Computation Offloading in Mobile Edge Computing-Enabled Blockchain using Matching and Contract theory

Yijun Li
School of Computer Sciences, Minnan Normal University, Zhangzhou, 363000, Fujian, China

Liming Chen
Zhangzhou Afffliated Hospital of Fujian Medical University, Zhangzhou, 106335, Fujian, China

Wenjie Zhang*
School of Computer Sciences, Minnan Normal University, Zhangzhou, 363000, Fujian, China

Yifeng Zheng
School of Computer Sciences, Minnan Normal University, Zhangzhou, 363000, Fujian, China

Ziqiong Lin
School of Computer Sciences, Minnan Normal University, Zhangzhou, 363000, Fujian, China

ABSTRACT

The application of blockchain technology on mobile devices requires significant computing resources. However, the limited capacity of mobile devices cannot meet the high computing power requirements of blockchain technology, which limits the development of blockchain technology. Mobile Edge Computing (MEC) is capable of alleviate the limited computing capacity of IoT terminals. This paper introduces an edge computing-enabled blockchain system consisting of an edge computing allocation center (ECAC), multiple edge service providers (ESPs), and multiple miners. To efficiently allocate resources between these miners and ESPs, we propose a hierarchical computation offloading strategy based on contract theory and matching game. During contract design phase, ECAC designs a contribution-reward contract to attract ESPs to join the trading market, and provide services to miners. By analyzing the attributes and conditions of feasible contracts of feasible contract, the optimal contract is devised using Lagrange multiplication. During matching phase, an iterative matching algorithm (IMA) is proposed to achieve the matching between ESPs and miners by constructing preference sets for miners and ESPs. Both the stability and convergence have been proved by theoretical analysis. Finally, we conduct experiments to validate the feasibility and effectiveness of the design contract. Experimental results also demonstrate the stability of IMA matching algorithm.

CCS CONCEPTS

• **Networks** → Network performance evaluation; Network performance analysis.

KEYWORDS

Mobile edge computing, Blockchain, Matching theory, Contract theory, Information scenario

ICNCC 2023, December 15–17, 2023, Osaka, Japan

ACM ISBN 979-8-4007-0926-5/23/12
https://doi.org/10.1145/3638837.3638874

ACM Reference Format:
Yijun Li, Liming Chen, Wenjie Zhang*, Yifeng Zheng, and Ziqiong Lin. 2023. Computation Offloading in Mobile Edge Computing-Enabled Blockchain using Matching and Contract theory. In *2023 the 12th International Conference on Networks, Communication and Computing (ICNCC) (ICNCC 2023), December 15–17, 2023, Osaka, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3638837.3638874

1 INTRODUCTION

In recent years, blockchain technology provides a new way of transacting, users can complete simpler and more secure transactions without the need of a third-party trusted organization. The system's security relies on the Proof-of-Work (PoW) mechanism, which requires greater computing power compared to traditional consensus algorithms [1]. The current challenges faced by most mobile terminal devices, such as small size and limited computing power, have restricted the application and development of blockchain technology [2]. To further address the issue of insufficient resources, Mobile Edge Computing (MEC) has been proposed, which provides low-latency and low-energy services by deploying edge servers at the network's edge [3].

Extensive research has drawn attention to the integration of MEC and blockchain technology [4]. These works mainly focus on investigating the security of computation offloading by assuming that each edge server is willing to provide services. The important issues such as server competition and the pricing of computing resources are largely ignored in these literatures. Some works have studied the computing resource of edge servers pricing problem using game theory and auctions in a complete information scenario [5]. However, the miners may retain some private information to ensure their transaction security, which increases the complexity of the computing resource pricing problem. Auction algorithms and game theory are not applicable for analyzing information incompleteness [6], especially when the computing resources are limited, ESPs might not have sufficient computing power for conducting auctions and facilitating cooperation. Therefore, how to address the resource allocation and pricing issues in edge computing-enabled blockchain systems in the scenarios of incomplete information still needs further investigation.

