Game Theoretic Storage Outsourcing in the Mobile Blockchain Mining Network

Suhan Jiang and Jie Wu
Department of Computer and Information Sciences, Temple University
{suhan.jiang, jiewu}@temple.edu

***Abstract*—Besides the computation limitation, the requirement of storing the entire blockchain is another challenge for blockchain mining in mobile environments, and thus has hindered the development of blockchain-powered mobile applications. Storage outsourcing to a cloud service provider (CSP) is a viable solution. An individual miner can store his blockchain in the cloud and then validate transactions by querying the CSP. However, validation outsourcing to a remote CSP incurs delay and damages a miner's winning probability in the mining competitions. To shorten such an unwanted delay, miners can also cache the unspent transaction output (UTXO) set in a nearby edge service provider (ESP) for fast transaction validations, which definitely brings extra costs. In this paper, we consider a two-layer outsourcing paradigm to solve storage shortage for mobile miners. Due to the delay-cost tradeoff when selecting service providers, we can model interactions among miners as a non-cooperative game and formulate a Nash equilibrium problem to investigate the effects of outsourcing on miners' utilities. We also study the access probability of UTXOs with different generation times. This will guide miners on how to select unspent transaction outputs if they decide only to cache the partial UTXO set in the edge. We further extend our game by modeling multiple mining rounds as a one-shot game to see how the cache update frequency affects miners' strategies. Numerical evaluation is conducted to show the feasibility of storage outsourcing and to validate the proposed models and theoretical results.**

***Index Terms*—Game theory, mobile blockchain mining, storage outsourcing, UTXO.**

I. Introduction

There is a wide adoption of blockchain technology by different fields ranging from cryptocurrency, financial services, IoT to public and social services. As a distributed ledger, blockchain records data in the form of linked blocks secured by cryptography. To achieve the tamper-proof, reliability and traceability of transactions in a trustless environment, a blockchain-powered application usually requires each miner to store its complete history, the storage of which is not negligible at all. For example, Bitcoin, a pioneer of public blockchain platforms, has a 279 GB onchain data at present, and its newly generated data for each year takes about 50 GB.

Such a high storage requirement poses challenges on mobile devices, and thus hinders the development of mobile blockchain services. To facilitate blockchain-powered applications in future mobile IoT systems, storage outsourcing appears to be a viable solution. Miners using mobile devices can overcome storage limitations by offloading the complete onchain data to an external cloud storage. Thereby, a mobile miner can query the corresponding cloud service provider (CSP) when validating a transaction. Obviously, storage outsourcing to a CSP alleviates storage shortage while bringing the query delay due to the communication time between miners and the CSP. A miner can turn to a nearby edge storage for help as a naive way to the query delay reduction. Given the limited capacity and the high price of an edge service provider (ESP), storing the complete blockchain in the edge may not be a wise choice. Instead, miners can cache the unspent transaction output (UTXO) set. A UTXO is an output of a blockchain transaction that has not been spent. A newly-issued transaction can be fast validated without querying the entire blockchain if its input(s) matches with a certain UTXO(s),

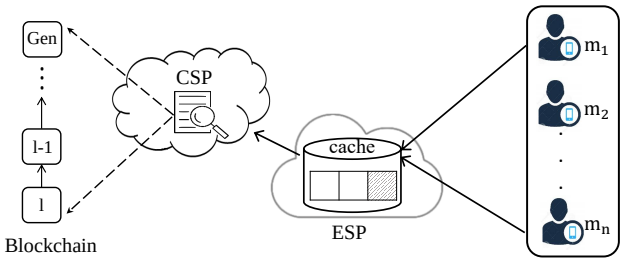


Fig. 1: Miners compete for cache resources from the ESP, while the CSP provides a backup database for transaction validations.

This paper considers a two-layer outsourcing paradigm, including a remote CSP and a nearby ESP, for mobile miners to store blockchain data and validate transactions. As depicted in Fig. 1, each miner outsources his blockchain to the cloud given the storage limitation of his device, and queries the CSP when he needs to validate transactions. Additionally, a miner can obtain fast transaction validations provided by the ESP if he chooses to cache the UXTO set in the edge. Due to the delay-sensitive nature of mining, a lower-latency validation service improves the winning probability of this miner but causes extra costs for him. In reality, a miner may not be able to cache the entire UTXO set in the edge, but he still can get benefit by partially caching the UTXO set if his strategy is reasonable (in fact, the main objective of this paper is to find the reasonable, or say, the optimal strategy for each miner). In this partial caching setting, if the ESP fails to validate a transaction for the miner due to cache missing, the transaction will be automatically redirected the CSP for further queries.

In this paper, we propose a non-cooperative game we call storage outsourcing game (SOG) to model interactions among all mobile miners. In the proposed game, each miner aims to maximize his utility, *i.e.*, a difference between the expected

block mining reward and the cost incurred by storage and validation outsourcing. In our most basic model, we assume that all UTXOs have an identical access probability, and miners will update their cache contents with new UTXOs before mining a new block. Thus, a single mining round is viewed as a one-shot game, and a miner only focuses on deciding the edge cache size without considering which UTXOs to select. Meanwhile, the miner also needs to decide his block size carefully, since a larger block containing more transactions indicates higher expected rewards, *i.e.*, more transaction fees, while also leading to longer validation delays, no matter where they are validated. After investigating real-world data from Bitcoin, we observe a fact that the probability of a UTXO is spent by its owner varies over time and is related to its generation time. This means that UTXOs with different generation times have different access probabilities at a given time point. We further study how the access probability of a certain UTXO changes as time goes, and we use a log-normal distribution to capture the corresponding relation.

Based on this observation, each miner is facing a challenge, where he has to jointly optimize the size and the content of his edge cache, given the fact that even randomly caching in a fixed storage may also bring different cache hit rates and thereby affect the transaction validation delays. Previous discussions are based on the requirement that each miner updates his cache every mining round. We extend this model by considering multiple mining rounds as a one-shot game. In this case, each miner lowers his cache update frequency, and his objective is to maximize the accumulative utilities in the following T mining rounds. In this paper, we consider T as a common knowledge in the proposed game. That is, the value of T is pre-defined and identical among all miners. In fact, T can be viewed as a variable in the miner's strategy space. If so, each miner needs to determine his own value of T for utility maximization. Given the different values of T, miners update their cache asynchronously, which makes time as an inevitable dimension in the game. The corresponding analysis and solutions are based on stochastic game theory. Due to the page limitation, we decide to make this case as our future work. The major contributions of this paper are as follows:

- We propose a non-cooperative game to solve a price-based resource management problem in a mobile blockchain mining network with two SPs.
- We study access probabilities of blockchain unspent transaction outputs and formulate a function to capture how an unspent transaction output's popularity changes over time.
- We analyze the existence and uniqueness of Nash equilibrium (NE) in two settings, *i.e.*, the single-round setting and the multiple-round setting, based on which algorithm is proposed to obtain NE solutions.
- We perform numerical evaluation based on real-world data and the results are consistent with all the theoretical results.

