex and compact. The utility function (2) of miner i is continuous in $[0, s_i]$.

To prove the concavity of the utility function, we calculate the first and second-order derivation of (2) which are written as follows,

$$\frac{\partial M_i(x_i)}{\partial x_i} = \frac{\alpha j_i}{(x_i + j_i)^2} - \beta_i,$$

and

$$\frac{\partial^2 M_i(x_i)}{\partial x_i^2} = -\frac{2\alpha j_i}{(x_i + j_i)^3} < 0.$$

The second order partial derivation is less than 0, which indicates that the utility function $M_i(x_i)$ is strictly concave about x_i. Accordingly, the Nash equilibrium exists in this non-cooperative IMP [37].

$M_i(x_i)$ is continuous and concave about x_i in the space $[0, s_i]$. It has only one optimal solution for this subproblem. Thus, the equilibrium is unique. □

The optimal strategy of the miner is decided by solving the optimization problem (2)-(3) for x_i, given that α is obtained from the virtual system and using it as the input. To get the optimal solution, for each node i, let

$$\frac{\partial M_i}{\partial x_i} = 0.$$

Since $x_i \geqslant 0$, we can get

$$x_i^* = \sqrt{\frac{\alpha j_i}{\beta_i}} - j_i. \tag{7}$$

After obtaining the devoted computational power of miner i, it is regarded as the input to the virtual system problem (4)-(6) to minimize the number of coins α in Stage I.

5.2.2 Stage I: Virtual system

Now we discuss Stage I. In this stage, the virtual system minimizes the incentive coin number α. This is regarded as miners try to limit the incentive with others while satisfying the computational hardness target. The virtual system considers the anticipated strategy from each miner and later determines the incentive coin number. Thus, for each miner i, we introduce the optimal storage strategy (7), and constraint (5) can be written as

$$\alpha^{\frac{3}{2}}\sqrt{\frac{j_i}{\beta_i}} \geqslant \frac{Z}{k}. \tag{8}$$

To solve the problem, each miner will calculate an α based on its own. We denote the alpha from each miner i as α_i. The goal is to minimize each α_i and make sure all α_i value is close to each other. Thus, for a specific miner i, the problem (4) to (6) is formulated as

$$\begin{aligned} \min \quad & U(\alpha_i) = \sum_{i\in\mathcal{I}} \alpha_i \\ \text{s.t.} \quad & (6), \\ & \alpha_i^{\frac{3}{2}}\sqrt{\frac{j_i}{\beta_i}} \geqslant \frac{Z}{k}, \forall i \in \mathcal{I}. \end{aligned} \tag{9}$$

The problem formulation above is a quasiconvex optimization [38] of the number of coins α for a PoW block. Equation (8) has square-root, which makes the problem quasiconvex. It is challenging to update the individual x_i from all the miners to the virtual system synchronously to minimize (4). Note that in Stage II, miner i needs to acquire the amount of aggregated computing resources to derive the optimal solution.

5.3 Incentive Allocation Algorithms

In this part, we describe two algorithms we proposed to solve the problem denoted by (9), which is Stage I of the Stackelberg game. As we mentioned above, we propose a "virtual system" for simplicity to represent the leader of the game. The virtual system is not a real entity, and all miners will need to follow.

To solve (9), a crucial part is to obtain the information of $\sum_i x_i$, and equivalently for node i, j_i. After obtaining j_i thus $\sum_i x_i$, node i can get its x_i^* from (7). Then a node can get the corresponding α using such information. For this process, we have developed two algorithms: an iterative algorithm to get the optimal result, and a heuristic algorithm to get a fast result.

Note that the algorithms are executed by the miners which want to propose a block. When a miner proposes a block, it also needs to present the calculated α. If the α is too large, other miners will not accept the proposed block.

We present the algorithms in detail as follows.

5.3.1 Iterative algorithm

We first propose a sequentially updating algorithm based on the Gauss-Seidel iteration shown in Algorithm 1. The iteration rounds of Algorithm 1 are indexed by τ. Each iteration is divided into $|\mathcal{I}|+1$ phases. In phase 0 of iteration 1, all miners concurrently calculate $x_i(1)$ based on their own resources condition. In phase n of iteration τ, miner n determines its computing resource from (7), and $j_n(\tau)$ is updated as follows,

$$j_n(\tau) = \sum_{l<n} x_l(\tau+1) + \sum_{l>n} x_l(\tau).$$

Then miner i calculates the $x_i(\tau)$ from (7) and transmits it to the virtual system. In phase $n+1$ of iteration τ, the virtual system transmits the update j_{n+1} to the miner $n+1$.

