Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Efficient Resource Allocation for Blockchain-Enabled Mobile Edge Computing: A Joint Optimization Approach

MOEIN VALITABAR[1], MOHAMMAD FATHI[2], and Keivan Navaie,[3], (Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Engineering, University of Kurdistan, Sanandaj 66177-15177, Iran (e-mail: moein.valitabar@uok.ac.ir)
[2]Department of Electrical Engineering, Faculty of Engineering, University of Kurdistan, Sanandaj 66177-15177, Iran (e-mail: mfathi@uok.ac.ir)
[3]Lancaster University, LA1 4WA, UK (e-mail: k.navaie@lancaster.ac.uk)

Corresponding author: Mohammad Fathi (e-mail: mfathi@uok.ac.ir).

ABSTRACT This paper addresses the critical challenge of optimizing resource allocation for task offloading in blockchain-enabled mobile edge computing (MEC), aiming to minimize total energy consumption within wireless networks equipped with edge servers (ESs). We propose a novel joint resource allocation framework that integrates task offloading and blockchain processes within MEC, formulating it as a mixed-integer nonlinear programming (MINLP) problem. The solution assigns mobile terminals (MTs) to ESs and optimally allocates computational resources at ESs for both task computation and block generation. To manage the complexity of this optimization, we employ a two-stage dual decomposition approach. Initially, the problem is separated into subproblems for MEC and blockchain functionalities. These subproblems are further decomposed across ESs and MTs, enabling us to derive analytical solutions for optimal computational frequency allocation for both task offloading and blockchain operations. Leveraging these insights, we develop two low-complexity algorithms, which utilizes a greedy assignment strategy for MTs to ESs, and optimally allocates computational frequencies within the MEC and blockchain components. Performance evaluation results demonstrate the effectiveness of these algorithms, achieving significant reductions in total energy consumption while maximizing the efficiency of communication and computational resources. The approach also contributes to reducing network outage probability. This work presents a promising framework for developing resource-efficient blockchain-enabled MEC systems, positioning it as a scalable solution for future wireless networks.

INDEX TERMS Mobile edge computing, blockchain, optimization, task offloading, resource allocation.

I. INTRODUCTION

MOBILE terminals (MTs) have recently encountered computation-intensive applications like virtual reality, IoT services, online interactive games, speech, and face recognition [1]. To address these challenges, mobile edge computing (MEC) has emerged as a promising solution. MEC involves offloading tasks from MTs to nearby servers called edge servers (ESs). This offloading strategy not only conserves the computational resources of MTs but also ensures that offloaded applications receive the required quality of service, particularly tolerable latency for delay-sensitive applications [2].

In contrast to centralized cloud computing, where cloud servers are far from MTs, leading to network congestion and high offloading delays, MEC offers several advantages. Firstly, MEC reduces the distance between the cloud and MTs, thereby enhancing overall bandwidth throughput. Secondly, MEC enables MTs to communicate with ESs through cellular communication protocols, further improving the efficiency of the network. Additionally, the number of ESs in MEC can be increased compared to centralized cloud computing, providing a more robust and scalable solution.

Blockchain, alongside MEC, has been proposed to enhance the security and privacy of task offloading [3]. Blockchain is a distributed digital ledger that facilitates secure, transparent, and tamper-proof transactions within a network. This ensures that data and parameters on task offloading remain unmodified and protected from eavesdropping, utilizing security-based operations within the blockchain system. Data and parameters are stored as transaction records on the blockchain. Unlike traditional centralized ledgers, blockchain operates as a distributed ledger system, maintaining transactions on a peer-to-peer network [4]. In essence, transactions are replicated across multiple nodes in a transparent, traceable, and secure manner. For instance, blockchain-enabled MEC can provide a secure framework for computing-based applications such as decentralized machine learning [5], crowd-intelligence ecosystem [6], security management [7], smart grid [8], and smart city applications [9].

The distributed structure of both MEC and blockchain drives the integration of these technologies to meet the quality of service requirements in next-generation services, as highlighted in [10]. In the context of MEC, transactions could encompass identifiers of MTs, functions of offloaded tasks, results returned to MTs, and so on. The optimization of resource allocation in a MEC system integrated with a blockchain system is a research area that has been explored in the literature. Researchers have addressed this problem using various techniques, including reinforcement learning (RL), game theory, heuristic approaches, and optimization methods.

A. RELATED WORK

A joint optimization of task offloading and user privacy preservation has been modeled as a Markov decision process in [11], [12] and a deep RL algorithm has been developed to make the best decisions. Similarly, in [13], a deep RL-based computation offloading scheme has been proposed to ensure the reliability and efficiency of vehicle-to-vehicle task offloading. A collective RL algorithm has also been adopted in [14] to adaptively allocate resources. An online learning-based secure task offloading algorithm has been proposed in [15] to learn the long-term optimal strategy of vehicles.

Furthermore, in [16], a deep RL-based algorithm has been developed to derive task offloading decisions for a multi-user with multi-task MEC system. A cost minimization scheme as a weighted sum of energy and delay has been proposed using DDQN in [17]. A distributed RL has also been proposed in [18] for edge computing based on a consortium blockchain. However, deep RL-based offloading approaches face challenges such as slow convergence and stability due to the high-dimensional action space and time-varying environments. In the training process, each agent only observes its local information without access to information from other agents.

An auction based task offloading of MTs has been modeled as a Stackelberg game in [19] for blockchain-enabled edge computing. A game was adopted to model interactions between an edge cloud operator and MTs to obtain the optimal resource price and devices' resource demands. n [20], a cooperative game-theoretic solution was developed to model the competition among MTs in offloading and mining as a potential game. Despite these contributions, auction-based and game-theory approaches lack the ability to adapt to changing environments and, therefore, cannot achieve long-term performance.

Resource allocation for blockchain-based MEC has also been addressed using analytical and heuristic optimization techniques. In [21], a joint resource allocation optimization to minimize a weighted sum of ESs consumption energy and latency has been formulated as mixed-integer nonlinear programming problem. In [22], a blockchain-based video streaming system with MEC has also been provided in [22] to design an incentive mechanism to facilitate collaboration among content creators, video transcoders, and consumers.

Resource allocation for blockchain-based MEC has also been addressed using analytical and heuristic optimization techniques. In [21], a joint resource allocation optimization problem was formulated to minimize a weighted sum of energy consumption and latency of edge services. In [22], a blockchain-based video streaming system with MEC was proposed to design an incentive mechanism for collaboration among content creators, video transcoders, and consumers. A non-cooperative game has also been employed to model MEC-based mobile blockchain networks in [23]. A new consensus algorithm called IPBFT for task offloading has also been proposed in [24].

Furthermore, a non-dominated sorting genetic algorithm was adopted in [25] to generate strategies for balanced task offloading of IoT devices. In [26], the Lyapunov optimization technique was applied to control computation and communication costs in a blockchain-enabled IoT-edge-cloud computing architecture. In [27], task offloading from IoT devices to MEC and a private blockchain via aided drones was investigated. However, heuristic optimization techniques do not guarantee the optimal solution, while analytic optimization techniques suffer from high computational complexity.

B. CONTRIBUTION

The integration of MEC and blockchain as computation and communication-intensive systems has been proposed in the literature. In particular, tasks offloaded to a MEC system require efficient resource allocation to meet the quality of service requirements of applications while minimizing energy consumption. On the other hand, in the blockchain system, transactions are grouped into units called blocks and stored in network nodes. This also necessitates the allocation of computing resources for block generation. Both systems involve resource-intensive operations, but their resource allocation designs have been addressed separately in the literature [4], [12], [28]. This approach may lead to suboptimal solutions. Notably, blockchain has been considered as an overlay system above MEC.