Our previous work combined MEC technology and blockchain, and addressed the computation offloading problem in edge-enable blockchain system with single ESP in incomplete information scenario [7]. This paper further investigates the computation offloading problem among multiple ESPs and miners. Compared to single ESP, the problem posed by the system with multiple ESPs is more complex and faces the following challenges [8]: Firstly, both the miners and ECAC don't know the information about the types and computing resources of ESPs. Similarly, ESPs are unaware of the profits they can earn when trading begins. Secondly, ESPs will compete with each other to provide services to miners when multiple ESPs join in the ECAC service market. In addition, the computing resources required for miners to solve PoW puzzles are uncertain, and the computing capacity and total service time provided by each ESP are also different. The task completing delay has a direct relationship with the computing resource allocated. Thus, different matches between ESP and miner will result in different performance. Therefore, how to achieve the optimal matching between ESPs and miners is the focus of this work.

To achieve this, this paper considers an edge computing-enabled blockchain system involving multiple ESPs and miners, and proposes a hierarchical computation offloading strategy using contract and matching theory to efficiently allocate resources between miners and ESPs. In contract design phase, ECAC sets a contribution-reward contract to attract ESPs to participate in cooperation after receiving resource requests from miners. By analyzing the individual rationality (IR) and incentive compatibility (IC) conditions of feasible contracts, the optimal contract is obtained. In matching phase, the miners prefer to choose ESPs with higher computing power, while ESPs prefer miners who require less service time. The optimal matching is achieved by designing an iterative matching algorithm (IMA) based on the preference sets for ESPs and miners. The effectiveness of the optimal contract and the stability of the matching results for the proposed IMA algorithm are verified through experimental results.

2 SYSTEM MODEL

We consider a edge computing-enabled blockchain system comprising an ECAC, multiple ESPs, and multiple miners, as shown in Figure 1. Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be the set of ESPs, and $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of miners. Due to limited computing resources, the miners need to purchase resources from ECAC to solve the PoW problem. ECAC acts as a resource information dissemination center, it can receive requests from miners but lacks the service resources to meet these demands. On the other hand, ESPs have some under-utilized free resources, ECAC will attract ESPs to join the trading market and share resources with miners. Once the miners achieve cooperation intention with ESPs via ECAC, miners can offload their computing tasks to the corresponding ESPs for processing.

However, since both ESPs and miners have some private information, this incomplete information scenario makes the resource allocation problem in the transaction market of high challenge. This work proposes a hierarchical computation offloading strategy based on contract and matching theory to solve the resource allocation problem between ESPs and miners in incomplete information. On contract phase, ECAC encourages ESPs to join resource trading market by creating a service-reward contract. On matching phase,

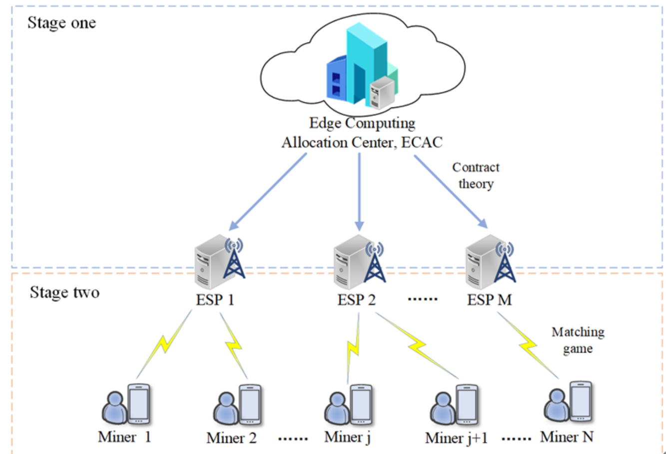

Figure 1: Edge Computing-Enabled Blockchain System

an optimal matching is achieved based on the preference sets. Once the two parties reach agreement, ESPs who have signed the contract need to provide computing resources to the miners. We assume that one miner can only associate with a ESP, while one ESP can serve multiple miners.