II. System Model and Problem Formulation

This paper focuses on a mobile blockchain mining network. The basic setting on blockchain technique is based on Bitcoin.

TABLE I: Summary of Notations.

Symbol	Description
p_e / p_c	price of edge VM / cloud query
d_e / d_c	single transaction validation delay from edge / cloud
b / ρ	block base reward / transaction fee density
S	number of network-wide unspent transaction outputs
a_k	probability of accessing unspent transaction output k
n	number of miners
m_i / B_i / U_i	miner i's mining power / budget / utility
R_i / P_i / C_i	miner i's expected reward / winning probability / cost
x_i / y_i	miner i's block size / edge storage request
z_{ik}	miner i's cache decision on unspent transaction output k
X / Y	total cloud-mining / self-mining units
X_{-i}	total cloud-mining units except m_i's, *i.e.*, $X_{-i}=X-x_i$
Y_{-i}	total self-mining units except m_i's , *i.e.*, $Y_{-i}=Y-y_i$
r_i	m_i's request vector, in the form of (x_i, y_i)
$\boldsymbol{r_{-i}}$ / $\boldsymbol{r}$	all miners except m_i's / all miners' request profile
β	discount factor caused by a unit-time delay

Edge cache storage unit is tailored as an unspent transaction output size.

That is, we assume all blockchain users follow the Proof-of-Work (PoW) consensus protocol and apply the UTXO account model. Corresponding notations are listed in Table I. Our model includes two service providers and a set of n miners using mobile devices. Fig. 1 depicts an overview of this network. The SP side consists of a remote CSP and a nearby ESP, offering storage and query services to miners. Usually, large datasets are outsourced to the CSP given that its resources are rich and cheap. If users want to get query answers quickly, then the ESP is a better choice due to its close physical location. The delay-cost tradeoff makes the coexistence of the CSP and the ESP in order to satisfy users of different service quality requirements and different budgets.

The user side is a network with n miners using different mobile devices. Miners compete against each other in hopes of generating new blocks and getting rewards. The process of adding a block to the blockchain is viewed as a mining round. In a mining round, each miner has to create his own candidate block. The process of creating a candidate block consists of 3 steps. First, a miner validates unconfirmed transactions, and then bundles them to form a Merkle tree structure, which produces a Merkle root. Finally, the miner uses his computation, *i.e.*, mining power, to solve a PoW puzzle based on the previously produced Merkle root. Due to the storage limitation of their devices, all miners outsource their blockchain in the cloud. A miner i will issue queries to the CSP when he needs to validate transactions. Each query is charged at the price of p_c and the corresponding answer returns at a delay of d_c. Thus, the cost and the time for miner i who wants to validate x_i transactions are $x_i p_c$ and $x_i d_c$, respectively. Once a miner finds a PoW solution, he will broadcast his block in the mining network for consensus. A miner whose block reaches consensus first will be the winner and get rewarded in that round. Mining rewards come from two sides: one is a fixed base reward for a block creation, the other is transaction fees accumulated in this block. Thus, if miner i successfully mines a block containing x_i transactions, his expected reward is $R_i = b + \rho x_i$, given the base reward b and the transaction fee density ρ.

Obviously, miners can have different mining rewards, depending on how they decide their block sizes. Selecting a large value of x_i (under the constraint that x_i is more than the number of unconfirmed transactions in the network) definitely brings a higher expected reward R_i for miner i. However, it also takes a longer delay to validate those transactions. As we mentioned before, miner i wins unless he is the first to solve his PoW puzzle and propagate his block to reach consensus. A large delay $x_i d_c$ damages miner i's winning probability in the mining competition. The tradeoff between the expected reward and the winning probability poses a challenge to miner i on deciding the value of x_i. Except shrinking his block for a shorter delay, miner i can turn to the ESP for a low-latency query service. In this case, miner i can cache the UTXO set in the edge for fast transaction validations. A transaction is valid if each of its inputs is available in the UTXO set. If miner i maintains a cache of the UTXO set in the edge, he can validate transactions without accessing the blockchain in the cloud. The ESP processes each query from miner i by iterating his cache space. Let d_e denote a single transaction validation delay from the ESP, and then the total transaction validation delay will be $x_i d_e$ if he wants to validate x_i transactions.

In fact, $x_i d_e$ is the optimal result for miner i with a block size x_i, given the fact that he maintains the entire UTXO set in the edge. In reality, given budget constraints, high prices and limited capacity of edge resources, miner i may prefer to cache part rather than all of UTXOs in the edge. Thus, the ESP will charge miner i a cost of $y_i p_e$ if he maintains a cache with y_i UTXOs (p_e is a combo charge of storage and query services). We assume that, the CSP offers a lower price, *i.e.*, $p_c < p_e$ while the ESP guarantees a shorter delay, *i.e.*, $d_e < d_c$. These assumptions always hold in the real world and also guarantee the problem discussed in this paper is meaningful.

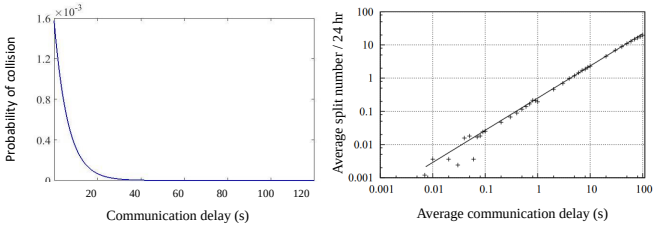
Miners participate in mining processes by requesting resources and services from the SPs. Each miner i's request is in the form of $r_i = (x_i, y_i)$, where x_i represents the number of transactions miner i decides to validate and put in his current block, and y_i represents the number of unspent transaction outputs he decides to cache in the edge. Let $\boldsymbol{r} = \{r_1, \cdots, r_n\}$ and $\boldsymbol{r}_{-i}$ represent the request profile of all miners and all other miners except i, respectively. As miners all want to make as much profit as possible, a competition among miners is formed, in which each miner optimizes his utility by deciding his request r_i under the current resource prices (p_e, p_c), while considering his own budget B_i and other miners' strategies $\boldsymbol{r}_{-i}$. Since requests are generated for individual utility maximization, a non-cooperative game is also formed. Miner i's optimization problem is defined as follows.

