

## Research Article

# User Energy Efficiency Fairness Algorithms for Task Offload in Cloud Edge Networks

Junjie Li <sup>1,2</sup>, Xiaolong Yang <sup>2,3</sup> and Yuexia Zhang <sup>1,3</sup>

<sup>1</sup>Key Laboratory of Information and Communication Systems, Ministry of Information Industry, Beijing Information Science and Technology University, Beijing 100101, China

<sup>2</sup>School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China

<sup>3</sup>Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing Information Science and Technology University, Beijing 100101, China

Correspondence should be addressed to Yuexia Zhang; [zhangyuexia@bistu.edu.cn](mailto:zhangyuexia@bistu.edu.cn)

Received 28 May 2022; Revised 30 June 2022; Accepted 25 July 2022; Published 17 August 2022

Academic Editor: Le Sun

Copyright © 2022 Junjie Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the emergence of several new services such as driverless vehicles and virtual reality, mobile communication networks face problems such as heavy load and insufficient computing resources. The development of cloud, edge, and mobile edge network computing provides a good solution to this problem. This paper proposes the development of a user energy efficiency fairness task unloading algorithm for cloud-side networks. First, a cloud-side network cooperation model is constructed. The model ensures the efficient use of user energy and addresses the task offloading decision and resource allocation optimization problem jointly. Using the generalized fraction theory, the optimization problem is transformed into an equivalent convex problem by introducing relaxation as well as auxiliary variables. Next, the centralized energy efficiency fairness (CEEF) and alternating direction method of multiplier (ADMM)-based energy efficiency fairness algorithms are implemented to obtain an optimal solution for the optimization problem. Finally, through experimental simulation, the convergence of the CEEF- and ADMM-based energy efficiency fairness algorithms is verified. Compared with noncooperative algorithms, the performance of our proposed method increased by 30.76%. The proposed algorithm has been verified to ensure the fairness of user energy efficiency.

## 1. Introduction

With the rapid advances in mobile communication, several new applications such as virtual/augmented reality, face recognition, and automated driving have been developed. These applications demand ultra-low delay and high reliability of mobile networks [1–3]. Although these new applications enhance the convenience of performing tasks, the amount of data and energy consumed to transmit data has increased at an astronomical rate. This has led to a heavy processing burden on the network communication equipment. Cloud-side networks can help realize comprehensive applications regarding communication computing and resource collaboration to efficiently mitigate the burden on the communication equipment [4–10]. Therefore, research on cloud-side networks is of great

significance. An incredible amount of research has been conducted in recent years.

The combination of cloud-side networks and mobile edge computing (MEC) [11–15] technology is able to fulfill the networking requirements of communication, computing, and business processing. It can realize an effective support system for the network providing novel applications. By applying MEC technology, users on the cloud-side network can upload computing tasks with high computing requirements and energy consumption to edge nodes. Edge computing nodes are very close to users. Therefore, data transmission pressure and link congestion can be effectively alleviated. Simultaneously, the network bandwidth demand and transmission delay in data computing and storage process can be significantly reduced [16–18].

At present, considerable advancements have been made in the field of mobile edge computing for cloud network-edge network applications. Previous research used edge nodes and mobile devices to form an edge computing system wherein mobile devices have computing tasks [19]. Considering the frequency adjustment and nonadjustment of edge access points, different methods were used to obtain the unloading decision. A user cooperative cloud network-edge mobile edge computing model was also proposed, whose primary objective is minimizing the task processing delay to make the optimal task offloading decision [20]. Stochastic optimization theory was also used to make the optimal unloading decision for the computational unloading problem [21]. The data cache in the mobile edge computing model was considered to obtain the unloading decision by considering the average minimum task processing delay [22]. These studies only focused on the optimization analysis of the task unloading decision-making and did not study the resource allocation problem.