2.1 Interaction of ECAC and ESPs

Based on the computing power and service time that ESP i can provide, the concept of type γ_i is introduced to indicate ESP's willingness to join ECAC. The larger the value of γ_i, the higher the ESP type, the stronger the willingness to join ECAC. We assume that ESP types are arranged in ascending order:

$$\gamma_1 < \gamma_2 < \ldots < \gamma_i < \ldots < \gamma_M$$

ECAC only knows information about the distribution of ESP types. We assume that the probability of ESP belonging to type γ_i is denoted as p_i, where $0 \le p_i \le 1$ and $\sum_{i=1}^{M} p_i = 1$. In the context below, we will refer to an ESP of type γ_i as ESP i. Based on contract theory, ECAC designs a contribution-reward contract for the ESPs, denoted as (T_i, R_i). Here, T_i represents the computing service time that ESP i can provide, and R_i denotes the corresponding reward paid to ESP i by ECAC, $0 \le R_i \le R_{i,max}$. Contract is established by ECAC for different types of ESPs, and ESPs decide whether to accept the contract.

ECAC utility model: when ESP i agrees to provide service to the miners under a contract, the utility function of ECAC is formulated as follows:

$$Z_i = C_i - R_i \tag{1}$$

where $C_i = \beta_i T_i^2$ denotes the expected revenue that ECAC can get by providing service time T_i, and β_i is a service valuation related to the contract price. However, since ECAC is unaware of the specific types of ESPs and the exact service time allocated to miners, the expected utility of ECAC can be expressed as follows:

$$Z = \sum_{i=1}^{M} p_i \left(\beta_i T_i^2 - R_i\right) \tag{2}$$

ESPs utility model: the utility function of ESP i is defined as follows:

$$V_i = \gamma_i g\left(R_i\right) - cT_i^2 \tag{3}$$

where $g(R_i) = k\ln(1+R_i)$ is the valuation function of rewards paid to ESP i by ECAC.

2.2 Interaction of ESPs and miners

After contract phase, ECAC knows information of ESPs, including the overall service time T_i, and the services amount provided per unit time, denoted as a_i. Then, the total service volume that ESP i can provide is calculated as $a_i \times T_i$. During matching phase, miners will retain private information. The service requirement for each miner is denoted as x_j. Let t_{ij} be the service time allocated to miner j by ESP i, the conditions for miner j to accept ESP i is $t_{ij} \times a_i \geq x_j$. Additionally, compared to the overall computing power, the relative computing power of miner $j \in \mathcal{N}$ is expressed as ℓ_{ij}

$$\ell_{ij} = \frac{a_i \times t_{ij}}{\sum_{j\in\mathcal{N}} x_j} \tag{4}$$

Miners compete with each other to solve the PoW puzzle in order to receive payment. The first miner who successfully mines and reaches a consensus receive reward. The reward for the first successful miner is $r \times m_j$, where r represents the designated reward factor. m_j is the block size indicating the number of transactions contained in the blocks mined by miner j. Let G denote the fixed cost incurred by miners during mining operations, then the utility of miner j, aligned with ESP i, denoted as U_{ij}, can be defined as follows:

$$U_{ij} = rm_j P\left(\ell_{ij}, m_j\right) - C_{ij} - G \tag{5}$$

where $C_{ij} = \beta_i t_{ij}^2$ denotes the cost that miner j needs to pay for getting service time from ESP i. Same as [3], we assume $P(\ell_{ij}, m_j)$ represents the probability of successful mining for miner j and can be formulated as:

$$P\left(\ell_{ij}, m_j\right) = \ell_{ij} e^{-\lambda z m_j} \tag{6}$$

By substituting Eq. (6) into Eq. (5), we have:

$$U_{ij} = \frac{rm_j a_i t_{ij} e^{-\lambda z m_j}}{\sum_{j\in\mathcal{N}} x_j} - C_{ij} - G \tag{7}$$

Let $\theta_j = \frac{rm_j e^{-\lambda z m_j}}{\sum_{j\in\mathcal{N}} x_j}$ represent the type of miner j, which involves all the private information of miner j. We assume that there are total N types of miners, denoted as $\theta_1, \theta_2, \ldots, \theta_j, \ldots, \theta_N$. Furthermore, the utility function for a miner belonging to type θ_j can be formulated as follows:

$$U_{ij} = \theta_j a_i t_{ij} - \beta_i t_{ij}^2 - G \tag{8}$$

3 CONTRACT DESIGN BETWEEN ECAC AND ESP

3.1 Optimal contract design problem

A feasible contract $\{(T_i, R_i), \forall i \in \mathcal{M}\}$ must satisfy the IR and IC conditions [9]. From IC constraint, it can be seen that ESP belonging to type γ_i will choose the contract item designed for its type. Thus, the contract-based incentive mechanism is a truth-telling mechanism, ECAC can know the type information of ESPs after signing the contract. With the goal of maximizing ECAC's utility while satisfying both IR and IC constraints, and keeping the reward bound hold, the optimal contract design problem can be expressed as follows:

$$\begin{aligned} \text{P1}: \quad & \max_{\{T_i,R_i\}} \sum_{i=1}^{M} p_i\left(\beta_i T_i^2 - R_i\right) \\ s.t. \quad & 0 \leq R_i \leq R_{i,max}, \\ & \gamma_i g\left(R_i\right) - cT_i^2 \geq 0, \forall i \in \mathcal{M}, \\ & \gamma_i g\left(R_i\right) - cT_i^2 \geq \gamma_i g\left(R_{i'}\right) - cT_{i'}^2, \forall i, i' \in \mathcal{M}, i \neq i'. \end{aligned}$$

The optimization problem in **P1** involves M IR constraints and M^2 IC constraints. It will bring great complexity to solve this optimization problem directly.

3.2 Feasible contract

Based on the definitions of IR constraint and IC constraint, we can obtain crucial attributes of a feasible contract. The first characteristic of a feasible contract is denoted by Lemma 1.

Lemma 1 *For any feasible contract* $\{(T_i, R_i), \forall i \in \mathcal{M}\}$, $R_i > R_{i'}$ *if and only if* $\gamma_i > \gamma_{i'}, \forall i, i' \in \mathcal{M}, i \neq i'$.

From Lemma 1, higher-type ESPs will obtain higher rewards. In other words, if two ESPs belong to the same type, then the reward they can receive must be the same. In this case, the reward strictly increases with type, i.e., higher-type ESPs are more willing to sign service contract with ECAC than lower-type ESPs.

Lemma 2 *For any feasible contract* $\{(T_i, R_i), \forall i \in \mathcal{M}\}$, $R_i > R_{i'}$ *if and only if* $T_i > T_{i'}, \forall i, i' \in \mathcal{M}, i \neq i'$.

As stated in Lemma 2, ESPs who contribute more service time to ECAC will receive more rewards, while ESPs will receive the same reward by providing the same service time.

Lemma 3 *For any feasible contract* $\{(T_i, R_i), \forall i \in \mathcal{M}\}$, *if* $\gamma_i > \gamma_{i'}$, *then the utility of ESPs must satisfy* $V_i > V_{i'}, \forall i, i' \in \mathcal{M}, i \neq i'$.

From Lemma 3, it is clear that the lowest type of ESPs has the lowest utility. Thus, when the IR conditions for the lowest type of ESPs hold, all higher types of ESPs also satisfy the IR constraints. Thus, the IR constraints can be simplified as

$$\gamma_1 g\left(R_1\right) - cT_1^2 \geq 0 \tag{9}$$

Since the utility of ECAC increases with T_i and decreases with R_i, if there exists a contract item (R_i, T_i) such that $V_i = \gamma_i g(R_i) - cT_i^2 > 0$, then ECAC can adjust the contract by decreasing R_i to raise its utility until $V_i = \gamma_i g(R_i) - cT_i^2 = 0$. Thus, to maximize the expected utility of ECAC, the rewards given by ECAC to ESPs should be as small as possible, so Eq. (13) can be further replaced by

$$\gamma_1 g\left(R_1\right) - cT_1^2 = 0 \tag{10}$$

Lemma 4 *The IC constraint can be reduced to the Local Downward Incentive Compatible (LDIC) condition*

$$\gamma_i g\left(R_i\right) - cT_i^2 \geq \gamma_i g\left(R_{i-1}\right) - cT_{i-1}^2, \forall i \in \{2, 3, \ldots, M\}$$

and Local Upward Incentive Compatible (LUIC)

$$\gamma_i g\left(R_i\right) - cT_i^2 \geq \gamma_i g\left(R_{i+1}\right) - cT_{i+1}^2, \forall i \in \{1, 2, \ldots, M-1\}$$