Algorithm 1 Incentive Allocation Iterative Algorithm

```
Input: A feasible solution set $\mathbf{x}$ of (3)
Output: $\mathbf{x}$
 1: while $DIF > Thres$ or $round < 100$ do
 2:     for all $i \in \mathcal{I}$ do
 3:         Get $j_i = \sum \mathbf{x_{-i}}$
 4:         Solve (9) to obtain $\alpha_i$
 5:         Get $x_i$ from (7)
 6:         Update $x_i$ in $\mathbf{x}$
 7:     end for
 8:     Calculate $DIF$
 9:     $round+ = 1$
10: end while
```

The basic idea is to solve the problem (9) to get an incentive that miner i proposes. Then, obtain x_i from (7) and using this x_i as a known quantity to solve the problem of other miners. In Algorithm 1, we first need a feasible solution of (3). Then, for each miner i, we get j_i as the summation of computational power devoted from all other miners (lines 2-3). Next, we solve the quasiconvex optimization problem to get α for this specific miner (line 4). We then obtain x_i from (7) and update $\mathbf{x}$ (line 5). After solving x_i, the next miner will use this x_i in its j_i and solve the problem (line 6). After getting all x_i, the algorithm will iterate until certain criteria are reached (line 8 and line 1). Since the problem is to get an α that every miner agrees, the algorithm will terminate when the variation of α is smaller than a certain threshold (DIF).

Next, we discuss the convergence of the algorithm.

Theorem 2. *Algorithm 1 can converge to the global optimum for the edge block system.*

Proof. First, we prove the convergence and global optimality of the Algorithm 1. It is a distributed modified Gauss-Seidel Algorithm. According to Proposition 2.1 of CH.3 in [39] which gives if

- $U(\alpha_i)$ is continuously differentiable and the constraints in convex and compact.
- Given a fixed α_i, miner i determines a unique and optimal response x_i to the virtual system.
- $U(\alpha_i)$ is a unique minimizer when given the feasible x_i.

It can be checked that $U(\alpha_i)$ can satisfy the first requirement because $U(\alpha_i)$ is a linear combination of α_i.

From Theorem 1, $M_i(x_i)$ is concave about x_i and has a unique solution in the given compact space. The second requirement is satisfied.

$U(\alpha_i)$ is a linear function, it has a unique solution given the fixed space which satisfies the third requirement. □

Next, we prove the equilibrium of game G_B. Let α^* denote the equilibrium of game G_B.

Theorem 3. *A Nash equilibrium $\alpha^* = \{\alpha_i^*, \forall i \in \mathcal{I}\}$ exists and is unique in the game* G_B.

Proof. We prove Theorem 3 by Definition 1. From equation (7) in Theorem 1, we can get the optimal x_i which is unique when given fixed α_i. When given the j_i, the miner i has one unique optimal solution. When x_i is returned to the virtual system as the response of incentives α_i, the unique optimal incentives are determined by minimizing $U(\alpha_i)$ which has been proved by Theorem 2. This is a one to one correspondence between x_i and α_i, the computing resource x_i can be written as a function of α_i. For each $i \in \mathcal{I}$, the following Nash equilibrium exists,

$$U(\alpha_i, \alpha_{-i}*) > U(\alpha_i^*, \alpha_{-i}^*),$$

where $\alpha_{-i} = \sum_{l \neq i | l \in \mathcal{I}} \alpha_l$. This equilibrium can be conducted sequentially for each $i \in \mathcal{I}$. For the linear property of $U(\alpha_i)$, the equilibrium is unique. □

5.3.2 Heuristic algorithm

Algorithm 1 can achieve the global optimum through iterations. However, the algorithm has high computational complexity when the network size is large. Therefore, we propose a heuristic algorithm that can run faster and achieve comparable results. We design the heuristic algorithm by observing the characteristic of the constraints, determining the inequality relationships, and estimating the result using these inequalities. The heuristic algorithm requires no iteration and can directly get the estimated number of coins α by a single equation.

We now introduce the detailed design. First, we recall that

$$j_i = \sum_{l \neq i | l \in \mathcal{I}} x_l,$$

it is easy to get that $\sum_i x_i = x_i + j_i$. We can further infer that

$$\sum_i j_i = (N-1) \sum_i x_i, \tag{10}$$

where N is the number of miners in the network. Next, from (7) and (10), under the optimal conditions, we get

$$\begin{aligned} N \sum_i x_i &= \sum_i \sqrt{\frac{\alpha_i j_i}{\beta_i}} \\ &\approx \sqrt{\frac{\bar{\alpha}}{\bar{\beta}}} \sum_i j_i. \end{aligned} \tag{11}$$

Transform (11) by square both sides and apply (10), we get

$$N^2 (\sum_i x_i)^2 \approx \frac{\bar{\alpha}}{\bar{\beta}} \cdot (N-1) \sum_i x_i,$$

$$\frac{N^2}{N-1} \sum_i x_i \approx \frac{\bar{\alpha}}{\bar{\beta}}. \tag{12}$$

Then, according to the constraints (8), we can further obtain

$$\frac{N^2}{N-1} \sum_i x_i \approx \sqrt[3]{\frac{\bar{\beta} Z^2}{k^2 \sum_i x_i}},$$

$$\frac{N^6}{(N-1)^2} (\sum_i x_i)^4 \approx \frac{\bar{\beta} Z^2}{k^2}.$$

Thus, the theoretical total contribution of miners for PoW is denoted as the following equation,

$$\sum_i x_i \approx \sqrt[4]{\frac{Z^2}{k^2 \bar{\beta}^2} \frac{(N-1)^2}{N^6}}. \tag{13}$$

Equation (13) estimates the minimal power devoted of miners under specific parameters.