In the cloud-side network, the intensive deployment of edge servers consumes considerable resources. Many studies have previously been conducted in this field. A resource allocation method was proposed to minimize the energy consumption of wireless power supply system in a cloud-edge computing scenario [23]. A computational resource allocation algorithm was proposed to minimize terminal energy consumption with delay as a constraint [24]. A fair aware communication and computing resource allocation scheme was proposed that aimed at minimizing the maximum loss between users as the objective function for optimization [25]. For the single-cell scenario, reference [23] proposed a task unloading and resource allocation scheme to minimize the energy consumption of the system with wireless power supply capability. Considering the user delay constraint, reference [24] proposed an access control and computing resource allocation algorithm to minimize terminal energy consumption. In the multicell scenario, considering that the user can unload tasks to multiple edge nodes, task allocation of the terminal was optimized to minimize the weighted sum of delay as well as terminal energy consumption [26]. The unloading decision, communication resource allocation (uplink bandwidth and downlink bandwidth), and computing resource allocation of each user were optimized to minimize the energy consumption and delay weight loss for all users under user delay constraints [27]. An optimization problem aimed at minimizing the maximum loss between users was constructed, and a fair and perceptual unloading decision was proposed for the communication and computing resource allocation scheme [25]. The joint scheduling problem of tasks and resources for multiple edge servers in the cloud network was also analyzed [28]. Here, although the resource management under the cloud network-edge end fusion network and load balancing between edge nodes are considered, the correlation between the user unloading decision and resource allocation is ignored. Accordingly, a comprehensive consideration of the user unloading decision and resource allocation is lacking.

On the cloud-side network, the interaction between computing and communication makes the resource optimization problem very complex and difficult to solve. In addition, there are multiple edge nodes in the network with vast differences in their communication computing power and load. In a cloud network-edge integration network, it is important that the nodes cooperate with each other in order to improve network efficiency and performance and realize efficient utilization and resource sharing. Therefore, this study aims to develop a cooperative cloud network-edge fusion network taking maximum and minimum user energy efficiency as the objective function and comprehensively considering user unloading decisions and resource allocation.

The primary contributions of this paper are summarized as follows:

- (i) An edge network model is built for the collaborative cloud network, and the fairness of user energy efficiency is analyzed based on the maximum and minimum criteria.
- (ii) The optimization of user energy efficiency fairness proposed in the cloud network edge-end fusion network model is a mixed-integer nonconvex fractional programming problem which is tough to solve. Using the generalized fraction theory, relaxation variables, and equivalent replacement of the optimization problem, we convert the optimization problem into a convex optimization problem. CEEF- and ADMM-based energy efficiency fairness algorithms are proposed to dispose of the problem.
- (iii) Through simulation, the performance of the proposed CEEF and ADMM energy efficiency fairness algorithms is observed. The efficiency of the proposed algorithm to guarantee user energy efficiency fairness is verified.

## 2. System Model

In this section, a system model for cloud-side network systems, including network, communication, and computation models, is presented.

*2.1. Network Model.* As shown in Figure 1, the cloud-side network model consists of  $I$  user terminals,  $M$  base stations, and a remote cloud server. Each base station in this network is equipped with a MEC server. The MEC server and user terminal sets can be represented as  $M = \{1, 2, \dots, m, \dots, M\}$ , and  $I = \{1, 2, \dots, i, \dots, I\}$ , respectively.

The user terminal has a computing task for processing.  $Q_i = (L_i, K_i)$  describes the computing task, where  $L_i$  represents the data size of the task, and  $K_i$  represents the number of CPU cycles required to accomplish the terminal task. The user terminal is able to choose to deal with the task locally or unload it to the MEC or cloud server. When the user terminal sends the task to the MEC server, the MEC server can process the task by itself, forward it to other MEC

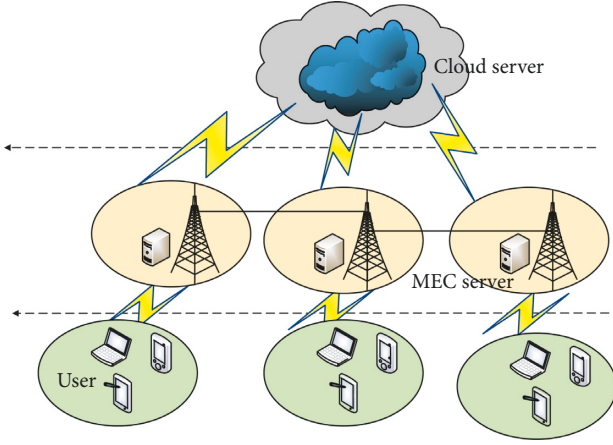


FIGURE 1: Cloud edge networks model.

TABLE 1: Notation and corresponding description.

$Q_i$	Computation task of user $i$
$L_i$	Data size of task
$K_i$	CPU cycles required to accomplish the task
$x_i, y_{i,m}, z_i$	Task offload decision
$R_{i,m}$	Uplink transmission rate between user $i$ and MEC server $m$
$p_i$	Uplink transmission power of user $i$
$f_i^L$	Local computing capacity of user $i$
$T_i^L$	Local calculation delay
$E_i^L$	Locally calculated energy consumption
$E_{i,m}^{comm}$	Transmission energy consumption
$T_{i,m}^{comp}$	MEC calculation delay
$T_i^c$	Cloud computing delay
$T_i$	Total delay
$E_i$	Total energy consumption

servers with richer computing resources, or further unload the task to the cloud server for processing. Table 1 summarizes the used notations and the corresponding definition in this paper.