3.3 Optimal contract

According to the attributes of feasible contract and the simplification of IR and IC constraints, the original optimization problem **P1** can be converted as **P2**:

$$
\begin{aligned}
\text{P2}: \quad & \max_{\{T_i, R_i\}} \sum_{i=1}^{M} p_i \left(\beta_i T_i^2 - R_i\right) \\
s.t. \quad & 0 \le R_i \le R_{i,max}, \\
& \gamma_1 g\left(R_1\right) - cT_1^2 = 0, \\
& cT_i^2 = cT_{i-1}^2 + \gamma_i \left(g\left(R_i\right) - g\left(R_{i-1}\right)\right), i = 2, ..., M.
\end{aligned}
$$

Theorem 1 Given contract incentives R_i, with $0 \le R_1 < \ldots < R_i < \ldots < R_M$, the optimal service time is given by

$$
\left(T_i^*\right)^2 = \begin{cases} \left(T_{i-1}^*\right)^2 + \frac{\gamma_i}{c}\left(g\left(R_i\right) - g\left(R_{i-1}\right)\right), i = 2, 3, \ldots, M \\ \frac{\gamma_i}{c} g\left(R_i\right), i = 1 \end{cases} \tag{11}
$$

All proofs can refer to our technical report in [10].

Substituting T_i^* into **P2**, we can obtain a relaxed optimal problem with R_i, as follows:

$$
\begin{aligned}
\text{P3}: \quad & \max_{R_i} \sum_{i=1}^{M} F\left(R_i\right) \\
s.t. \quad & 0 \le R_i \le R_{i,max}
\end{aligned}
$$

where $F(R_i) = p_i(\beta_i(\frac{\gamma_1}{c} g(R_1) + \sum_{s=1}^{i} {}_s) - R_i)$ and ${}_s = \{ \begin{matrix} 0, s = 1 \\ \frac{\gamma_s}{c}(g(R_s) - g(R_{s-1})), 1 < s \le i \end{matrix}$.

P3 is the optimization problem with single variable. In this case, the computational complexity can be effectively reduced. By using Lagrange multiplication, we can get the solution of P3, i.e., R_i^*, given by

$$
R_i^* = \frac{cR_{i,max} - y - c + \sqrt{(y-c)^2 + cR_{i,max}\left(2c - 2y + cR_{i,max}\right)}}{2c}, \quad \forall i \in \mathcal{M} \tag{12}
$$

where $y = p_i \beta_i \gamma_i - p_{i+1} \beta_{i+1} \gamma_{i+1}$.

The expressions of service time and reward given in Eq. (11) and Eq. (12) form the optimal contract, denoted as $\{(T_i^*, R_i^*), \forall i \in \mathcal{M}\}$. In the following matching stage, the specific types of ESPs are known, but the specific types of miners remain unknown.

4 MATCHING STAGE

The subsequent matching step can be triggered after obtaining the optimal contract. A binary variable ρ_{ij} is introduced with $\rho_{ij} = 1$ denoting that miner j is served by ESP i, otherwise $\rho_{ij} = 0$. The matching problem is NP hard due to 0-1 integer variables involving and the interaction between ESPs and miners. In this paper, we construct the model following the classical admissions market, where the miners play as students and the ESPs can be considered as schools. A student can only choose one school to attend, while a school can accept multiple students. In this case, the demand and service relationship between miners and ESPs can be modeled as a multi-to-one matching problem. The optimization goal is to maximize the miner's profit, the matching problem between ESPs and miners can be formulated as:

$$
\max_{\rho_{ij}} \sum_{i=1}^{M} \sum_{j=1}^{N} \rho_{ij} \left(\theta_j a_i t_{ij} - \beta_i t_{ij}^2 - G\right) \tag{13}
$$

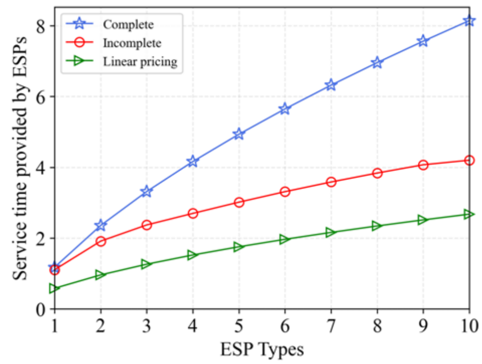

Figure 2: Comparison of service time provided by ESPs

$$
\begin{aligned}
s.t. \quad & \text{a)} \sum_{j \in \mathcal{N}} \rho_{ij} t_{ij} \le T_i, \forall i \in \mathcal{M}, \\
& \text{b)} a_i \times \rho_{ij} t_{ij} = x_j, \forall i \in \mathcal{M}, j \in \mathcal{N}, \\
& \text{c)} \sum_{i \in \mathcal{M}} \rho_{ij} \le 1, \forall j \in \mathcal{N}, \\
& \text{d)} \rho_{ij} = \{0, 1\}, \forall i, j.
\end{aligned}
$$