Next, we need to find α as the number of coins for a new block as $\sum_i x_i$ is obtained. In the previous section, equation (5) connects the minimal computational contribution and block generation period, and equation (8) extends (5) under the optimal strategy. From (5) and (8) we can obtain

$$\bar{\alpha} \approx 2 \sqrt[3]{\frac{Z^2 \bar{\beta}}{k^2 \sum_i x_i}}, \tag{14}$$

where $\bar{\alpha}$ is the average of $\{\alpha_i\}$ and the average number of coins for a new PoW block we need to find.

By combining (13) and (14), we can get

$$\bar{\alpha} \approx 2 \sqrt{\frac{Z \bar{\beta} N}{k}} \sqrt[4]{\frac{1}{(N-1)}}. \tag{15}$$

This equation is used to estimate the average coin for the new PoW block without knowing the different contributions from devices. The heuristic solutions can directly derive an estimated α from this equation given corresponding parameter settings of the network. The process is presented in Algorithm 2.

Algorithm 2 Incentive Allocation Heuristic Algorithm

```
Input: Estimation of $Z, k, \beta, N$
Output: $\sum \mathbf{x}$ and $\alpha$
1: $\sum_i x_i \approx \sqrt[4]{\frac{Z^2}{k^2 \bar{\beta}^2} \frac{(N-1)^3}{N^6}}$
2: $\alpha \approx 2\sqrt{\frac{Z \bar{\beta} N}{k}} \sqrt[4]{\frac{1}{(N-1)}}$
```

The heuristic solution is simple and does not require iterations, and it can give comparable results in a much shorter time. The complexity is $\mathcal{O}(1)$ under such situations. Meanwhile, the heuristic solution does not have any guarantees over the performance, as (13) and (14) do not have the same inequality directions, and the estimations are often inaccurate. Thus, $\bar{\alpha}$ is only an estimation. In real situations, the number of coins obtained is often larger than the optimal result but following the same trends. We will describe the comparison and our discoveries in detail in Section 6.

The virtual system can use the algorithms in the mining process based on the network conditions (discussion on algorithm choices is in Section 6.2.4). When using iterative algorithm, the miners will exchange x_i of themselves, and every miner will collection all other x_i and using (7) to update its own x_i. It will send the updated x_i to other miners for the next round. When using the heuristic algorithm, the author will use parameters Z, k, β, N obtained from the previous block, and using the heuristic algorithm to get $\sum \mathbf{x}$ and α directly.

6 Evaluation

In this section, we implement a simple hybrid consensus protocol over several real edge devices and conduct numerical simulations over incentive assignments. To fully understand the performance of the algorithms, we aim to answer the following three questions:

1) How does the number of coins vary under different settings of network parameters and different computational power distributions?
2) How does the computational power devoted by miners change under different settings of network parameters?
3) How does the mechanism perform with heterogeneous devices in real edge scenarios?

In this paper, we compare the performance of the proposed game-based iterative Algorithm 1 and the heuristic algorithm presented in Section 5.3.2. We conduct several simulations to evaluate our proposed hybrid blockchain incentive assignment mechanism. We focus on evaluating the number of coins given to and computational power devoted by a miner under different settings of PoW mining incentives in the hybrid blockchain. The results are presented in Section 6.2. We further justify the effectiveness of the proposed blockchain system in real devices. We implement the different consensus protocols on three different edge devices and run a small-scale blockchain experiment. The results are presented in Section 6.3.

6.1 Numerical Simulation Setup

We consider a blockchain system consists of 20 miners, which is realistic in the real edge system [40]. The expected ratio of the number of PoS and PoW τ is set as 20. We assume that the maximum computational power of a miner is 1% of the expected power to generate a PoW block in unit time. For fair comparison, we set the related PoS settings as $L = 1000000$, $N = 100$, and $B = 100$. We also implement different distributions of capabilities of the computational power of each device. The setting of different distributions is as follows.

- Uniform distributions: We set the capabilities of computational power at 10 units of each device.
- Normal distribution: We set the average at 10 units and $\sigma = 1$.
- Zipf distribution: $p(k) = \frac{1}{Hk^{\alpha}}$ where we set $\alpha = 1$ and $K = 0.18$.
- Cluster distribution: We set four clusters of miners that have the computational capacity of 4, 8, 12, and 16 respectively. In each cluster, the miners have the same computational capacity, and there are multiple clusters in the network. This simulates that edge devices often have different makes and models, and there are many different kinds of devices in the network and each kind may have several devices.

All of the above settings under different situations have the same average capabilities of computational power for a fair comparison.