Problem 1 (OP_{MINER})**.**

$$\textit{maximize} \qquad U_i = R_i \cdot P_i - C_i, \tag{1a}$$

$$\textit{subject to} \qquad C_i \leq B_i, \quad x_i \geq 0, \quad y_i \geq 0, \tag{1b}$$

P_i and C_i represent miner i's winning probability and his cost charged by the CSP and/or the ESP, respectively. Given their complexity, accurate definitions and detailed explanations of P_i and C_i will be given in the following part. Each miner i aims to maximize his utility and constraint (1b) ensures that i is within his budget B_i.



(a) Probability density function of a conflicting block being found while there exists another block being propagated in the network [1].

(b) Average number of blockchain forks per 24 hours as a function of communication delay, averaged over all the nodes in the network [2].

Fig. 2: Communication delay can cause damage winning probability.

III. Individual Winning Probability and Cost

A. Cache Hit/Miss and Cost

As we mentioned before, miner i can validate any unconfirmed transaction without accessing his cloud storage only if he maintains the complete UTXO set in the edge. Currently, the UTXO set is close to 4 GB and it also grows as fast as the blockchain itself. It has been predicted that the size of the UTXO set will grow to close to 20 GB within the next few years, even when some effective actions are implemented for size reductions. Taking cost-efficiency into consideration, miner i may maintain partial instead of all UTXOs in the edge. We assume the complete UTXO set contains S unspent transaction outputs in total and the size of each is identical. If miner i caches a size of y_i UTXOs ($y_i < S$), then not all transactions can be validated in the edge, given the fact that some UTXOs are missing in the miner i's edge cache. In the case of cache missing, all those transactions failing to be validated in the edge will be redirected to the CSP for further confirmation, and miner i is responsible for paying the corresponding query cost to the CSP. Let h_i denote miner i's cache hit rate in the edge, then his total cost can be expressed as $C_i = p_c x_i (1 - h_i) + p_e y_i$. Definitely, h_i is a function over the variable y_i. Intuitively, the relationship between h_i and y_i can be characterized by a uniform distribution, *i.e.*, $h_i = y_i / S$.

B. Validation delay and Winning Probability

The mining power m_i characterizes the probability that miner i happens to solve a PoW puzzle. However, outsourced transaction validations incur delay and discount miners' winning probability. Miner i starts mining until his x_i transactions are validated. The waiting time damages miner i's winning probability, since other miners with shorter delay can find and publish their blocks during that time. Thus, mining is not just a race on miners' contributed computing power. Generally, miner i's winning probability P_i is discounted by the delay. Their relation has been studied in Bitcoin [3], which is subject to an exponential distribution as shown in Fig. 2(a). Thereby, the discount rate is almost linearly proportional to the delay, as shown in Fig. 2(b).

In this paper, we assume that the proposed network follows the same pattern in Bitcoin. In our setting, the transaction validation delays between the SPs and miners can be an important inducement that lowers miner i's winning probability. We define β as the unit-time discount factor. Given each miner j's delay d_j and his mining power m_j, the weighted average delay is $\sum_{j=1}^{n} m_j d_j$, which leads to a winning probability discount rate of $\beta \sum_{j=1}^{n} m_j d_j$ in the entire mining network. Thus, miner i's winning probability can be captured as $P_i = m_i \left(1 - \beta \sum_{j=1}^{n} m_j d_j\right)$. To focus on the influence of the transaction validation delay, we neglect the block broadcast delay. Miner i's transaction validation delay d_i depends on his request. If he decides to only use the service provided by the CSP, *i.e.*, $y_i = 0$, then d_i is linear to his block size x_i. However, if he caches UTXOs in the edge, all x_i transactions are filtered by the ESP first, and then, with a cache miss rate of $1 - h_i$, the remaining $x_i(1 - h_i)$ transactions are redirected to the CSP for further confirmation. Thus, d_i can be expressed in Eq. (2).

$$d_i = \begin{cases} d_c x_i & y_i = 0 \\ d_e x_i + d_c x_i (1 - h_i) & y_i > 0 \end{cases}. \tag{2}$$

IV. Game under Uniform Access Probability

In a single mining round, all miners focus on validating transactions and solving their own PoW puzzles. Once a block is found, all miners move on to find the next block. This process is repeated indefinitely. The repeated generation of blocks becomes a series of independent one-shot competitions. We consider each mining round as a one-shot game played by all miners. A miner's strategy is the choices of the block size and the cache size in the edge. The choices are made a-priori by all miners. The cached unspent transaction outputs are just randomly picked from the complete UTXO set based on a miner's request on the edge cache size, given the uniform distribution assumption in the cache hit rate.

A. Unlimited Resource Capacity of ESP

We start with a scenario where the ESP has unlimited resource capacity, which means all miners' requests to the ESP will be completely fulfilled. In this scenario, a miner optimizes his utility by solving Problem 1.

Theorem 1. *A Nash equilibrium exists in* OP_{MINER} *if the ESP has unlimited resource capacity.*

Proof. Any game has NEs if its equivalent variational inequality (VI) problem [4] has a nonempty solution set. Given a VI problem, $VI(K, G)$, if K is convex and compact, and F is monotone on K, then the solution set of $VI(K, G)$ is nonempty, closed, and convex.

We start with the definition on the equivalent VI problem $VI(K, G) \equiv OP(X, U)$, where

$$G := (\nabla_i U_i)_{i=1}^n, \quad X := ((x_i, y_i))_{i=1}^n, \quad U := (U_i)_{i=1}^n,$$
$$K := \prod_{i=1}^n K_i, \quad K_i := \{(x_i, y_i) | C_i \le b_i, x_i, y_i \ge 0\}.$$

It can be easily verified that K_i is convex and closed, $\forall i$. Thus, K is convex and compact. G is monotone if and only if $U_i(r_i, \boldsymbol{r}_{-i})$ is concave in r_i for given $\boldsymbol{r}_{-i}$, $\forall i$, which is true as shown below. Since the VI problem has a nonempty solution set, the existence of NE thus follows the sufficient conditions.