1) Preferences of miners: From the perspective of miners, each miner want to choose the ESP that maximizes its utility, the preference of miner j for ESP i denoted as PL_{ij}^M, can be defined as:

$$
PL_{ij}^M = \frac{\theta_j a_i}{2\beta_i}, \forall i, j \tag{14}
$$

Sort each row of the preference matrix $PL^M = [PL_{ij}^M]_{N \times M}$ in descending order, then the j^{th} row PL_j^M represents the preference of miner j for all ESPs.

2) Preferences of ESPs: An ESP always want to associate with a miner that requires smaller service time. Therefore, the preference of ESP i for miner j can be defined as:

$$
PL_{ij}^E = t_{ij} = \frac{x_j}{a_i}, \forall i, j \tag{15}
$$

Let $PL^E = [PL_{ij}^E]_{M \times N}$ denote the preference set of ESPs. Similarly, the preference of ESP i PL_i^E can be obtained by sorting the i^{th} row of PL^E in ascending order.

In multi-to-one matching, ECAC aims to seek a stable match between all miners and ESPs. In two finite and disjoint sets $\mathcal{M}$ and $\mathcal{N}$, when $\mu_s^e(i) = j$ and $\mu_s^m(j) = i$ do not unilaterally change due to external factors, the matching $\{\rho_{ij}\}_{M \times N}$ is stable. The multi-to-one matching model analyzed in this paper is IMA based on the traditional Gale-Shapley (GS) algorithm, which generates stable matching results based on the mutual preference sets between ESPs and miners. The algorithm is executed iteratively when the preferences of each miner and ESP are known. Then the miners will update their preferences during the iteration. The multi-to-one matching algorithm is outlined in detail in Algorithm 1.

5 NUMERICAL RESULTS AND ANALYSIS

In the section, numerical simulation is performed to evaluate the effective of contract design and matching algorithm. We consider a single ECAC, ten different types of ESPs, and twenty different types of miners.

Algorithm 1 Iterative Matching Algorithm, IMA

Input: M, N, PL^M, PL^E;
Output: stable matching results $\{\rho_{ij}\}_{M\times N}$ between ESPs and miners;
Initialization: set $\{\mu_s^e(i)\} = \emptyset$, $\{\mu_s^m(j)\} = \emptyset$, $flag = 0$, $\{\rho_{ij}\}_{M\times N} = 0$;
Construct PL^M, PL^E according to Eqs. (14) and (15);
while $flag = 0$ do
  for miner $j \in \mathcal{N}$ and $\{\mu_s^m(j)\} = \emptyset$ do
    if $PL_j^M = \emptyset$ then
      Miner j exits the match and $\mathcal{N} = \mathcal{N} \setminus j$;
    else
      Send matching request to the first ESP i in PL_j^M, then move the already requested ESP i to the end of PL_j^M and assign a value i to $\mu_s^m(j)$;
    end
  end
  for ESP $i \in \mathcal{M}$ do
    if Miner j sends a matching request to ESP i then
      if $\sum_{j\in\mathcal{N}} \rho_{ij} t_{ij} \le T_i$ then
        Add miner j to $\{\mu_s^e(i)\}$ and sort the miners in $\{\mu_s^e(i)\}$ according to the preference list PL_i^E in descending order;
      else if $j \succ \mu_s^e(i)$ and $\sum_{j\in\mathcal{N}} \rho_{ij} t_{ij} \le T_i$ then
        Reject the last miner in $\{\mu_s^e(i)\}$ and add miner j to $\{\mu_s^e(i)\}$, then sort the miners in $\{\mu_s^e(i)\}$ according to preference list PL_i^E in descending order;
        Set $\{\mu_s^m(j)\} = \emptyset$ for the rejected miner;
      end
    end
  end
  if $\{\mu_s^e(i)\}$ keeps unchanged then
    $flag = 1$;
    if $\mu_s^e(i) = j$ and $\mu_s^m(j) = i$ then
      $\rho_{ij} = 1$;
    end
  end
end
return the matching results $\{\rho_{ij}\}_{M\times N}$.