The iteration of Algorithm 1 will stop when the variance of α of all miners is less than or equal to 3%. The programming environment is Python 3. The quasiconvex optimization problem (9) is solved by CVXPY [41].

6.2 Numerical Simulation Results

With the setup mentioned above, we perform several simulations to evaluate the preference of the proposed algorithm. We measure the incentive to the miner as the number of coins α that the miner can get when it mines a new block.

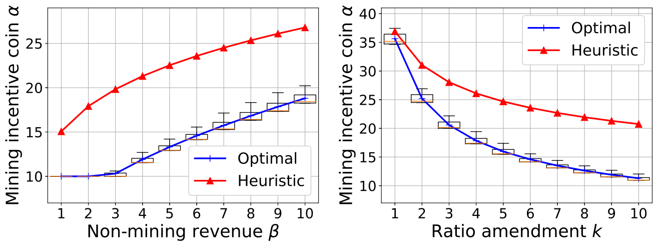
6.2.1 The impact of incentive assignment setting to coin α

In Fig. 4 and Fig. 5, we show how non-mining revenue β and coin number distribution k influence the number of coins α.

Fig. 4(a) exhibits the relationship between the number of coins α and non-mining revenue β when $k = 6$. When β is larger, the miners have a stronger intuition to conduct non-mining tasks rather than PoW mining. Thus, a larger α is needed to incentivize more miners to participate in the mining process. The figure also shows that the mean α (shown in blue crosses) is larger than the median (the orange bars inside each box) under all parameter settings. This indicates that most miners tend to offer a lower α since miners want to lower the incentive of other miners. In real cases, the mean value can be chosen as the value for the number of coins for a new block.

Fig. 4(b) depicts that how the number of coins α varies when ratio amendment k changes when $\beta = 6$. $k = \frac{T}{\alpha}$ denotes the ratio amendment. When k increases, α decreases accordingly both in the optimal and the heuristic algorithm. It indicates that when the mining duration is longer, miners will devote less computational power per time unit and still can satisfy the given difficulty. The overall α will decrease accordingly. Meanwhile, there will be more PoS blocks between two PoW blocks when T is larger. This illustrates miners may need more resources to validate, which in turn can give miners more transaction fees to compensate costs.

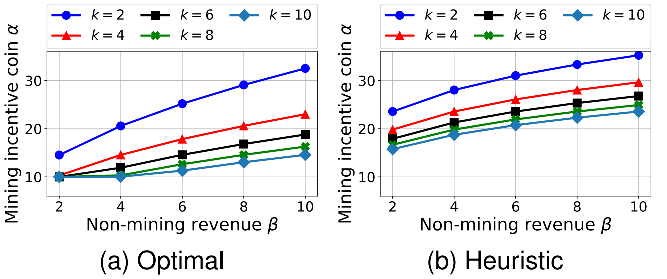
The heuristic algorithm can get the number of coins α in a very short time and offers comparable results. The results are presented in Fig. 4 as red triangles. It shows that heuristic results follow the same trend as optimal results. Overall, it performs well under the same parameter settings. The heuristic algorithm offers between 3.45% to 29.8% more coins compare to the iterative algorithm.



(a) Coin distribution (β) (b) Coin distribution (k)

Fig. 4. The number of coins for mining a new block under different settings of β (a) and k (b) with two different solutions. The heuristic solution follows the same trend as the optimal one and offers fewer coins for new blocks.

To illustrate the performance of proposed algorithms more clearly, we vary β and k at the same time where the computational power follows the uniform distribution. In Fig. 5, k and β vary from 2 to 10 with the interval 2. When β is fixed, α decreases with the increase of k, when k is fixed, α increase as β increases. For all given k, when β increases, α does not increase linearly either in Algorithm 1 and the heuristic algorithm. This is because the probability of mining a block does not grow linearly with the power devoted, and the relation of α and β, k is not linear. The trend of α under different β and k is the same as that in Fig. 4. Meanwhile, since the mining process must satisfy the ratio of PoS and PoW blocks, i.e. constraint (6), α will not decrease when β is small enough to encourage enough participation to satisfy such constraint.



(a) Optimal

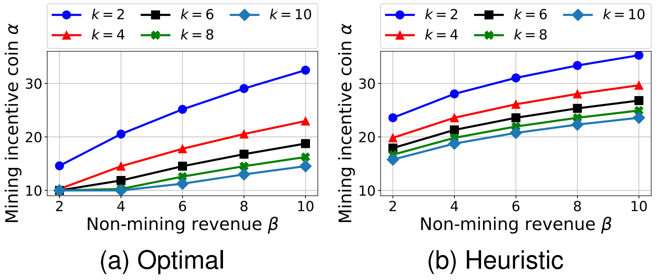
(b) Heuristic

Fig. 5. The average number of coins for mining a new block under different β and k settings of the optimal solution and the heuristic solution. The number of coins increases as non-mining revenue β grows, and decreases as ratio amendment k grows.