The unique design of MEC and blockchain leads to inefficient resource allocation throughout the system. This results in high latency for task offloading and block generation, as well as high energy consumption. To address this gap, we present a joint optimization approach that optimizes both computation and communication resources for MEC and blockchain systems.

In particular, our work is different in that we not only consider the energy consumption of ESs but also the energy consumption of MTs for data transmission to ESs via wireless channels. This is crucial because MTs are battery-powered, and minimizing their energy consumption should be a part of the objective function for resource allocation. Additionally, we extend the previous work by considering a maximum tolerable delay associated with each task completion and a maximum tolerable delay for block generation in the blockchain system. Here are the key contributions of this paper:

1) We derive the offloading delay and energy consumption associated with task offloading and block generation within the context of blockchain-enabled MEC.
2) We model the joint allocation optimization of communication and computation resources for blockchain-enabled MEC to minimize total system energy consumption while satisfying maximum tolerable delays for each offloaded task completion and block generation in the blockchain system.
3) To address the complexity of the resource allocation problem, we decompose it into two subproblems using dual decomposition. One subproblem involves resource allocation in MEC, while the other involves resource allocation in the blockchain system for block generation. The subproblem associated with MEC is further decomposed over ESs, and an analytical solution is derived to allocate computation frequency to offloaded tasks. The subproblem associated with block generation in the blockchain system is also analyzed.
4) Based on the analysis of derived subproblems, we propose two low-complexity optimization algorithms for resource allocation in blockchain-enabled MEC system. An intelligent assignment of MTs to ESs is performed and the allocated computation frequency to each MT is computed. Additionally, an ES that is responsible for block generation in the blockchain system and the amount of required computation frequency are determined accordingly.

Furthermore, the computational complexity of the proposed algorithms is analyzed and compared to the complexity of the optimal solution. Numerical simulations are conducted to evaluate the performance of the proposed method.

The remainder of this paper is structured as follows: Section II presents the system model encompassing communication, computation, and blockchain models. Section III delves into the proposed joint optimization approach, encompassing the problem formulation and analytic solution. In Section IV, two low-complex resource allocation algorithms are presented, along with their computational complexity analysis. Section V presents numerical results for performance comparison. Finally, Section VI draws conclusions.

II. SYSTEM MODEL

We consider a MEC system, as depicted in FIGURE 1. In this system, we have a set $\mathcal{N} \triangleq \{n : n = 1, \ldots, N\}$ of ESs situated at an equivalent set of base stations (BSs). In the first layer of the system, BSs cover a set $\mathcal{M} \triangleq \{m : m = 1, \ldots, M\}$ of MTs positioned randomly in a wireless network with single hop transmission. Each MT_m handles the offloading of a task to the system with a data size of D_m bits. Moreover, the required CPU cycles per bit by a task of MT_m or its computation density is denoted by C_m CPU cycles per bit.

TABLE 1. Model Parameters

Parameter	Description
$\mathcal{M}$	the set of MTs
$\mathcal{N}$	the set of ESs
D_m	task data size of MT_m in bits
C_m	computation density of MT_m in CPU cycles
$x_{m,n}$	the assignment of MT_m to ES_n binary variable
y_n	the selection of ES_n for block generation binary variable
$h_{m,n}$	channel gain from MT_m to BS_n
Ω_n	the set of MTs assigned to ES_n
p_T	transmission power of MTs
σ^2	noise power
B	channel bandwidth of BSs
$r_{m,n}$	transmit rate from MT_m to BS_n
t_m^{tr}	transmit delay of MT_m
t_m^{cmp}	computation delay of MT_m
E_m^{tr}	transmit energy consumption of MT_m
E_m^{cmp}	computation energy consumption for MT_m
$f_{m,n}$	computation frequency allocated from ES_n to MT_m
$\mathcal{F} = \{f_{m,n}\}$	the set of allocated computation frequencies
F_n	computation frequency budget of ES_n
I_b	blockchain block size in bits
C_b	blockchain computation density in CPU cycles
f_b	computation frequency allocated for block generation
K	effective capacitance of ESs CPU
t_b	delay of block generation
E_b	energy consumption of block generation
τ_m	maximum tolerable delay for MT_m
τ_b	maximum tolerable delay for block generation

Alongside the wireless network, in the second layer of the system, ESs are connected by a high-capacity wired communication infrastructure. This infrastructure serves as a blockchain network between ESs, where ESs participate as blockchain nodes. MTs transmit data of computation-intensive tasks to ESs via time-varying and fading channel from MTs to BSs. The computation resources to execute the tasks are provided by ESs, which are co-located with BSs. Finally, the outcome of this computation is sent back to MTs.

The defined parameters are summarized in Table 1, and the problem formulation is done in the following subsections.

A. COMMUNICATION MODEL

The communication channel between MTs and BSs is assumed to follow time-varying and flat-fading channels. Let's consider $h_{m,n}$ as the channel gain from MT_m to BS_n. The assignment of MT_m to BS_n is denoted by a binary variable $x_{m,n}$. If MT_m offload its own task to ES_n via communication with BS_n, then $x_{m,n} = 1$, otherwise $x_{m,n} = 0$. Since transmission for each MT is done only via one BS, we enforce the constraint $\sum_{n=1}^{N} x_{m,n} = 1$. Moreover, let's consider B as the total bandwidth available at each BS to be divided among assigned MTs. Considering $\Omega_n \triangleq \{m : x_{m,n} = 1\}$ as the set of assigned MTs to BS_n, transmit rate from each MT_m to BS_n is written as

$$r_{m,n} = \frac{B}{|\Omega_n|} \log_2 \left(1 + \frac{h_{m,n} p_T}{\sigma^2}\right) \tag{1}$$

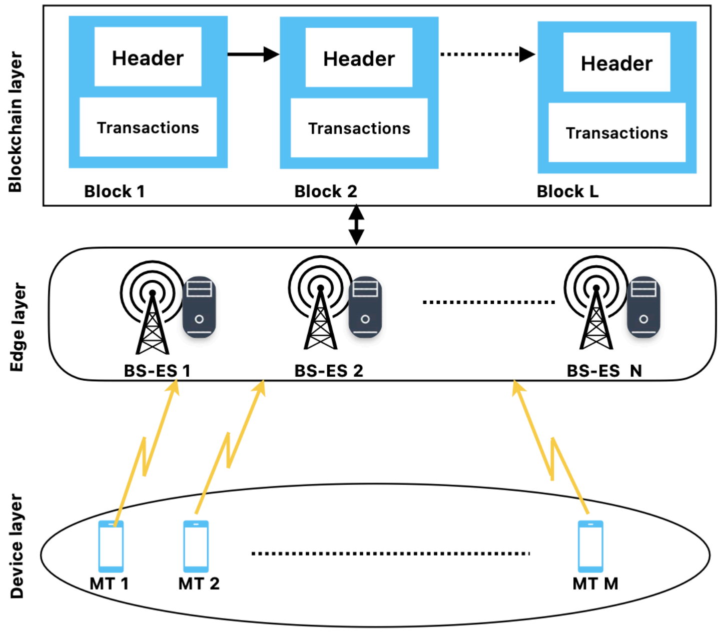

FIGURE 1. System Model

where $|\Omega_n|$ is the cardinality of Ω_n, p_T is a constant transmit power by MTs, and σ^2 is the noise power. If D_m is the task data size of MT_m to be transmitted to BS_n, the delay and consumption energy of this transmission are computed as

$$t_m^{tr} = \sum_{n=1}^{N} \frac{D_m}{r_{m,n}} x_{m,n} \tag{2}$$

and

$$E_m^{tr} = \sum_{n=1}^{N} p_T t_m^{tr} x_{m,n} \tag{3}$$

respectively.