We start with the simple case where miner i decides to query the CSP for transaction validation without investing edge cache resources. In this case, $y_i = 0$ holds and miner i needs to solve a single-variable maximization problem if other miners' decisions are known to him. Obviously, $U_i = (b + r x_i) m_i \left(1 - \beta \sum_{j \neq i} m_j d_j - \beta m_i d_i\right) - p_c x_i$ is a concave quadratic function. We then move to the case where miner i decides to cache some UTXO sets in the edge in order to speed up transaction validation. In this case, he has to determine how much storage to request from the ESP as well as his block size. His transaction validation delay is $d_e x_i + d_c x_i (1 - h_i)$ and his cost charged by both the CSP and the ESP is equal to $p_e y_i + p_c x_i (1 - h_i)$. To find miner i's best response strategy, we investigate the concavity of his utility function.

Denote H for the Hessian matrix of U_i:

$$H := \begin{bmatrix} U_{xx}^i & U_{xy}^i \\ U_{yx}^i & U_{yy}^i \end{bmatrix}$$

where $U_{xx}^i = \dfrac{\partial^2 U_i}{\partial x_i^2}, U_{xy}^i = U_{yx}^i = \dfrac{\partial^2 U_i}{\partial x_i \partial y_i}, U_{yy}^i = \dfrac{\partial^2 U_i}{\partial y_i^2}$.

Then the first-order derivative of miner i's utility function is:

$$\begin{aligned} \frac{\partial U_i}{\partial x_i} =& m_i r \left(1 - \beta \sum\nolimits_{j=1}^{n} m_j d_j\right) - p_c (1 - h_i) \\ & - \beta m_i^2 (b + r x_i) \left[d_e + d_c (1 - h_i)\right], \\ \frac{\partial U_i}{\partial y_i} =& \left[\beta d_c x_i (b + r x_i) m_i^2 + p_c x_i\right] / S - p_e. \end{aligned}$$

The expressions of the Hessian elements are as below:

$$\begin{aligned} U_{xx}^i =& -2\beta r m_i^2 \left[d_e + d_c (1 - h(y_i))\right], \\ U_{xy}^i =& U_{yx}^i = \left[\beta d_c m_i^2 (b + 2 r x_i) + p_c\right] / S \\ U_{yy}^i =& 0. \end{aligned}$$

Next, we show H is negative definite by proving its leading principal minors, *i.e.*, U_{xx}^i and $det(H)$, are smaller than 0.

$$\begin{aligned} det(H) &= U_{xx}^i U_{yy}^i - U_{xy}^i U_{yx}^i \\ &= -\left[(p_c + \beta b d_c m_i^2 + 2\beta r d_c m_i^2 x_i)^2\right] / S^2, \end{aligned} \tag{3}$$

the sign of which is always negative for a non-empty block. Obviously, miner i has a concave utility function that definitely will yield a maximal utility point. Therefore, we have proved that U_i is strictly concave with respect to (x_i, y_i). Accordingly, the Nash equilibrium exists in this game. The proof is now completed. □

Here, we provide a distributed algorithm (Algorithm 1) which computes the NE solution to the OP_{MINER}. When updating his request vector, miner i always applies a standard Lagrange multipliers optimization solution based on his own OP_{MINER}.

B. Resource Limitation

Edge Computing is praised for its short delay while also being criticized for its limited resource capacity. In reality, it

is possible that the ESP cannot fulfill all requests from miners.

1) Generalized Nash Equilibrium: In the perspective of game theory, we can model this game as a generalized Nash equilibrium problem (GNEP). GNEPs differ from classical Nash equilibrium problems (NEP) in that, while in an NEP only the players' objective functions depend on the other players' strategies, in a GNEP both the objective functions and the strategy sets depend on the other players' strategies. In our case, the ESP only has a total of Y_{max} resource units, where Y_{max} is a common knowledge in this game. It has to reject some requests when overloaded. Thus, the aggregate requests from all miners should be no more than Y_{max} in order to avoid being rejected. Thus, given other miners' requests $\boldsymbol{r}_{-i}$, miner i should ensure that y_i can be satisfied by the ESP. Mathematically, this can be written as $\sum_{j=1}^{n} y_j \leq Y_{max}$.

Now, we reformulate the OP_{MINER} problem in the following.

Problem 1b ($\text{GNEP}_{\text{MINER}}$)**.**

$$\textit{maximize} \qquad U_i = R_i \cdot P_i - C_i, \tag{4a}$$

$$\textit{subject to} \qquad \sum\nolimits_{j=1}^{n} y_j \leq Y_{max}, \tag{4b}$$

$$C_i \leq B_i, \quad x_i \geq 0, \quad y_i \geq 0. \tag{4c}$$

Constraint (4b) ensures that miner i's request to the ESP can be fully satisfied. Since all miners' requests are mutually dependent, the $\text{GNEP}_{\text{MINER}}$ problem is a Generalized Nash Equilibrium Problem (GNEP). In $\text{GNEP}_{\text{MINER}}$, the dependence of each miner's strategy set on the other miners' strategies is represented by the (linear) constraint (4b), which includes each miner's request y_i to the ESP. More specifically, since the miners all share a jointly convex shared constraint, this game is known as a jointly convex game.

Theorem 2. *Given a price set (p_e, p_c) from the SP side, there exist at least one Nash equilibrium for the non-cooperative game at miner side given that the ESP's resource capacity Y_{max} is a common knowledge to all miners.*

Similar with the proof for OP_{MINER} NE in Theorem 1, the existence of NE in $\text{GNEP}_{\text{MINER}}$ is easily followed by capitalizing on the variational inequality theory. In general, a GNEP could have infinite solutions. Namely, there are multiple NEs among miners, and thus there is no efficient algorithm to obtain the global optimal strategy in the proposed game. Algorithm 1 is still feasible here if each miner i updates his request vector using a standard Lagrange multipliers optimization solution based on his own $\text{GNEP}_{\text{MINER}}$. Note that, Algorithm 1 promises to compute a solution while there is no guarantee that the produced NE is a global optima.