6.2.2 *The impact of different computational power distributions*

We conduct a series of simulations under different distributions of computational power capabilities to observe the impact on the incentive coin α. k and β have the same setting as that in Fig. 5. Fig. 6-8 demonstrate the uniform distribution, Zipf distribution, normal distribution and cluster distribution respectively. The trend of α with different distributions is the same. These three distributions are mathematically different. However, both the optimal and heuristic algorithms have a similar performance as we set the same average capabilities of computational power. The difference of α under different distributions is less than 0.13%. It indicates that our proposed algorithms can effectively deal with various environments and guarantee stable performance under the heterogeneity of devices in edge scenarios.

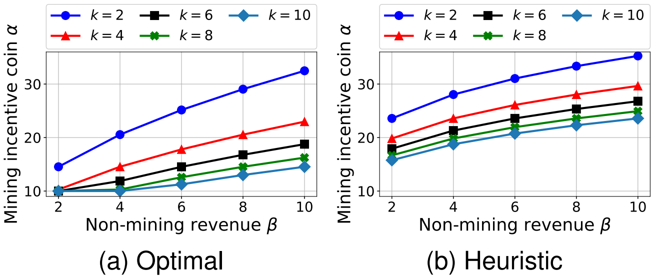
For the heuristic algorithm, the result generated also follows the same trend as optimal results. Overall, the heuristic algorithm gives an average α 19.4% to 19.5% larger than those of the optimal result in the lower k settings. This shows that our proposed heuristic algorithm can also be applied in scenarios with limited resources and offer comparable results.



(a) Optimal

(b) Heuristic

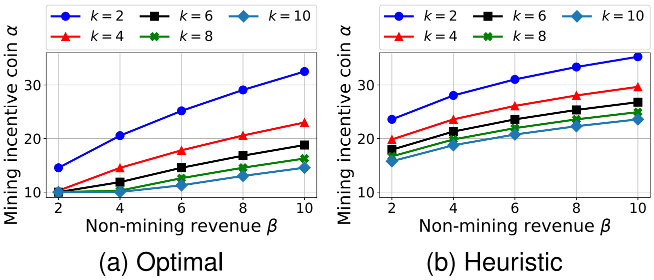
Fig. 6. The number of coins for mining a new block under the **Zipf** distribution of computational capacities of miners. The general trend follows the same as the uniform random distribution which shows that the number of coins increases as β grows, and decreases as k grows.



(a) Optimal

(b) Heuristic

Fig. 7. The number of coins for mining a new block under the **normal** distribution of computational capacities of miners. The general trend follows the same as other distributions which shows that the number of coins increases as β grows, and decreases as k grows.



(a) Optimal

(b) Heuristic

Fig. 8. The number of coins for mining a new block under the **cluster** distribution of computational capacities of miners. The general trend follows the same as other distributions which shows that the number of coins increases as β grows, and decreases as k grows.

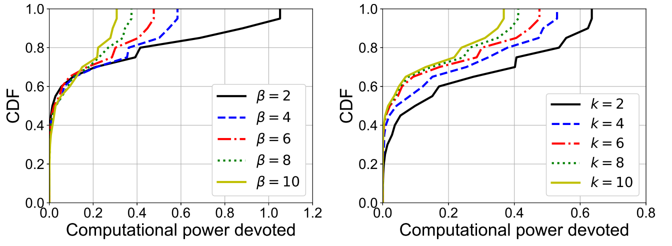
6.2.3 *The impact of incentive assignment setting to power devoted in the PoW process*

We further investigate the impact of incentive assignment setting by observing the variation of the miner's cumulative power devoted and the average computational power needed for the PoW process.

Fig. 9(a) depicts the cumulative distribution of the computational power devoted by miners to the PoW mining process under different non-mining revenue factors β when $k = 6$. When β is larger, miners have weaker motivation to devote computational power for mining because miners can get more income from non-mining tasks. For a specific case, when $\beta = 8$, all miners devoted 0.4 computational power units or less. As a comparison when $\beta = 2$, 20% miners devoted more than 0.9 computational power units.

Fig. 9(b) exhibits the cumulative distribution of the computational power devoted by miners to the PoW mining process under different ratio amendment k when $\beta = 6$. When k is larger, miners will be less motivated to devote to the mining since the given incentive α decreases. For an illustration, when $k = 10$, all miners devoted 0.4 computational power unit or less, while 20% miners devoted more than 0.4 computational power unit when $k = 4$. When k is larger, the miners will have a longer time for mining blocks and less computational power will be required on mining.

In Fig. 10, we show the average computational power devoted under different settings of k and β. When k or β is larger, the average computation power required is smaller which follows the conclusions we mentioned above. We demonstrate this using the example as $\beta = 2$ and $\beta = 4$.

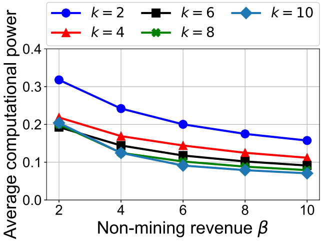


(a) Power devoted (β)

(b) Power devoted (k)

Fig. 9. The distribution of power devoted of different miners under different β (a) and k (b) under different parameter settings.