B. COMPUTATION MODEL

Following the objective of task offloading in MEC, offloaded tasks to ESs are executed by leveraging their computation resources. Let's denote the total computation capacity at each ES_n as F_n in cycles per second. This capacity is allocated to offloaded tasks $m \in \Omega_n$ using predetermined rules to meet the offloading requirements. If C_m be the required CPU cycles per bit of MT_m, and $f_{m,n}$ be the allocated computation frequency to MT_m by ES_n, the delay and consumption energy of this computation are calculated as

$$t_m^{cmp} = \sum_{n=1}^{N} \frac{C_m D_m}{f_{m,n}} x_{m,n} \tag{4}$$

and

$$E_m^{cmp} = \sum_{n=1}^{N} K C_m D_m f_{m,n}^2 x_{m,n} = \sum_{n=1}^{N} \alpha_m f_{m,n}^2 x_{m,n} \tag{5}$$

respectively, where K is ES's CPU effective capacitance, and $\alpha_m \triangleq K C_m D_m$ is a constant value [21].

C. BLOCKCHAIN SYSTEM

As mentioned, in addition to the wireless network, there's a blockchain network that ensures the security and privacy of offloaded tasks. Since ESs have sufficient storage and computation resources, they're considered as blockchain nodes. These nodes participate in packaging task offloading data into transactions and then into blocks that are uploaded to the blockchain system. This process involves two steps: block generation and consensus. In the block generation step, transactions are collected and a corresponding block is generated. This step is computationally intensive and can cause delays in the blockchain system. After block generation, in the consensus process, the generated block is broadcast to all ESs, who verify it before saving it in the system.

In a blockchain system, we select one ES for block generation based on reputation-based voting [29]. We define a binary variable y_n to indicate whether ES_n is selected for block generation. If $y_n = 1$, ES_n is chosen, and if $y_n = 0$, it is not. Since only one node can generate a block, we enforce the constraint that $\sum_{n=1}^{N} y_n = 1$.

To calculate the delay and energy consumption associated with block generation, we define the block size in bits as I_b and the required CPU cycles per bit or computation intensity of block generation as C_b. If an ES allocates f_b CPU cycles per second for block generation, we can compute the time required to generate a block and the consumed energy as follows:

$$t_b = \sum_{n=1}^{N} \frac{C_b I_b}{f_b} y_n \tag{6}$$

and

$$E_b = \sum_{n=1}^{N} KC_b I_b f_b^2 y_n = \sum_{n=1}^{N} \alpha_b f_b^2 y_n. \tag{7}$$

respectively, where $\alpha_b \triangleq KC_b I_b$ is a constant value.

III. JOINT OPTIMIZATION

In this paper, we present a joint optimization problem to minimize the total energy consumption of MEC system encompassing the energy of communication, computation, and the blockchain system.

A. PROBLEM FORMULATION

The objective is to minimize the total energy as

$$E_T = \sum_{m=1}^{M} (E_m^{tr} + E_m^{cmp}) + E_b \tag{8}$$

where the first term is the energy consumed by MTs to transmit the tasks to BSs and the energy consumed by ESs to execute the tasks, and the second term is the energy of the blockchain system for block generation.

The minimization of energy is done while considering the constraints on the communication and computation resources, as well as the delay requirements of task offloading and block generation. Each MT_m is associated with a maximum tolerable delay, denoted as τ_m. In other words, the sum of communication and computation delays in (2) and (4) must not exceed a predetermined threshold τ_m, which is equivalent to $t_m^{tr} + t_m^{cmp} \leq \tau_m$. The time required for other operations such as signaling exchange and returning the results of task offloading are neglected due to their small values [2]. Moreover, the blockchain system is associated with a maximum tolerable delay, denoted as τ_b, for block generation, which is expressed as $t_b \leq \tau_b$.

Considering the introduced objective and constraints, the problem formulation is presented as follows:

$$(P_1): \min_{\mathbf{X},\mathbf{Y},\mathcal{F},f_b} \; E_T \tag{9a}$$

$$\text{s.t.} \quad t_m^{tr} + t_m^{cmp} \leq \tau_m \qquad \forall m \in \mathcal{M} \tag{9b}$$

$$t_b \leq \tau_b \tag{9c}$$

$$\sum_{m=1}^{M} x_{m,n} f_{m,n} + y_n f_b \leq F_n \qquad \forall n \in \mathcal{N} \tag{9d}$$

$$\sum_{n=1}^{N} x_{m,n} = 1 \qquad \forall m \in \mathcal{M} \tag{9e}$$

$$\sum_{n=1}^{N} y_n = 1 \tag{9f}$$

$$\mathbf{X}, \mathbf{Y} \in \{0,1\} \tag{9g}$$

where $\mathbf{X} = \{x_{m,n} : m \in \mathcal{M}, n \in \mathcal{N}\}$ and $\mathbf{Y} = \{y_n : n \in \mathcal{N}\}$ are the sets of binary variables, and $\mathcal{F} = \{f_{m,n} : m \in \mathcal{M}, n \in \mathcal{N}\}$ is the set of allocated computation frequencies by ESs to MTs. Constraints (9b) and (9c) express the maximum tolerable delays by MTs and the blockchain system, respectively. Constraint (9d) declares that the total computational frequencies allocated to MTs and the block generator by each ES_n must not exceed its computation frequency budget F_n. Each MT_m is constrained to be assigned to only one ES as in (9e), and only one ES is selected for block generation as in (9f).

Problem (9) is nonlinear programming with both binary and continuous optimization variables. Therefore, it is a mixed integer nonlinear programming (MINLP), which is NP-hard with exponential computational complexity and no analytical solution in general.

To overcome the complexity of problem (9), we relax the coupling constraints between MTs in (9d) and form a partial Lagrangian function as

$$L(\mathbf{X},\mathbf{Y},\mathcal{F},f_b,\Lambda) \triangleq E_T + \sum_{n=1}^{N} \lambda_n \left(\sum_{m=1}^{M} x_{m,n} f_{m,n} + y_n f_n - F_n \right) \tag{10}$$

where $\Lambda = \{\lambda_n \geq 0 : n \in \mathcal{N})$ is the set of Lagrange multipliers associated with N constraints in (9d). Substituting E_T with its equivalent terms in (8), this function is rewritten as

$$L(\mathbf{X},\mathbf{Y},\mathcal{F},f_b,\Lambda) = \sum_{n=1}^{N} \left(\sum_{m=1}^{M} \left(p_T t_m^{tr} + \alpha_m f_{m,n}^2 \right) x_{m,n} + \alpha_b f_b^2 y_n \right) + \sum_{n=1}^{N} \lambda_n \left(\sum_{m=1}^{M} x_{m,n} f_{m,n} + y_n f_b - F_n \right) \tag{11}$$

which is equivalent to

$$L(\mathbf{X},\mathbf{Y},\mathcal{F},f_b,\Lambda) = = \sum_{n=1}^{N} \sum_{m=1}^{M} \left(p_T t_m^{tr} + \alpha_m f_{m,n}^2 + \lambda_n f_{m,n} \right) x_{m,n} + \sum_{n=1}^{N} \left(\alpha_b f_b^2 + \lambda_n f_b \right) y_n - \sum_{n=1}^{N} \lambda_n F_n \tag{12}$$