2) Auction-based Partial Fulfillment: The application of GNEP has two deficiencies in reality. First, it usually has infinite solutions and no guarantee on global optimization, leading to an unpredictable equilibrium. Second, its convergence speed is a big concern as well. Another real-world problem is that Y_{max} may not be revealed to all miners and usually it may vary in each mining round, given the fact that the ESP also serves users outside the mining network.

Algorithm 1 Best-Response Algorithm

Output: $\boldsymbol{r} = \{r_1, \cdots, r_n\}$ where $r_i = (x_i, y_i), \quad i \in \{1, n\}$

Input: Choose any feasible starting point $\boldsymbol{r}^{(0)}$: each miner chooses the decision using the local computing

1: **for** round k **do**
2: **for** miner i **do**
3: Decide $r_i^{(k)} = r_i^{(k-1)} + \Delta \frac{\partial U_i\left(r_i, r_{-i}^{(k-1)}\right)}{\partial r_i}$
4: Send the request $r_i^{(k)}$ to SPs
5: SPs collect the request profile $\boldsymbol{r}^{(k)}$
6: **if** $\boldsymbol{r}^{(k)} = \boldsymbol{r}^{(k-1)}$ **then** Stop

Since all the resources/services are requested before a mining game starts, we can consider an alternative solution: a miner-side auction. Auctions help allocate and price scarce resources in settings of uncertainty. In this situation, each miner i simultaneously reports his bid on the amount y_i and the unit price p_e^i to the ESP. Then, the ESP applies a VCG mechanism to allocate resources to miners with certain charges, based on the value of Y_{max} and miners' bids at that time. This will result in a case that miner i's request on y_i is partially satisfied, which fits well with the reality.

V. Joint Optimization of Cache Size and Content

Previously, we simply characterized the access probability of each UTXO in a given mining round to be identical. However, lots of recent works [5, 6] on the Bitcoin UTXO set reflects a fact that the lifespan of a UTXO, the period from the time when it becomes spendable to the time when it is confirmed to be spent by its owner, varies. This observation indicates that, UTXOs with different birth times should have different access probabilities at a given time. Thus, our basic model, which assumes all UTXOs have an identical access probability no matter when they become spendable, seems a little bit rough and should be refined to be more in line with reality. Section VI will discuss how the access probability of an unspent transaction output changes as time goes. In this section, we assume the relationship is given.

Given the fact that UTXOs may have different probabilities of being accessed in a specific mining round, miner i who plans to invest on edge resources cannot randomly pick from the UTXO set for caching any more. He should not only consider the cache size, but also the cache content, *i.e.*, which UTXOs should be selected in the requested cache space. Thereby, miner i faces a joint optimization problem where the cache size and the cache content have to be decided simultaneously. In the below, we focus on this more realistic setting where contents to be selected and cached have different accessing probabilities.

A. Single Mining Rounds as a One-shot Game

Intuitively, miner i always tends to cache contents with high access probabilities in the hope of improving his cache hit rate and hence shortening his delay and avoiding extra costs to the CSP. Thus, in a given round, all UTXOs, S in total, are sorted

based on their access probability a_k in the descending order. Define z_{ik} as a decision variable, indicating whether miner i decides to cache the k-th UTXO in the edge. That is, z_{ik} equals to either 1 if the k-th UXTO is selected by miner i, or 0 otherwise. Obviously, his requested cache size can be expressed as $y_i = \sum_{k=1}^{S} z_{ik}$, and the corresponding cache hit rate also can be rewritten as $h_i = \sum_{k=1}^{S} z_{ik}a_k / \sum_{k=1}^{S} a_k$. Now, miner i's strategy space is extended into three dimensions: (1) block size x_i, (2) cache size y_i, and (3) cache content z_{ik}, $\forall k \in [1, S]$. And his utility becomes a function over variables x_i and z_{ik}, $\forall k \in [1, S]$. We reformulate the optimization problem as follows.

Problem 1c (OP_{MINER})**.**

$$\textit{maximize} \quad U_i = R_i \cdot P_i - C_i, \tag{5a}$$

$$\textit{subject to} \quad x_i \geq 0, \quad z_{ik} \in \{0, 1\}, \forall k \in [1, S]. \tag{5b}$$

where $y_i = \sum_{k=1}^{S} z_{ik}$*, and* $h_i = \sum_{k=1}^{S} z_{ik}a_k / \sum_{k=1}^{S} a_k$.

Corollary 1. *Nash equilibrium still exists even if each unspent transaction output has non-uniform access probability in a given mining round.*

The uniform-access-probability setting is a special case where all a_ks are identical. Similar to the proof for NE in Theorem 1, the existence of NE for miners in a non-uniform-access-probability setting is followed by capitalizing on the variational inequality theory. Based on the previous analysis, we need to show that U_i in Problem 1c is a concave function over variables x_i and z_{ik}, $\forall k \in [1, S]$. According to the proof in Theorem 1, we can obtain the fact that U_i is a concave function over variables x_i and y_i. For a given mining round, all a_ks are constants, so y_i is an affine function over z_{ik}, $\forall k \in [1, S]$. Therefore, the composite function U_i is still concave over x_i and z_{ik}, $\forall k \in [1, S]$.

B. Multiple Mining Rounds as a One-shot Game

Now, miner i figures out how to dedicate his cache storage in order to maximize his cache hit rate. His cache replacement policy is still to delete transaction outputs spent in the previous mining round, and refill with new unspent transaction outputs with high access probabilities in the next mining round. In fact, the update of cached contents may not be very frequent (*e.g.* on the order of hours) so as to reduce overload cost and complexity. Another issue is that, usually users take advantage of edge resources in a pre-ordered way instead of a preemptive way, as a preemption process incurs too much uncertainty. A non-preemptible usage of edge resources requires a user to report what and how many resources he wants, as well as how long he will occupy the requested resources. These two facts lead us to think about a more realistic scenario, *i.e.*, miners dedicate their cache storage (both sizes and contents) in a relatively longer-term view, *i.e.*, the minimum time period requested by the ESP for providing non-preemptible services.