The average computational power devoted is not change when k decreases. This is because, the incentive α is small when $\beta = 2$ and 4, and increasing k can not satisfy the minimum requirement for the ratio between the numbers of PoS and PoW blocks. Under such circumstances, there still needs certain computational power for mining to keep the required ratio between the number of PoS and PoW blocks.



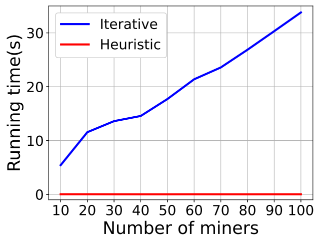


Fig. 10. The average power devoted from miners for the Proof of Work process under different parameter settings.

Fig. 11. The simulation running time on different numbers of miners using iterative algorithm and heuristic algorithm.

6.2.4 *Running time of the iterative and the heuristic algorithms*

For the above tests, we record the iteration time and the running time of both algorithms, which runs on a PC with AMD R7-5700U and 16GB RAM, with a single thread. For the iterative algorithm, the average number of iteration of all tests is 3.991, and the total running time of all 100 tests is 1167.575 seconds. Meanwhile, the total running time of the heuristic algorithm is 2.967 seconds for all 100 tests. The average time of one test is 11.676 seconds and 0.030 seconds for the iterative and the heuristic algorithms respectively. This shows that if a PoW block needs to be generated in less than two minutes, the heuristic algorithm is needed for getting the incentive in time.

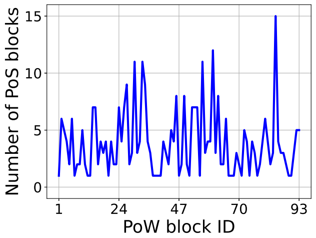
We further conduct simulations over different numbers of miners and record the running time. The results are shown in Fig. 11. The running time of the iterative algorithm increases as the number of miners increases, from 5.43s for 10 miners to 33.80s for 100 miners. The running time of the heuristic algorithm keeps almost the same in different numbers of miners, and all results are below 0.04 seconds. Based on the running time results, we design a mechanism that the system can choose to use the iterative and the heuristic algorithms based on the network sizes and the expected block generating time. The threshold is set to be 1.2 seconds per miner. The expression of the threshold is as follows.

$$threshold = \frac{\mathbb{E}(t_0)}{1.2} \times |\mathcal{I}|,$$

where $\mathbb{E}(t_0)$ is the expected time between two PoW blocks, and $|\mathcal{I}|$ is the number of PoW miners in the network. When the number of nodes is larger than the threshold, the system uses the heuristic algorithm; otherwise, the system uses the iterative algorithm.

6.3 Hybrid Protocol Over Real Edge Devices

In the last part, we evaluate the performance of the hybrid consensus algorithm over real edge devices. We conduct a small-scale experiment of the hybrid consensus system on 5 edge devices and 3 virtual machines, and different devices running different consensus protocols. The PoW mining tests run over three virtual machines on a server with Intel Xeon E5-2560Lv3 CPU and 128GB RAM. Each virtual machine is with 2 vCPU cores and 4GB RAM. For the PoW consensus, the difficulty is set to 7 consecutive zeros in the front of the SHA-256 hash in the hexadecimal form. The mining takes about 80 seconds on our particular virtual machines. The number of incentive coins is calculated using the heuristic algorithm. The PoS minting process uses the setting introduced in Section 4.3, and it is tested using 5 different Raspberry Pi 4B. The expected average generation time t_0 is set to 20 seconds. The largest possible hit value of L is set to 1000000. We focus on the implementation of the consensus protocols and communications between devices. Thus, we use the same dummy transactions for all blocks. In the experiment, a device will send block information to all other devices. Once a user has mined or minted a new block, the device which receives the information will verify the block and start a new mining or minting process.



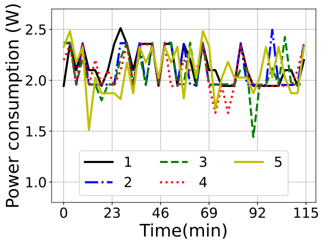


Fig. 12. The variation of the number of PoS block minted between two PoW blocks. The number varies from 0 to 15. On average, 3.89 PoS blocks are generated between two PoW blocks.

Fig. 13. The power consumption of all five minters in the network running the heuristic algorithm. The power consumptions range between 1.4W to 3.5W, with an average of 2.1W.