Optimizing the Lagrangian function with respect to the primal variables $(\mathbf{X},\mathbf{Y},\mathcal{F},f_b)$ yields the dual function

$$D(\Lambda) \triangleq \inf_{\mathbf{X},\mathbf{Y},\mathcal{F},f_b} L(\mathbf{X},\mathbf{Y},\mathcal{F},f_b,\Lambda) \tag{13}$$

$$\text{s.t.} \quad (9b), (9c), (9e), (9f), (9g) \tag{14}$$

which provides a lower bound on the optimal solution of (9) for every feasible variable Λ [30]. The tightest lower bound is obtained by the dual problem

$$\max_{\Lambda \geq 0} \; D(\Lambda). \tag{15}$$

Prior to solving the problem in the dual domain, we need to evaluate $D(\Lambda)$ in (13) for a given $\Lambda \geq 0$. Thanks to the decomposable form of $L(\mathbf{X},\mathbf{Y},\mathcal{F},f_b,\Lambda)$ in (11), we can leverage dual decomposition to decouple (13) into subproblems across ESs as

$$
\begin{aligned}
(P_2): \min_{\mathbf{X},\mathcal{F}} \quad & \sum_{n=1}^{N}\sum_{m=1}^{M}\left(p_T t_m^{tr} + \alpha_m f_{m,n}^2 + \lambda_n f_{m,n}\right) x_{m,n} \\
\text{s.t.} \quad & t_m^{tr} + t_m^{cmp} \le \tau_m, \quad \forall m \in \mathcal{M} \\
& \sum_{n=1}^{N} x_{m,n} = 1, \quad \forall m \in \mathcal{M} \\
& \mathbf{X} \in \{0,1\}
\end{aligned}
$$

and

$$
\begin{aligned}
(P_3): \min_{\mathbf{Y},f_b} \quad & \sum_{n=1}^{N}\left(\alpha_b f_b^2 + \lambda_n f_b\right) y_n \\
\text{s.t.} \quad & t_b \le \tau_b \\
& \sum_{n=1}^{N} y_n = 1 \\
& \mathbf{Y} \in \{0,1\}.
\end{aligned}
$$

As seen, leveraging dual decomposition, problem (9) has been decoupled into P_2 and P_3, where both are MINLP. These problems are to be solved in the following subsections.

B. PROBLEM P_2

To solve P_2, for now, let's assume that binary variables in $\mathbf{X}$ are known. In other words, we assume that the set of MTs assigned to each BS_n is known and denoted by Ω_n. Consequently, P_2 can be decomposed into N subproblems across ESs, each of which is written as follows:

$$
\min_{\mathcal{F}_n} \sum_{m\in\Omega_n} p_T t_m^{tr} + \alpha_m f_{m,n}^2 + \lambda_n f_{m,n} \tag{18a}
$$

$$
\text{s.t.} \quad t_m^{tr} + t_m^{cmp} \le \tau_m, \quad \forall m \in \mathcal{M} \in \Omega_n \tag{18b}
$$

This problem can also be decoupled into subproblems over MTs in Ω_n , each of which is expressed as

$$
\min_{f_{m,n}} \quad p_T t_m^{tr} + \alpha_m f_{m,n}^2 + \lambda_n f_{m,n} \tag{19a}
$$

$$
\text{s.t.} \quad t_m^{tr} + t_m^{cmp} \le \tau_m. \tag{19b}
$$

The objective function in (19a) is a quadratic function of $f_{m,n}$, where $p_T t_m^{tr}$ is a constant value, and coefficients α_m and λ_n are positive. Consequently, the optimal value of $f_{m,n}$ is only constrained by the delay requirement in (19b). Given the known values of $x_{m,n}$'s, we can derive the value of t_m^{tr} in (2) and then derive $t_m^{cmp} \le \tau_m - t_m^{tr}$. Substituting t_m^{cmp} with its equivalent term $\frac{C_m D_m}{f_{m,n}}$ in (4), the optimal value of computing frequency from ES_n to MT_m is derived as

$$
f_{m,n}^* = \frac{C_m D_m}{\tau_m - t_m^{tr}} \tag{20}
$$

if $x_{m,n} = 1$, otherwise $f_{m,n}^* = 0$. This derivation is leveraged to design resource allocation algorithms in Section IV.

Algorithm 1 Low-complex blockchain-enabled resource allocation in MEC (LC1-BE-MEC)

Initialization: $\mathbf{X} = \mathbf{0}$, $\mathbf{Y} = \mathbf{0}$, $\mathcal{F} = \mathbf{0}$, $\Omega_n = \emptyset$ for all $n \in \mathcal{N}$.
for all $m \in \mathcal{M}$ **do**
  $n_m \leftarrow \arg\max_n h_{m,n}$
  $\Omega_{n_m} = \Omega_{n_m} \cup m$
end for
for all $n \in \mathcal{N}$ **do**
  if $\Omega_n \neq \emptyset$ **then**
    compute $f_{m,n}$ as in (20) for all $m \in \Omega_n$.
    if any $f_{m,n} < 0$ or $\sum_{m\in\Omega_n} f_{m,n} > F_n$ **then**
      declare the solution as infeasible.
    end if
  end if
end for
compute f_b^* as in (21).
find an ES_n with $F_n \ge f_b^*$.
set $y_n = 1$.

C. PROBLEM P_3

The optimization variable f_b in P_3 is provided only by one ES, as constrained by $\sum_{n=1}^{N} y_n = 1$. Moreover, the coefficients α_b and λ_n in the objective function are both positive. Consequently, the objective function is non-decreasing in terms of f_b, implying that the optimal value of f_b is only limited by the frequency requirement to meet constraint $t_b \le \tau_b$. To fulfill this constraint, following the derivation of f_b in (6), we calculate the optimal value of f_b as

$$
f_b^* = \frac{C_b I_b}{\tau_b}. \tag{21}
$$

This computation frequency must be provided by an ES. To achieve this, let's denote the residual computation frequency of each ES_n as $\hat{F}_n = F_n - \sum_{m=1}^{M} x_{m,n} f_{m,n}^*$ after the resource allocation has been done in problem P_2. Subsequently, f_b^* can be provided by any ES with $\hat{F}_n \ge f_b$. Consequently, the corresponding y_n value to that ES is set to 1, while all other y_n values are set to 0.

IV. PROPOSED ALGORITHMS

In this section, we consider the derived solutions for problems P_2 and P_3 in (20) and (21). Consequently, we propose two low-complex blockchain-enabled resource allocation schemes for MEC, named LC1-BE-MEC and LC2-BE-MEC, respectively.