Figure 2 illustrates the relationship between the service time allocated by ESPs and the types of ESPs under three scenarios: complete information, incomplete information, and linear pricing. As depicted in Figure 2, the amount of service time provided by ESPs increases with their types. This observation is consistent with the analysis presented in Corollary 4. From Figure 2, we can also observe that ESPs are capable of providing the highest amount of service time in complete information scenario. This is because that ECAC knows comprehensive information about ESPs and can fully utilize this information to design reasonable pricing strategies to motivate the ESPs to provide service time furthest. On the other hand, the linear pricing scheme exhibits the lowest contribution, as the linear pricing mechanism usually employs a fixed pricing structure, lacking the ability to dynamically adapt and adjust the service time of ESPs.

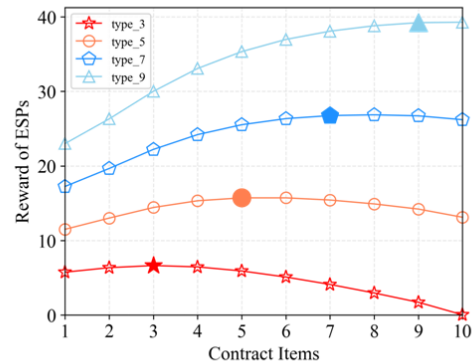

Figure 3: IR and IC constraint verification

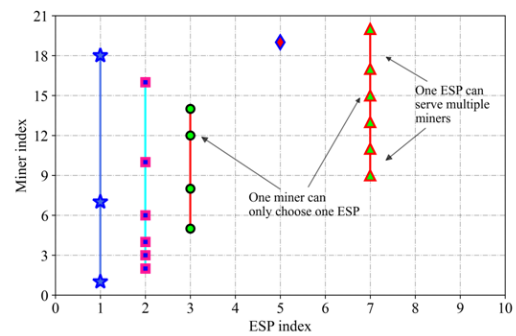

Figure 4: Example of matching results

Figure 3 validate the IC and IR constraints. The utilities of types ESPs $\{3,\ 5,\ 7,\ 9\}$ are illustrated. It can be observed that ESPs can maximize their utility by choosing contract items that design for their respective types. This observation proves that the valid of IC constraints. Additionally, it is obvious that ESPs can achieve non-negative utility by selecting different contract items, which verifies the satisfaction of IR constraints. Consequently, when ESPs choose contract items designed for their types, ECAC can know the specific types information of ESPs. The incentive mechanism can effectively overcome the incomplete information between ECAC and ESPs. Furthermore, Figure 3 also highlight that the higher-type ESPs can get higher utility than the lower-type ESPs, thereby proving the findings of Lemma 1.

5.1 Matching algorithm evaluation

Figure 4 display a matching result of ten ESPs and twenty miners. The preferences of miners are related to the computing power of ESPs, and different ESPs have different computing power. However, a high-type ESPs represent that the ESPs can provide more service time, rather than have higher computing power. From Figure 4, we can observe that some ESPs may not be chosen by any miner due to low preference, such as ESPs indexed as $\{4, 6, 8, 9, 10\}$ in Figure 4. Furthermore, as illustrated in the figure, a miner can select only one ESP, whereas an ESP can provide service to multiple miners.