Table 2 shows some aspects of the experiment results recorded from the chain and devices. We set the total running time as about 2 hours. During this time, 94 PoW blocks are mined throughout the three virtual machines. The incentive for each PoW block is about 10.83, using the heuristic algorithm (Algorithm 2). The average generation time of a PoW block on the specific setting of difficulty is about 36.32 seconds. The average generation time of a PoS block is close to the setting of t_0. On average, there are 3.892 PoS blocks generated between two PoW blocks. Fig. 12 shows the variation of the number of PoS blocks

TABLE 2
Hybrid consensus experiment results

Number of PoW blocks generated	94
Number of PoS blocks generated	362
Average generation time of a PoW block	79.91s
Average generation time of a PoS block	20.53s
Average number of PoS blocks between two PoW blocks	3.892
Maximum number of PoS blocks between two PoW blocks	15
Proposed PoW block incentive coins	10.8279
Average power used for the PoS minting using Raspberry Pi 4	2.1028W

between two PoW blocks. Occasionally, two PoW blocks are mined very quickly and will have only one PoS blocks in between, and there are also spikes with as many as 15 blocks in between.

To measure the energy used by the devices using PoS consensus, we use external power meters and record the instantaneous power consumption of each Raspberry Pi. Fig. 13 shows the instant power consumption of all 5 Raspberry Pi. The average power consumption is about 2.10 Watt. Considering the Raspberry Pi basic power consumption, the power used for PoS is very limited. Overall, the small-scale experiment shows that the proposed system can work over different edge devices.

7 Conclusion

In this paper, we have proposed a PoS-PoW hybrid consensus blockchain system considering the limitations of the edge environments. The system utilizes the heterogeneity of devices making some resource-rich users conduct Proof of Work to enhance the security for transactions in edge environments. We have raised the incentive assignment problem for Proof of Work miners to get fair incentives when mining the new block. We have formulated the problem into a two-stage Stackelberg game and have proposed a Gauss-Seidel based iterative algorithm. We have proven that the proposed algorithm can converge and obtain the global optimum. We have also proposed a heuristic algorithm that can run faster and gives comparable results. We have also conducted a small-scale experiment and extensive simulations. The results show that our proposed incentive assignment mechanism let miners for new PoS block get a reasonable incentive under different system parameters in a small-scale, private edge blockchain.

Acknowledgements

This work is supported in part by US National Science Foundation under grant numbers 1513719 and 1730291.

References

[1] "Gumroad," https://gumroad.com/, [Online; accessed 10-Feb-2019].

[2] "Is gumroad a scam?" https://www.quora.com/Is-Gumroad-a-scam, [Online; accessed 10-Feb-2019].

[3] M. Elbadry, F. Ye, P. Milder, and Y. Yang, "Pub/sub in the air: A novel data-centric radio supporting robust multicast in edge environments," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020.

[4] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," *arXiv preprint arXiv:1301.0159*, 2013.

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[7] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, "Resource allocation and consensus on edge blockchain in pervasive edge computing environments," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1476–1486.

[8] K. Fanning and D. P. Centers, "Blockchain and its coming impact on financial services," *Journal of Corporate Accounting & Finance*, vol. 27, no. 5, pp. 53–57, 2016.

[9] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[10] Nxt community, "Nxt whitepaper," http://nxtwiki.org/wiki/Whitepaper:Nxt, 2014, [Online; accessed 10-Feb-2019].

[11] S. King and S. Nadal, "PPcoin: Peer-to-peer crypto-currency with proof-of-stake," https://peercoin.net/whitepaper, 2012, [Online; accessed 10-Feb-2019].

[12] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[13] Microsoft Research, "Edge computing," https://www.microsoft.com/en-us/research/project/edge-computing/, Oct. 2008.

[14] Y. Huang, X. Song, F. Ye, Y. Yang, and X. Li, "Fair caching algorithms for peer data sharing in pervasive edge computing environments," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 605–614.

[15] A. Samanta, L. Jiao, M. Mühlhäuser, and L. Wang, "Incentivizing microservices for online resource sharing in edge clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 420–430.

[16] Y. Zhong, K. Xu, X.-Y. Li, H. Su, and Q. Xiao, "Estra: Incentivizing storage trading for edge caching in mobile content delivery," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[17] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial iot: Blockchain system with credit-based consensus mechanism," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3680–3689, 2019.

[18] A. S. Sani, D. Yuan, W. Bao, P. L. Yeoh, Z. Y. Dong, B. Vucetic, and E. Bertino, "Xyreum: A high-performance and scalable blockchain for iiot security and privacy," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1920–1930.

[19] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[20] D. B. Rawat, "Fusion of software defined networking, edge computing, and blockchain technology for wireless network virtualization," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 50–55, 2019.

[21] L. Wu, L. Li, X. Li, Y. Yu, L. Zhang, M. Pan, and Z. Han, "Resource allocation in blockchain system based on mobile edge computing networks," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2019, pp. 1–6.

[22] Z. Liu, S. Tang, S. S. Chow, Z. Liu, and Y. Long, "Fork-free hybrid consensus with flexible proof-of-activity," *Future Generation Computer Systems*, vol. 96, pp. 515–524, 2019.

[23] R. P. d. Santos, "Pow, pos, & hybrid protocols: A matter of complexity?" *arXiv preprint arXiv:1805.08674*, 2018.