As highlighted in the solution of problem P_2, under the assumption of fixed binary variables in $\mathbf{X}$, which determines the assignment of MTs to ESs, the optimal values of computation frequency allocation in in (20) can be computed. To exploit this derivation, in Algorithms 1 and 2, we first consider a greedy assignment of MTs to ESs. This means that each MT is preferentially assigned to an ES with the maximum channel gain, given the available computation frequencies. The rationale behind this idea is that a higher channel gain leads to a

Algorithm 2 Low-complex blockchain-enabled resource allocation in MEC (LC2-BE-MEC)

```
1: Initialization: $\hat{\mathcal{M}} \leftarrow \mathcal{M}$, $\hat{\mathcal{N}} \leftarrow \mathcal{N}$, $\mathbf{X} = \mathbf{0}$, $\mathbf{Y} = \mathbf{0}$.
2: while $\hat{\mathcal{M}} \neq \emptyset$ do
3:     select $n \in \hat{\mathcal{N}}$ randomly.
4:     for all $m \in \hat{\mathcal{M}}$ do
5:         $n_m^* \leftarrow \arg\max_n h_{m,n}$
6:         if $n_m^* = n$ then
7:             $\Omega_n = \Omega_n \cup m$
8:         end if
9:     end for
10:    if $\Omega_n \neq \emptyset$ then
11:        compute $f_{m,n}$ as in (20) for all $m \in \Omega_n$.
12:        sort $f_{m,n}$ values in the ascending order in a vector as $F_{sort}$ and indexes in $I$.
13:        remove values from $F_{sort}$ and indexes from $I$ with $f_{m,n} \leq 0$.
14:        if $F_{sort} \neq \emptyset$ then
15:            for $m \in I$ do
16:                if $F_n - F_{sort}(m) \geq 0$ then
17:                    $F_n = F_n - F_{sort}(m)$.
18:                    $x_{m,n} = 1$.
19:                    $\hat{\mathcal{M}} = \hat{\mathcal{M}} \setminus \{m\}$.
20:                end if
21:            end for
22:        else
23:            $\hat{\mathcal{N}} = \hat{\mathcal{N}} \setminus \{n\}$
24:        end if
25:    end if
26:    if $\hat{\mathcal{M}} \neq \emptyset$ and $\hat{\mathcal{N}} = \emptyset$ then
27:        the solution is infeasible.
28:    end if
29: end while
30: compute $f_b^*$ as in (21).
31: find an ES$_n$ with $F_n \geq f_b^*$.
32: set $y_n = 1$.
```

higher transmission rate in (1), which subsequently results in a lower transmission delay in (2). As in (20), lower values for t_m^{tr} result in lower frequency values $f_{m,n}$. This computation ultimately leads to reduced communication and consumption energies at both MTs and ESs.

A. ALGORITHM 1

The first algorithm, named as LC1-BE-MEC, is presented in Algorithm 1. The set of MTs assigned to each ES$_n$ is maintained in a set Ω_n. After the greedy assignment of MTs to BSs (lines 2-5), the derived frequencies (line 8) are check against infeasible solutions (lines 9). In other words, the solution is declared infeasible if any $f_{m,n} < 0$ or $\sum_{m\in\Omega_n} f_{m,n} > F_n$ for any ES. A negative $f_{m,n}$ value in (20) indicates that the transmission delay exceeds the maximum tolerable delay in in (19b). Finally, an ES with sufficient computation resources exceeding f_b^* is selected for block generation.

B. ALGORITHM 2

The computation in (20) faces a challenge due to the possibility of infeasible solutions when the aggregate allocated frequency by an ES exceeds the computation frequency budget F_n. This violates the constraint (9d). To address this issue and enhance resource allocation, we introduce LC2-BE-MEC in Algorithm 2, where we always check out avoiding infeasible solutions as far as possible whenever each MT is assigned to an ES.

In LC2-BE-MEC, we initialize $\hat{\mathcal{M}}$ and $\hat{\mathcal{N}}$ as temporary sets of MTs and ESs. In a while loop, we select ESs randomly. For each selected ES$_n$, we identify MTs with the maximum channel gain among all ESs and include them in a set Ω_n (lines 4-9). These MTs are potential candidates for assignment to ES$_n$. If $\Omega_n \neq \emptyset$, we compute the allocated computation frequencies $f_{m,n}$ in (20) under the assumption that MTs in Ω_n are assigned to ES$_n$ (line 11).

Computed frequencies $f_{m,n}$ for $m \in \Omega_n$ are sorted in ascending order in the vector F_{sort}. Corresponding indexes are saved in the vector I. If $F_{sort} \neq \emptyset$, each MT$_m$ in Ω_n is definitely assigned to ES$_n$ provided there are sufficient computational resources, i.e., $F_n - F_{sort}(m) \geq 0$. If this condition is met, we reduce the frequency budget F_n by $F_{sort}(m)$, set $x_{m,n} = 1$, and proceed to the next iteration and exclude MT$_m$ from $\hat{\mathcal{M}}$ (lines 15-21). Recall that $\hat{\mathcal{M}}$ has been defined as the set of unassigned MTs. $F_{sort} = \emptyset$ for an ES$_n$ means that there aren't enough available frequency resources to meet the frequency demand of any MT. Consequently, this ES is removed from the set of available ESs in $\hat{\mathcal{N}}$ (line 23).

The while loop continues until all MTs are assigned to ESs. If there's no ES in $\hat{\mathcal{N}}$ for a given MT in $\hat{\mathcal{M}}$, the problem is declared infeasible (line 27). Finally, the computation frequency requirement for block generation is calculated, and an ES with available computation frequency is selected, as explained in the solution to P_3 (lines 30-32).

C. COMPLEXITY ANALYSIS

As mentioned earlier, the formulated optimization problem in (9) is a MINLP with exponential complexity. Notably, it includes both binary variables in $\mathbf{X}$ and $\mathbf{Y}$ and continuous variables in $\mathcal{F}$ and f_b. An exhaustive search to explore the optimal values of $\mathbf{X}$ and $\mathbf{Y}$ requires $\mathcal{O}(NN^M)$ computational complexity, where M MTs are assigned to N ESs, and the block generator node is selected from among N ESs. Moreover, the joint optimization of binary and continuous variables can further increase the computational complexity of problem (9).

Algorithm 1 includes a search operation among N ESs for each MT, followed by the computation of frequency for all MTs. The worst-case complexity of these operations can be estimated as $\mathcal{O}(2MN)$.

To address the computational complexity of Algorithm 2, we derive the worst-case complexity in terms of the number of iterations of the **while** loop. This loop searches for the ES with the maximum channel gain for each MT in the first **for** loop, computes, sorts, and checks out $f_{m,n}$ values, and then executes

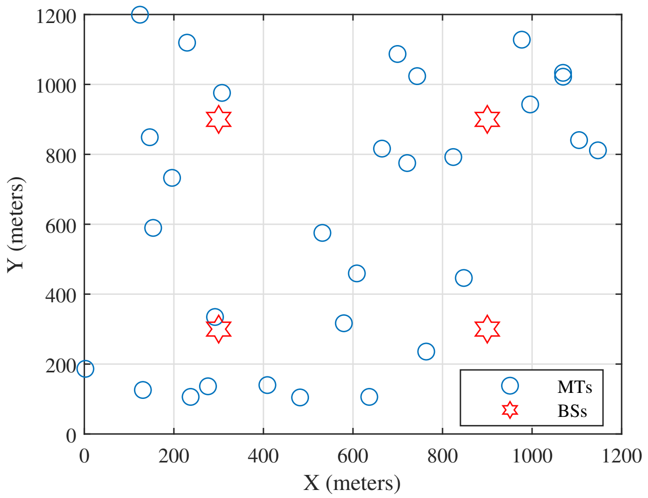

FIGURE 2. Network topology with random locations of MTs

the last **for** loop to assign MTs to ESs. Considering the worst-case complexity of these terms, the overall algorithm complexity is estimated as $\mathcal{O}(N(MN + 5M))$.

As observed, unlike the exponential complexity of the optimal problem solution (9), the proposed Algorithms have a polynomial complexity in terms of the number of MTs and ESs.

V. PERFORMANCE EVALUATION

Numerical results are conducted to evaluate the performance of proposed LC1-BE-MEC and LC2-BE-MEC algorithms. Extensive experiments are applied to a network consisting of four BSs with locations typically indicated in FIGURE 2. Associated with each BS, there is a co-located ES to provide computational frequency for task offloading. Several MTs are randomly positioned within the network, where each MT is associated with a task to be offloaded.