Assuming that the minimum time period contains T mining rounds, then each miner's strategy is made to maximize his utility in the following T mining rounds (our previous analysis can be viewed as a special case of $T = 1$). When these T mining rounds end, miner i resubmits his requests to the SPs and updates his cache in the edge. Now, we move to a new scenario where every T mining rounds are viewed as a one-shot game. In such a one-shot game with T mining rounds, we assume all miners start mining at the same time point, *i.e.*, mining round 1. Before round 1 begins, the cached contents should be reasonably updated so that they can be repeatedly accessed in a long timescale, *i.e.*, from round 1 to round T. Based on our previous analysis, the probability of accessing unspent transaction output k varies as time goes. We now denote $a_k(t)$ to represent the probability of accessing unspent transaction output k at mining round t in this game. Thereby, the expected cache hit rate for miner i at mining round t can be formulated as $h_i(t) = \sum_{k=1}^{S} z_{ik}a_k(t) / \sum_{k=1}^{S} a_k(t)$. The expected delay and cost at mining round t for miner i are also updated as $d_i(t) = d_e x_i + d_c x_i (1 - h_i(t))$ and $C_i(t) = p_e y_i + p_c x_i (1 - h_i(t))$, respectively. Now, the OP_{MINER} problem in this scenario can be reformulated as in Eq. (6), of which the utility function U_i is a sum over T mining rounds.

Problem 1d ($\text{OP}_{\text{MULTIROUND}}$)**.**

$$\textit{maximize} \quad U_i = \sum\nolimits_{t=1}^{T} R_i \cdot P_i(t) - \sum\nolimits_{t=1}^{T} C_i(t) \tag{6a}$$

$$\textit{subject to} \quad P_i(t) = m_i \left(1 - \beta \sum\nolimits_{j=1}^{n} m_j d_j(t)\right) \tag{6b}$$

$$x_i \geq 0, \quad z_{ik} \in \{0, 1\}, \forall k \in [1, S]. \tag{6c}$$

Corollary 2. $\text{OP}_{\text{MULTIROUND}}$ *can converge to some point(s) where each miner will keep a certain strategy given the fact that all miners simultaneously update their cache contents every* T *rounds.*

The objective function presented in Eq. (6) is a concave function, since it is a sum of T concave functions, as we have proved in Corollary 2. This concavity guarantees the existence of the Nash equilibrium of its corresponding game.

VI. Modeling Access Probability Dynamics

This section seeks to determine how likely a certain unspent transaction output will be accessed in a given mining round. We resort to an educated approximation. Our objective is to find an access probability function $a_k(t, t_b)$ to model how the chance of spending an unspent transaction output k evolves as time t goes, by taking its unique birth time t_b into account. (Note that, both t and t_b represent a certain mining round rather than an exact time point.) Accurately predicting over time the possibility that an unspent transaction output is spent by its owner is out of the scope of this work, since more factors are involved, *e.g.* the amount of this output, its owner's activeness, and even the cybercurrency's market price. We consider proposing a general model to capture the most cases. Thus, we also ignore some special outputs generated by transactions (*i.e.*, coinbase transactions in Bitcoin) that reward block creators, as those outputs by default have a longer waiting time before they become spendable.

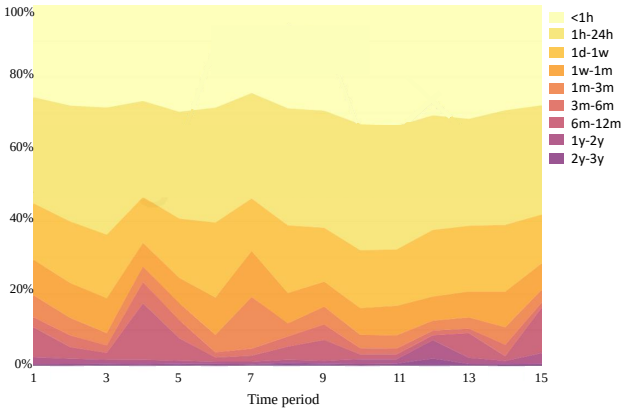


Fig. 3: Daily spent transaction output lifespan within 15 days.

Time period	Median lifespan (day)	Average lifespan (day)
April 01 - April 07	1.60	41.60
April 08 - April 14	1.44	37.33
April 15 - April 21	1.90	50.23
April 22 - April 28	1.36	34.60
April 29 - May 05	1.29	31.72

TABLE II: Comparison of median lifespan and average lifespan every 7 days.

A. Data Collection and Analysis

Our data is collected from Glassnode Studio [7] and Blockchain.info [8]. First, we conduct the following measurement starting from April 1st, 2020. We obtain all spent transaction outputs and their corresponding lifespans, *i.e.*, the duration from the time an output becomes spendable to the time it is confirmed to be spent by its owner, and then present the daily spent transaction output lifespan bands from April 1st to April 15th in Fig. 3. It is obvious that most transaction outputs are spent within 24 hours, but there exist some transactions that stayed in the system for years before they were spent. We further analyze both the median lifespan and the average lifespan of those collected spent transaction outputs. We take 7 days as a period and list the median lifespans and the average lifespans of each period from April 1st to May 5th in Table II. (Note that, outputs with a lifespan of more than 3 months are discarded.) The result indicates that the median lifespan is much shorter than the average lifespan. This observation guides us to fit $a_k(t, t_b)$ with either an exponential distribution or a lognormal distribution.

B. Parameter Fitting and Model Validation

To decide the shape (exponential or lognormal) and corresponding parameter values of $a_k(t, t_b)$, we further collect the birth time and the redeeming time of 5000 spent transaction outputs, as our training dataset. These transaction outputs are divided into 4 traces, each containing 1250 outputs, based on their creation years from 2016 to 2019. We finally decide to use lognormal distribution for the access probability model. (In fact, we also tried exponential distribution as well as Gaussian distribution, and we found lognormal distribution fits best.) We apply a generalized linear model with a log transformation on the access probability. In Fig. 4, we show the potential access

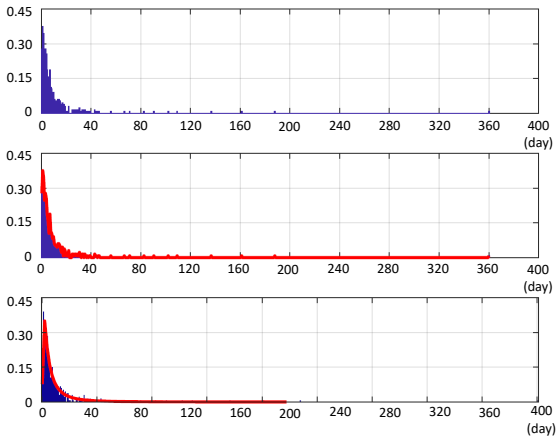


Fig. 4: Access probability function fitting.

probability as a function of time. The first and second pictures are the histogram of the original data and its corresponding frequency polygon, and the last picture is the fitting result. We calculate the squared correlation value and get an average of $R^2 = 0.91$. Now, we can conclude that a lognormal approximation is reasonable and the access probability of different unspent transaction outputs can be different in a given mining round. Therefore, it is necessary for miners to tactically select cache contents for utility optimization.