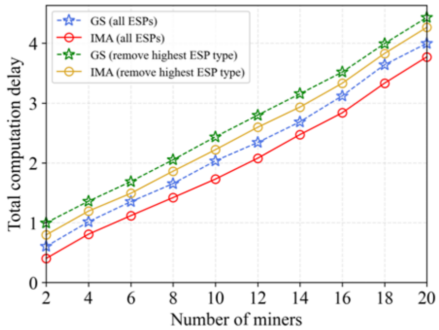

Figure 5: Computing latency comparison

The comparison of computing latency between GS and IMA matching algorithm is illustrated in Figure 5. Two scenarios have been considered. The first scenario involves all types of ESPs to serve miners. The other scenario removes the highest type of ESP, i.e., the ESP with greatest computing power does not participate in matching. It's obvious that in both cases, the total computing latency increases as the number of miners grows. Moreover, the result in Figure 5 also show that the computing latency can be further reduced by adding the highest type of ESP. Therefore, designing an incentive mechanism to attract ESPs, especially high-type of ESPs, to join ECAC has a significant impact on the system's performance.

6 CONCLUSION

This paper considers the computation offloading problem for an edge computing-enable blockchain system involving multiple ESPs and miners. By introducing ECAC, the proposed computation offloading strategy can be divided into two phases. In contract design phase, ECAC formulates contract to attract different types of ESPs to participate in the matching market. ESPs decide whether to accept a contract item and provide service to miners. By analyzing the necessary and sufficient conditions for a feasible contract, the optimal contract that maximizes ECAC's utility can be obtained using Lagrange multiplication under incomplete information. The objective of matching phase is to maximize the miners' utility by optimal matching between ESPs and miners. The paper proposes an IMA based on GS to determine the demand-service matching between ESPs and miners by setting the preference sets. The feasibility and optimality of the contract, as well as the stability and convergence of the IMA matching algorithm have been validated via simulation results.

ACKNOWLEDGMENTS

This work is supported by the Nature Science Foundation of Fujian Province (Grant No.2021J011002, No.2021J011004), High level cultivation projects of Minnan Normal University (Grant No. MSGJB2021007).

REFERENCES

[1] Zehui Xiong, Shaohan Feng, Dusit Niyato, Ping Wang and Zhu Han. 2018. Optimal pricing-based edge computing resource management in mobile blockchain. 2018 IEEE International Conference on Communications (ICC). IEEE, Kansas City, MO, USA, 1-6.

[2] Marko Vukolić. 2016. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In: Camenisch, J., Kesdoğan, D. (eds) Open Problems in Network Security. iNetSec 2015. Lecture Notes in Computer Science, Springer, Cham, 9591, 112-125.

[3] Satyanarayanan Mahadev. 2017. The emergence of edge computing. Computer. 50, 1 (January 2017), 30-39.

[4] Shafkat Islam, Shahriar Badsha, Shamik Sengupta, Hung La, Ibrahim Khalil and Mohammed Atiquzzaman. 2021. Blockchain-enabled intelligent vehicular edge computing. IEEE Network. 35, 3 (June 2021), 125-131. https://doi.org/10.1109/MNET.011.2000554

[5] Alia Asheralieva and Dusit Niyato. 2019. Learning-based mobile edge computing resource management to support public blockchain networks. IEEE Transactions on Mobile Computing. 20, 3 (March 2021), 1092-1109. https://doi.org/10.1109/TMC.2019.2959772

[6] Junfei Wang, Jing Li, Zhen Gao, Zhu Han, Chao Qiu and Xiaofei Wang. 2023. Resource management and pricing for cloud computing based mobile blockchain with pooling. IEEE Transactions on Cloud Computing. 11, 1 (March 2023), 128-138.

[7] Yijun Li, Ziqiong Lin, Wenjie Zhang, Yifeng Zheng and Jingming Yang. 2022. Mobile edge computing-enabled blockchain: contract-guided computation offloading. The Journal of Supercomputing. 79, 7 (December 2022), 7970-7996. https://doi.org/10.1007/s11227-022-04990-7

[8] Kun Xie, Xin Wang, Gaogang Xie, Dongliang Xie, Jiannong Cao, Yuqin Ji and Jigang Wen. 2019. Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing. IEEE Transactions on Services Computing. 12, 6 (2019), 925-940.

[9] Lin Gao, Xinbing Wang, Youyun Xu and Qian Zhang. 2011. Spectrum trading in cognitive radio networks: a contract-theoretic modeling approach. Selected Areas in Communications. 29, 4 (April 2011), 843-855. https://doi.org/10.1109/JSAC.2011.110415

[10] https://pan.baidu.com/s/1PnigWtcZ-5IPb4uc5Ae_HQ?pwd$=$yruq