The communication channel between MTs and BSs is assumed to follow a Rayleigh flat fading channel. The received power at a BS is represented by an exponential probability density function with a mean of $p_T - 30 - 30\log_{10} d + \psi$, where p_T is the transmit power of a MT, d is the distance, and ψ is the shadowing effect modeled by a normal distribution with a mean of 3 dB. The values of communication and network parameters during the simulation are summarized in Table 2.

We also consider the following schemes for comparison in parallel with the proposed LC1-BE-MEC and LC2-BE-MEC algorithms.

1) One common approach for computation frequency allocation is to divide the computation frequency resources at each ES by the number of assigned MTs. To investigate the impact of uniform allocation of computation frequency, we consider an algorithm that performs the greedy assignment of MTs as in Algorithm 1, and then divides the computation frequency resources at each ES by the number of assigned MTs. We label this approach as *greedy+uniform frequency* in the simulation results.
2) As stated in Section I-A, deep RL is a well known technique for task offloading as in [16], [17], and [18]. This technique is adopted to solve our task offloading problem by considering binary decision variables in $\mathbf{X}$ and $\mathbf{Y}$ as the action space in the RL algorithm. For a given value of actions in deep RL, computation frequencies as continuous variables are derived optimally as in (20) and (21). We also consider this approach for comparison with other algorithms and label it as *DRL+optimal frequency* in the simulation results.

TABLE 2. Simulation parameters

Parameter	notation	value
MTs transmit power	p_T	0 dBm
Noise power	σ^2	-100 dB
BS channel bandwidth	B	100 MHz
Task data size of MTs	D_m	1 Mb
MTs computation density	C_m	1 cycle
Computation frequency budget	F_n	5 GHz
Blockchain block size	I_b	1 Mb
Blockchain computation density	C_b	10 cycles
Effective capacitance of ESs CPU	K	10^{-26}
maximum tolerable delay for MTs	τ_m	200 ms
maximum tolerable delay for block generation	τ_b	100 ms

Performance evaluation is done in terms of communication energy $E_{com} = \sum_{m=1}^{M} E_m^{tr}$ as the sum of MTs transmission energies, computation energy $E_{cmp} = \sum_{m=1}^{M} E_m^{cmp}$ as the sum of MTs computation energies, and total consumption energy $E_T = E_{com} + E_{cmp} + E_b$. Moreover, to measure the performance in terms of the number of infeasible solutions during the simulation, we introduce a measure named as outage probability P_{out}, which quantifies the likelihood of an infeasible solution occurring at each numerical experiment, or equivalently

$$P_{out} = \frac{n_f}{T} \tag{22}$$

where n_f is the number of iterations in which an infeasible solution has been declared among T iterations. T as the total number of iterations is set to 1000. Additionally, we define another performance measure called normalized E_T, which represents the product of E_T and P_{out}. This metric facilitates logical comparisons of energy consumption when algorithms exhibit varying outage probabilities or energy consumption levels.

The variation of considered metrics is first evaluated versus the number of MTs in the network with a given communication and computation resources, as shown in Table 2. The communication energy is depicted in FIGURE 3 versus the variation in the number of MTs. It is evident that all algorithms experience an increase in energy as the number of MTs increases. Notably, the algorithms mostly demonstrate the same performance since they consider the status of channel gains for the assignment of MTs to BSs. The computation energy is also shown in FIGURE 4. This energy is significantly high when the computation frequency at each