[24] K. D. Gupta, A. Rahman, S. Poudyal, M. N. Huda, and M. P. Mahmud, "A hybrid pow-pos implementation against 51 percent attack in cryptocurrency system," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2019, pp. 396–403.

[25] J. Hu, M. J. Reed, M. Al-Naday, and N. Thomos, "Hybrid blockchain for iot—energy analysis and reward plan," *Sensors*, vol. 21, no. 1, p. 305, 2021.

[26] M. Harvilla and J. Du, "Prospective hybrid consensus for project pai," *arXiv preprint arXiv:1902.02469*, 2019.

[27] "Decred," https://decred.org/.

[28] Y. Abuidris, R. Kumar, T. Yang, and J. Onginjo, "Secure large-scale e-voting system based on blockchain contract using a hybrid consensus model combined with sharding," *Etri Journal*, vol. 43, no. 2, pp. 357–370, 2021.

[29] Z. Chen, Y. Liu, B. Zhou, and M. Tao, "Caching incentive design in wireless d2d networks: A stackelberg game approach," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[30] Y. Zeng, P. Zhou, J. Liu, and Y. Yang, "A stackelberg game framework for mobile data gathering in leasing residential sensor networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–6.

[31] C. Ye, G. Li, H. Cai, Y. Gu, and A. Fukuda, "Analysis of security in blockchain: Case study in 51%-attack detecting," in *2018 5th International Conference on Dependable Systems and Their Applications (DSA)*. IEEE, 2018, pp. 15–24.

[32] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," *arXiv preprint arXiv:1402.1718*, 2014.

[33] P. Fairley, "Ethereum will cut back its absurd energy use," *IEEE Spectrum*, vol. 56, no. 1, pp. 29–32, 2018.

[34] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 297–315.

[35] S. Andreina, J.-M. Bohli, G. O. Karame, W. Li, and G. A. Marson, "Pots-a secure proof of tee-stake for permissionless blockchains." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 1135, 2018.

[36] K. Poularakis, G. Iosifidis, and L. Tassiulas, "A framework for mobile data offloading to leased cache-endowed small cell networks," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2014, pp. 327–335.

[37] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[38] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

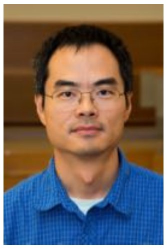
[39] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[40] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1085–1100.

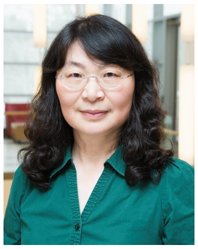
[41] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

Yaodong Huang received his B.E. in Computer Science and Technology from University of Electronic Science and Technology of China, and his Ph.D. degree in Computer Engineering from Stony Brook University in New York. He is now working at the College of Computer Science and Software Engineering of Shenzhen University. His research interests are in mobile edge computing, with focus on data caching, security, privacy and energy-efficiency in edge environments.

Yiming Zeng received the B.Eng degree in Information Engineering from the Shanghai Jiao Tong University, Shanghai, China. He is a Ph.D. candidate in Computer and Electrical Engineering at Stony Brook University, New York. His research focuses on addressing computing, privacy, and caching issues in edge networks.



Fan Ye is an Associate Professor in the ECE department of Stony Brook University, before that he was a Research Staff Member at IBM T. J. Watson Research after getting his Ph.D. from UCLA CS department in 2004. His research interests include mobile sensing platforms, systems and applications in healthcare and location based services, edge computing, Internet-of-Things, and data-centric wireless communication. He has published over 100 papers with 12,000+ citations according to Google Scholar, and 30 granted/pending patents/applications. He has received NSF CAREER award, Google Faculty Research Award, IBM Research Division Award and 5 Invention Achievement Plateau awards, Best Paper Award for IEEE ICCP 2008. He has been a panelist for NSF, NIH and Canada, Hong Kong government funding agencies, on program/organizing committees for conferences including IEEE Infocom, IEEE ICDCS, ACM Mobicom, ACM Sensys.



Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University, Beijing, China, and the MSE and PhD degrees in computer science from Johns Hopkins University, Baltimore, Maryland. She is a SUNY Distinguished Professor of computer engineering and computer science at Stony Brook University, New York, and is currently on leave at the National Science Foundation as a Program Director. Her research interests include edge computing, data center networks, cloud computing and wireless networks. She has published over 480 papers in major journals and refereed conference proceedings and holds seven US patents in these areas. She is currently the Editor-in-Chief for IEEE Transactions on Cloud Computing and an Associate Editor for ACM Computing Surveys. She has served as the Associate Editor-in-Chief for IEEE Transactions on Computers and IEEE Transactions on Cloud Computing and Associate Editor for IEEE Transactions on Parallel and Distributed Systems and IEEE Transactions on Computers. She has also served as a general chair, program chair, or vice chair for several major conferences and a program committee member for numerous conferences. She is an IEEE Fellow.