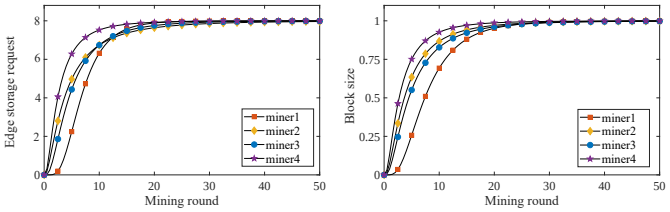
VII. Evaluation

Our evaluation includes two main parts. First, we examine how miners decide their optimal strategies using our proposed algorithm, if we model each single mining round as a one-shot game (Subsection VII.A). We conduct our experiments based on different sets of parameters to show how miners' decisions will be affected by external factors. Second, we take the network settings into consideration and analyze how the number of mining rounds in a one-shot game can influence the achieved equilibrium in our proposed game (Subsection VII.B).

A. Equilibrium in Games of a Single Mining Round

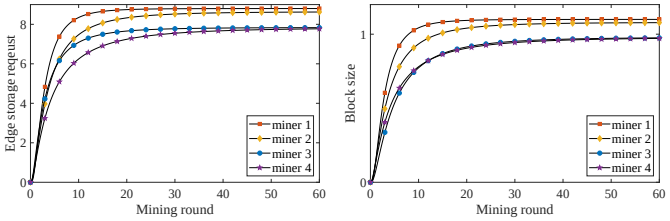
Our experiments evaluate the influence of important parameters on each miner's strategies. We start with a small mobile mining network with 4 homogeneous miners with unlimited budgets. We first consider the simplest case, where we just assume all unspent transaction outputs have the uniform access probability. Then we differentiate miners' mining power in order to make our simulation more realistic. Finally, we focus on how the non-uniform access probability will affect a miner's strategy.

1) Uniform Access Probability: We analyze the results of 4-identical-miner game. Fig. 5 shows how miners' strategies evolve using our proposed algorithm 1. Since miners are identical, their optimal strategies converge to the same point. As we can observe in Fig. 5, the equilibrium is reached after 25 rounds, which is efficient. Next, we move to a heterogeneous miner setting and modify the mining power of each miner as $(m_1, m_2, m_3, m_4) = (0.18, 0.22, 0.3, 0.3)$. Based on the



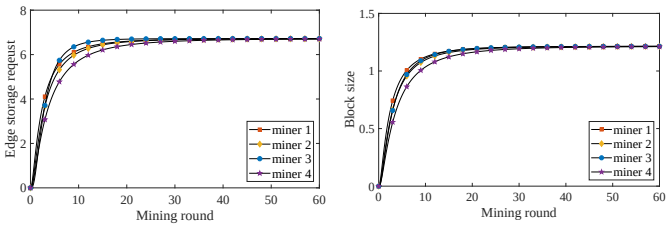
(a) Miners' cache requests (b) Miners' block sizes

Fig. 5: 4 identical miners' single-round repeated mining game.



(a) Miners' cache requests (b) Miners' block sizes

Fig. 6: 4 heterogeneous miners' single-round repeated mining game.



(a) Miners' cache requests (b) Miners' block sizes

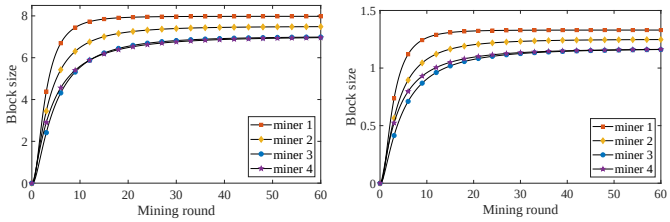
Fig. 7: 4 identical miners' single-round repeated mining game.

results shown in Fig. 6, we could see miner 3 and miner 4 share the same utility since they are identical in this experiment. Meanwhile, we could see miner 1 and miner 2 have a very close utility while the utility of miner 2 is always higher than that of miner 1. This is reasonable as we are discussing a budget-unlimited situation, where the mining power matters. This result is in line with the Bitcoin mining design principle, *i.e.*, more mining power leads to higher profits.

2) Non-uniform Access Probability: We now investigate how the non-uniform access probability will affect miners' strategies. We modify the 4-identical-miner setting by assigning different values to each unspent transaction output and the corresponding result is given in Fig. 7. When comparing with Fig. 5, we observe that miners' strategies have changed. Each miner enlarged his block size while shrinking his cache size. It is reasonable since miners have more information on each output's access probability so that they can filter some outputs since caching them only brings a negative marginal utility. Fig. 8 shows the result of the 4-heterogeneous-miner setting and the observation is similar.

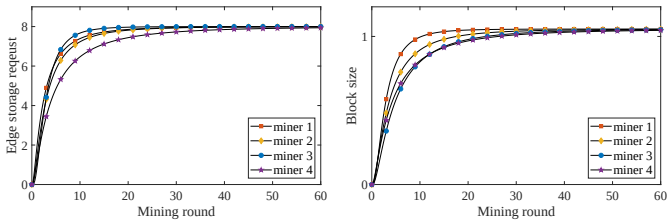
B. Equilibrium in Games with Multiple Mining Rounds

We perform our experiment in the 4-identical-miner setting. We assume that T is fixed as 3. We compare each miner's strategy in Fig. 9 with that in Fig. 5. Since miners are identical, their strategies finally converge to the same point. Obviously, the converge speed becomes slow compared with that in Fig. 5, as lowering update frequency decreases the chance for



(a) Miners' cache requests (b) Miners' block sizes

Fig. 8: 4 heterogeneous miners' single-round repeated mining game.