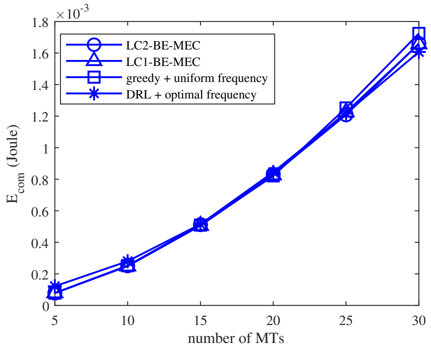

FIGURE 3. **Communication energy vs. number of MTs**

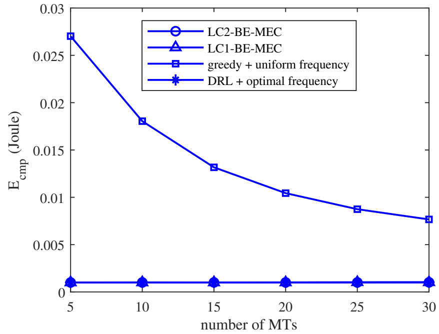

FIGURE 4. **Computation energy vs. number of MTs**

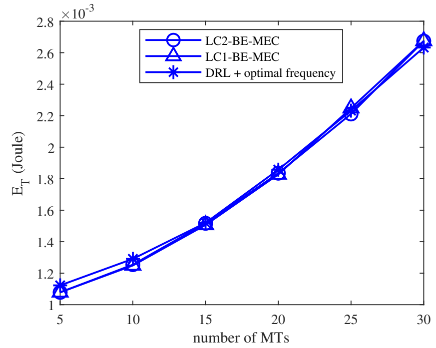

FIGURE 5. **Total energy vs. number of MTs**

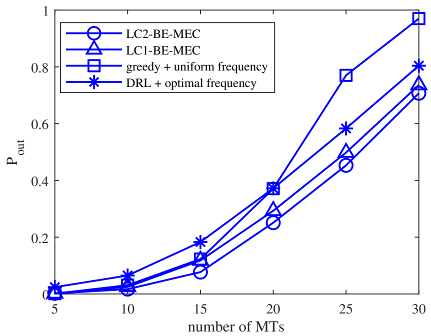

FIGURE 6. **Outage probability vs. number of MTs**

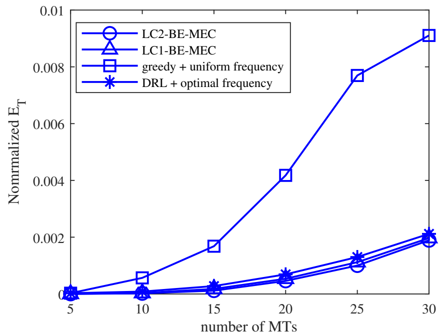

FIGURE 7. **Normalized total energy vs. number of MTs**

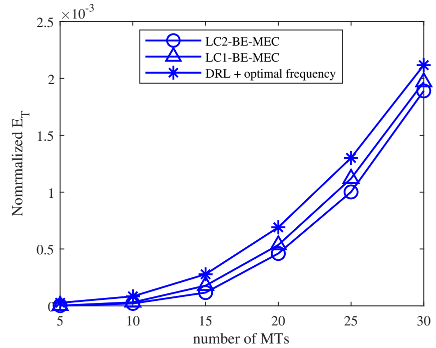

FIGURE 8. **Normalized total energy vs. number of MTs**

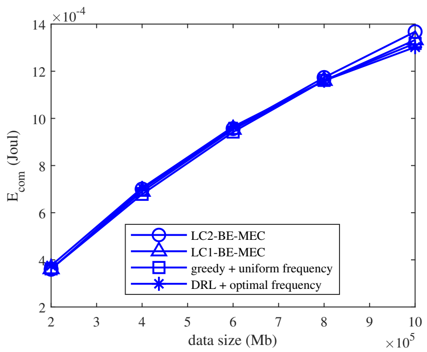

FIGURE 9. **Communication energy vs. task data size**

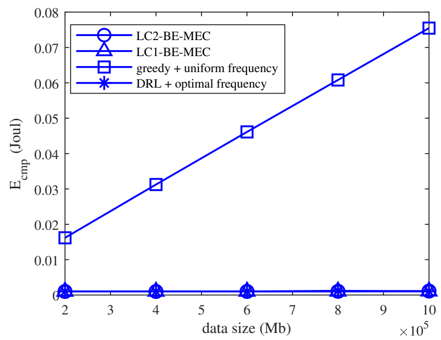

FIGURE 10. **Computation energy vs. task data size**

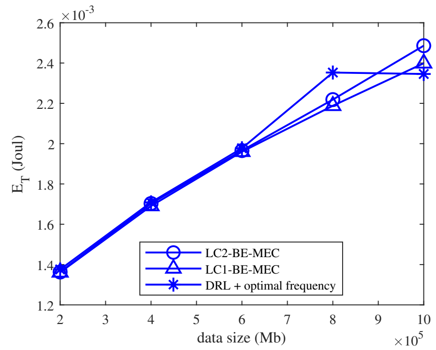

FIGURE 11. **Total energy vs. task data size**

ES is uniformly allocated to assigned MTs. This highlights the efficiency of derived optimal computation frequency allocation in (20) compared to uniform frequency allocation. The decrease in computation energy with an increase in the number of MTs for uniform frequency allocation is a results of the quadratic nature of this energy in (5), where smaller divisions of frequency lead to lower energy.

In FIGURE 5, the total consumed energy E_T is plotted against the number of MTs. To highlight a detailed comparison between the other algorithms, the uniform frequency allocation E_T is eliminated. The other three algorithms allocate the computation frequency in a similar manner, resulting in computation and total energies that are close to each other.

Outage probability comparison is also illustrated in Figure 6. As observed, the probability of infeasible solutions increases with the increase in the number of MTs. In other words, the communication and computation resources in the network no longer suffice to meet the maximum tolerable delays specified in problem (9). Our proposed algorithms undoubtedly offer lower levels of outage probabilities, especially when the number of MTs in the network increases. This improvement is the result of the modified assignment of MTs to ESs and computation frequency allocation. Especially, from line 15 to 21 in the LC2-BE-MEC algorithm, if there is not sufficient computation frequency for a MT at an ES, this MT will be assigned to another ES in the next round of the

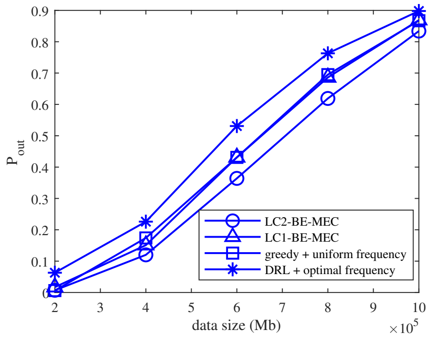

FIGURE 12. Outage probability vs. task data size

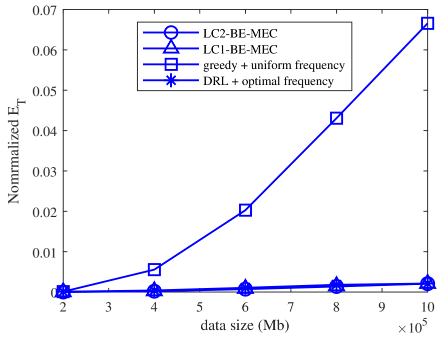

FIGURE 13. Normalized energy vs. task data size

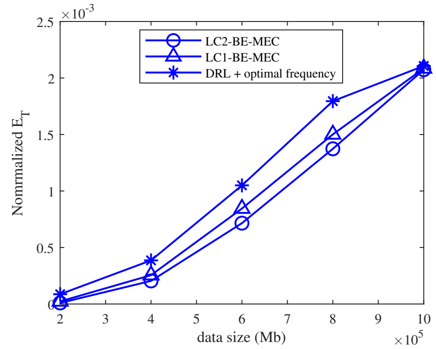

FIGURE 14. Normalized energy vs. task data size

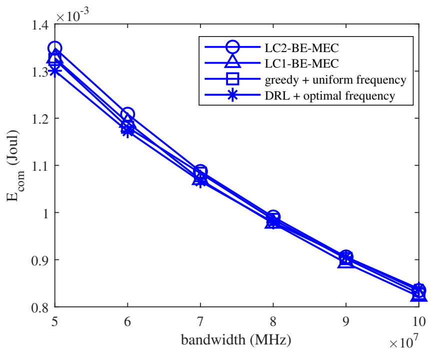

FIGURE 15. Communication energy vs. bandwidth

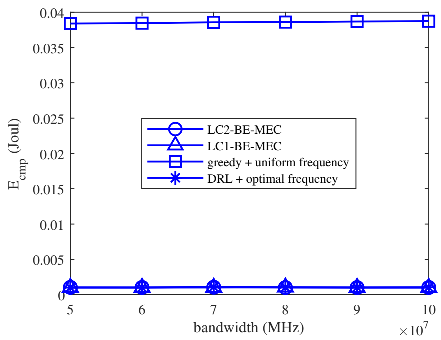

FIGURE 16. Computation energy vs. bandwidth

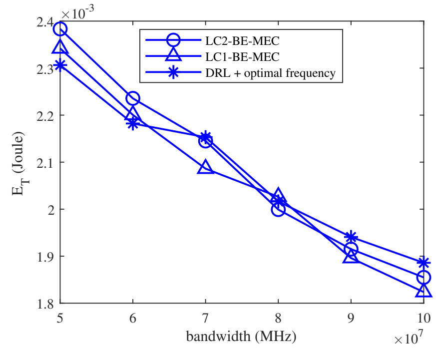

FIGURE 17. Total energy vs. bandwidth

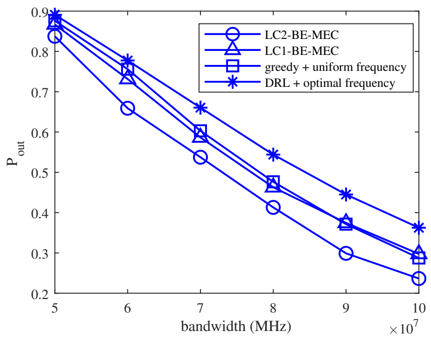

FIGURE 18. Outage probability vs. bandwidth

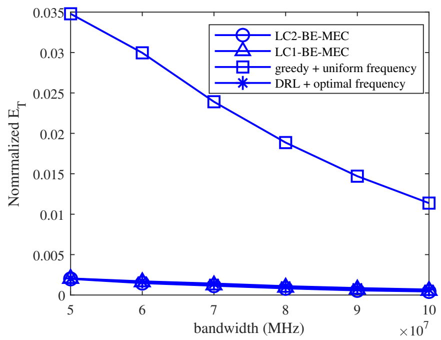

FIGURE 19. Normalized energy vs. bandwidth

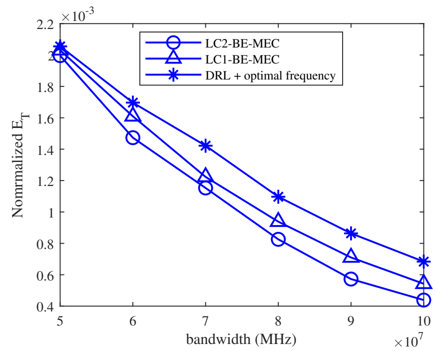

FIGURE 20. Normalized energy vs. bandwidth

algorithm. This amendment results in more feasible solutions.

The normalized E_T performance is illustrated in Figure 7. This performance is also depicted in Figure 8, which eliminates the uniform frequency allocation for a more comprehensive comparison between the other algorithms. As evident, the proposed LC1-BE-MEC and LC2-BE-MEC algorithms exhibit enhanced performance metrics compared to DRL and uniform frequency allocation.