(a) Miners' cache requests (b) Miners' block sizes

Fig. 9: 4 identical miners' 3-round repeated mining game.

miners to adjust their own strategies. However, we find that in this setting, miners become more aggressive on purchasing resources from the ESP. This is reasonable as since some cached contents will become stale in the processing of the game, miners definitely will get benefit from storing more contents in advance if they don't have budget limitations. For the decision of block sizes, all miners' decisions are nearly the same as the decisions in a single-round game.

VIII. Related Work

1) Scalability Problem of Blockchain Size: Blockchain is an append-only ledger and should be fully replicated by all users in an untrustworthy environment. The exponentially increased blockchain size poses a challenge for its application in the IoT field. Many works focus on the blockchain size reduction. Pruning is a solution that has been implemented in BitcoinCore [9]. A node in its pruned mode only stores the UTXO set and several most recent blocks. [10] uses summary blocks to replace the actual blocks with storage compression. An improved memorization mechanism for the Bitcoin blockchain is proposed in [11]. Another idea is to split the entire blockchain into pieces so that each node only needs to store some of them [12, 13]. We apply a traditional database outsourcing solution by considering both cloud and edge resources.

2) Game Theory in Offloading Mechanism: Game theory is a widely-used model in the field of offloading mechanisms. A large body of existing literature [14–24] focuses on minimizing offloading users' computation overhead in terms of energy and latency. To this end, researchers have developed distributed decision making methodologies. In the field of mobile blockchain mining offloading [25–27], there are few works and most of them are in the PoW-mining scenario where mobile miners only offload their computation to a service provider. Our outsourcing scheme can be viewed as a combination of computation offloading and storage offloading.

3) Blockchain Balance Model: There are two popular models for recording each user's balance in today's blockchain networks. One is the unspent transaction output model, and the other is the account model. The UTXO model is applied by Bitcoin [28]. It can be abstracted as a directed graph of assets moving between users. The account model is used in Ethereum [29]. It is a database reflecting the asset distribution state in the current network. In our work, we focus on the UTXO model and characterize the unspent transaction output's access probability. Some researches have started to analyze the properties of the Bitcoin UTXO set [5, 6].

IX. Conclusion

We consider storage outsourcing as a solution to deal with the storage shortage problem for miners using mobile devices, and then propose a non-cooperative game among miners to obtain optimal storage outsourcing strategies given the existence of both the CSP and the ESP. We analyze how each unspent transaction output's access probability evolves over time and model a single mining round and multiple mining rounds as a one-shot game, respectively. We prove the existence of Nash equilibrium and design a distributed algorithm to achieve NE point(s) for the proposed game. Both numerical evaluation and testbed experiment on Bitcoin are conducted to show the correctness of the proposed access probability pattern and to validate the proposed models and theoretical results. Through our evaluation, we see how different game settings and parameters affect miners' strategies and utilities.

X. Acknowledgements

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

References

[1] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing*.

[2] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay," *Performance Evaluation*, vol. 104, pp. 23–41, 2016.

[3] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[4] F. Giannessi and A. Maugeri, *Variational inequalities and network equilibrium problems*. Springer, 1995.

[5] R. Konrad and S. Pinto, "Bitcoin utxo lifespan prediction," *CS229.stanford.edu*, 2015.

[6] S. Delgado-Segura, C. Pérez-Sola, G. Navarro-Arribas, and J. Herrera-Joancomartí, "Analysis of the bitcoin utxo set," in *International Conference on Financial Cryptography and Data Security*. Springer, 2018, pp. 78–91.

[7] "Glassnode studio." [Online]. Available: https://studio.glassnode.com

[8] "Blockchain.info." [Online]. Available: https://blockchain.info/q

[9] "A full node in pruned mode." [Online]. Available: https://bitcoin.org/en/full-node

[10] A. Palai, M. Vora, and A. Shah, "Empowering light nodes in blockchains with block summarization," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2018, pp. 1–5.

[11] U. Nadiya, K. Mutijarsa, and C. Y. Rizqi, "Block summarization and compression in bitcoin blockchain," in *2018 International Symposium on Electronics and Smart Devices (ISESD)*. IEEE, 2018, pp. 1–4.

[12] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22 970–22 975, 2018.

[13] Y. Xu, "Section-blockchain: A storage reduced blockchain protocol, the foundation of an autotrophic decentralized storage architecture," in *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*. IEEE, 2018, pp. 115–125.

[14] L. Xiao, C. Xie, T. Chen, H. Dai, and H. V. Poor, "A mobile offloading game against smart attacks," *IEEE Access*.

[15] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: a stackelberg game approach," *Computer Networks*.

[16] X. Wang, X. Chen, W. Wu, N. An, and L. Wang, "Cooperative application execution in mobile cloud computing: A stackelberg game approach," *IEEE Communications Letters*.

[17] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*.

[18] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Fog computing in multi-tier data center networks: a hierarchical game approach," in *2016 IEEE international conference on communications*.

[19] T. M. Ho, N. H. Tran, C. T. Do, S. A. Kazmi, T. LeAnh, and C. S. Hong, "Data offloading in heterogeneous cellular networks: Stackelberg game based approach," in *2015 Asia-Pacific Network Operations and Management Symposium*.

[20] Y. Sun, H. Shao, X. Liu, J. Zhang, J. Qiu, and Y. Xu, "Traffic offloading in two-tier multi-mode small cell networks over unlicensed bands: A hierarchical learning framework." *TIIS*.

[21] X. Zhang, L. Guo, M. Li, and Y. Fang, "Social-enabled data offloading via mobile participation-a game-theoretical approach," in *2016 IEEE Global Communications Conference*.

[22] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2017.

[23] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2018.

[24] Q. Xia, W. Liang, Z. Xu, and B. Zhou, "Online algorithms for location-aware task offloading in two-tiered mobile cloud environments," in *Proceedings of the 2014 IEEE/ACM 7th international conference on utility and cloud computing*. IEEE Computer Society, 2014, pp. 109–116.

[25] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Joint computation offloading and content caching for wireless blockchain networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops*.

[26] Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, "Optimal pricing-based edge computing resource management in mobile blockchain," in *2018 IEEE International Conference on Communications*.

[27] S. Jiang, X. Li, and J. Wu, "Hierarchical edge-cloud computing for mobile blockchain mining game," in *Proc. of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS 2019)*, vol. 15, 2019.

[28] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[29] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.