In the following, we investigate the variation of performance metrics versus the variation in the data size of offloaded tasks, i.e., D_m. The number of MTs remains constant at $M = 20$ and other parameters are considered as shown in Table 2. Communication and computation energies are depicted in FIGURE 9 and FIGURE 10, respectively. Communication energy increases with the task data size, and once again, the consumption energy is substantially high for the uniform frequency allocation, similar to variations in the number of MTs. The total consumed energy is also shown in FIGURE 11, eliminating the energy of uniform frequency allocation for a detailed comparison between the other algorithms. Except for the uniform frequency allocation, which has a significantly high energy consumption, other algorithms generally have the same total consumed energy.

Outage probability versus increase in tasks data size is shown in FIGURE 12. As the data size increases, it becomes more demanding in terms of communication and computational resources. Consequently, the likelihood of encountering infeasible solutions rises with larger data sizes. Similar to the increase in the number of MTs, the proposed LC2-BE-MEC algorithm demonstrates a reduction in outage probabilities. This improvement is undoubtedly a consequence of the modifications made to Algorithm 2, which effectively prevents the generation of infeasible solutions.

Finally, the normalized E_T metric is presented in FIGURE 13 and repeated in FIGURE 14, eliminating uniform frequency allocation to facilitate a more detailed comparison between the other algorithms. This metric also demonstrates the superior performance of the proposed LC1-BE-MEC and LC2-BE-MEC algorithms.

The next step in performance evaluation involves comparing the performance of different algorithms against the variation in communication bandwidth, i.e., B. The communication and computation energies are shown in FIGURE 15 and FIGURE 16, respectively. The channel bandwidth between MTs and BSs directly influences the transmission delay and, consequently, the communication energy. A wider bandwidth results in reduced transmission delay and lower communication energy for all algorithms, as evident in FIGURE 15. However, the computation energy is independent on the channel bandwidth and thus it remains constant across all algorithms in FIGURE 16. Considering these observations, the total energy consumption decreases with increasing bandwidth, as depicted in FIGURE 17. Additionally, the variation of outage probability is shown in FIGURE 18. This metric also decreases with an increases in the communication bandwidth. Notably, the lower values of outage probability for the LC2-BE-MEC algorithm are observed in FIGURE 18.

Finally, the normalized E_T performance is also depicted in FIGURE 19 and is repeated in FIGURE 20, eliminating the uniform frequency allocation. While the uniform frequency allocation demonstrates high normalized E_T values, the proposed LC2-BE-MEC algorithm achieves lower values.

VI. CONCLUSION

This paper investigated the optimization of joint resource allocation in blockchain-enabled MEC. Analytical and numerical results validated the efficiency of a greedy approach for assigning MTs to BSs. Each MT is prioritized for assignment to a BS with the highest channel gain. Additionally, computation frequency is allocated to offloaded tasks based on optimal solutions derived from this approach. The low-complex proposed algorithms based on this approach exhibited lower outage probabilities in comparison with other methods such DRL, especially when communication and computation resources become scarce or the number of MTs increases. Furthermore, lower normalized total energy consumption in the proposed algorithms demonstrated the effectiveness of this approach.

REFERENCES

[1] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency iot services in multi-access edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 668–682, 2019.

[2] M. Fathi, M. Saroughi, and A. Zareie, "Delay-sensitive task offloading optimization by geometric programming," *IEEE Transactions on Cloud Computing*, vol. 12, no. 3, pp. 889–896, 2024.

[3] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE internet of things journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[4] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.

[5] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[6] J. Xu, S. Wang, B. K. Bhargava, and F. Yang, "A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3538–3547, 2019.

[7] J. A. Alzubi, O. A. Alzubi, A. Singh, and T. Mahmod Alzubi, "A blockchain-enabled security management framework for mobile edge computing," *International Journal of Network Management*, vol. 33, no. 5, p. e2240, 2023.

[8] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.

[9] Z. Khan, A. G. Abbasi, and Z. Pervez, "Blockchain and edge computing–based architecture for participatory smart city applications," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 12, p. e5566, 2020.

[10] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2016.

[11] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2536–2549, 2020.

[12] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2019.

[13] J. Shi, J. Du, Y. Shen, J. Wang, J. Yuan, and Z. Han, "Drl-based v2v computation offloading for blockchain-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, 2022.

[14] P. Lin, Q. Song, F. R. Yu, D. Wang, and L. Guo, "Task offloading for wireless vr-enabled medical treatment with blockchain security using collective reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 749–15 761, 2021.

[15] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4051–4063, 2020.

[16] A. Samy, I. A. Elgendy, H. Yu, W. Zhang, and H. Zhang, "Secure task offloading in blockchain-enabled mobile edge computing with deep reinforcement learning," *IEEE Transactions on network and service management*, vol. 19, no. 4, pp. 4872–4887, 2022.

[17] K. Moghaddasi, S. Rajabi, and F. S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled iov-mec systems: a double deep q-network approach," *IEEE Access*, vol. 12, pp. 3437–3463, 2024.

[18] Y. Liu, X. Guan, Y. Peng, H. Chen, T. Ohtsuki, and Z. Han, "Blockchain-based task offloading for edge computing on low-quality data via distributed learning in the internet of energy," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 657–676, 2021.

[19] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5549–5561, 2020.

[20] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, 2021.

[21] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4321–4334, 2020.

[22] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 695–708, 2018.

[23] Y. Zuo, S. Jin, S. Zhang, Y. Han, and K.-K. Wong, "Delay-limited computation offloading for mec-assisted mobile blockchain networks," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8569–8584, 2021.

[24] J. Du, Z. Yu, A. Sun, J. Jiang, H. Zhao, N. Zhang, C. Wu, and F. R. Yu, "Secure task offloading in blockchain-enabled mec networks with improved pbft consensus," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[25] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "Become: Blockchain-enabled computation offloading for iot in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2019.

[26] H. Wu, K. Wolter, P. Jiao, Y. Deng, Y. Zhao, and M. Xu, "Eedto: An energy-efficient dynamic task offloading algorithm for blockchain-enabled iot-edge-cloud orchestrated computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2163–2176, 2020.

[27] S. Luo, H. Li, Z. Wen, B. Qian, G. Morgan, A. Longo, O. Rana, and R. Ranjan, "Blockchain-based task offloading in drone-aided mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 124–129, 2021.

[28] M. Liu, F. R. Yu, Y. Teng, V. C. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 008–11 021, 2018.

[29] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[30] M. Fathi and H. Bevrani, *Optimization in electrical engineering*. Springer, 2019.

MOEIN VALITABAR is a Master's student in Communication Systems at the University of Kurdistan, graduating in 2025. His research focuses on optimization techniques and distributed learning for wireless communication systems. Prior to pursuing his Master's degree, he earned a Bachelor's degree in Electrical Engineering from Razi University.

MOHAMMAD FATHI received his M.Sc. and Ph.D. in electrical engineering from Amirkabir University of Technology, Tehran, Iran in 2003 and 2010, respectively. He works as an Associate Professor with the Department of Electrical Engineering, University of Kurdistan, Sanandaj, Iran. His current research interests include machine learning, resource allocation optimization in wireless networks, cloud and edge computing, and network security.

KEIVAN NAVAIE is a Professor in Intelligent Networks at Lancaster University's School of Computing and Communications. Additionally, he is also a member of the Independent Scientific Advisory Committee at the prestigious Alan Turing Institute. He is a Fellow of the Institution of Engineering and Technology (IET), a Chartered Engineer in the UK, and a Senior Fellow of the Higher Education Academy (HEA).