Binary variables  $x_i, y_{i,m}, z_i \in \{0, 1\}$  are defined, where  $x_i = 1$  shows that the task of the user terminal  $i$  is processed locally and  $x_i = 0$  means that the task is not processed locally. Similarly,  $y_{i,m} = 1$  indicates that the task of user terminal  $i$  is processed by a MEC server  $m$  and  $y_{i,m} = 0$  indicates that the task is not processed on the MEC server  $m$ . Additionally,  $z_i = 1$  indicates that the task of user terminal  $i$  is processed by a cloud server and  $z_i = 0$  indicates that the task of user terminal  $i$  is not processed by the cloud server. The unloading decision to be made by the user terminal for computing tasks needs to meet the following constraints:

$$x_i + \sum_{m=1}^M y_{i,m} + z_i = 1, i \in I. \quad (1)$$

This constraint means that the computing task of each user terminal can only be processed by one of three means: local processing, MEC server processing, and cloud server processing. This implies that, for a user terminal  $i$ , only one binary variable can be equal to 1.

**2.2. Communication Model.** In the cloud-edge networks model, when the user terminal  $i$  sends a task to the MEC server  $m$ , the channel gain is expressed as

$$G_{i,m} = g_{i,m} d_{i,m}^{-\alpha}, \quad (2)$$

where  $g_{i,m}$  is the channel power gain coefficient when the user terminal  $i$  sends tasks to the MEC server  $m$ ,  $d_{i,m}$  is the distance between user terminal  $i$  and MEC server  $m$ , and  $\alpha$  is the path loss factor.

Assuming that the user movement rate is minimal during unloading calculations,  $g_{i,m}$  can be considered to be a constant. The uplink transmission rate between  $i$  and  $m$  is

$$R_{i,m} = B \log_2 \left( 1 + \frac{G_{i,m} p_i}{\sum_{j \in I \setminus \{i\}} G_{j,m} p_j + \sigma^2} \right), \quad (3)$$

where  $B$  is the available spectrum bandwidth,  $p_i$  is the uplink transmission power of user terminal  $i$ , and  $\sigma^2$  is the noise power.

**2.3. Computation Model.** The local computing capacity of  $i$  is represented by  $f_i^L$ , and the power consumed during local computing is represented by  $P_i^L$ . Therefore, when  $i$  processes the task  $Q_i$  locally, the calculation delay can be expressed as

$$T_i^L = \frac{K_i}{f_i^L}. \quad (4)$$

The energy consumed can be expressed as

$$E_i^L = P_i^L T_i^L = P_i^L \frac{K_i}{f_i^L}. \quad (5)$$

When  $i$  sends the task to  $m$  for processing, the computing power of  $m$  is represented by  $F_m$  and the computing resources allocated to  $i$  are represented by  $f_{i,m}$ . When the user terminal  $i$  sends a task, the transmission delay can be expressed as

$$T_{i,m}^{comm} = \frac{L_i}{R_{i,m}}. \quad (6)$$

The energy consumed during transmission by  $i$  can be expressed as

$$E_{i,m}^{comm} = p_i T_{i,m}^{comm} = p_i \frac{L_i}{R_{i,m}}. \quad (7)$$

Accordingly, the calculation delay of  $m$  for task processing can be expressed as

$$T_{i,m}^{comp} = \frac{K_i}{f_{i,m}}. \quad (8)$$

When  $i$  unloads the task on  $m$ ,  $m$  can choose to process the task itself or send it to other MEC servers with richer computing resources. The order  $T_{m,k}$  indicates the average round-trip time of task forwarding between the MEC server  $m$  and another MEC server  $k$ . When  $m = k$ ,  $T_{m,k} = 0$ . Therefore, when  $i$  transmits a task to  $m$  for processing, the total delay is divided into three parts: the transmission delay

between  $i$  and  $m$ , the forwarding delay between  $m$  and  $k$ , and the calculation delay after  $k$  receives the task. The total delay can therefore be expressed as

$$T_{i,k} = T_{i,m}^{comm} + T_{m,k} + T_{i,k}^{comp}. \quad (9)$$

When  $i$  unloads a task to the cloud server for processing, the cloud computing capacity allocated for it is represented by  $f_i^c$ , the average round-trip time of task transmission between  $m$  and the cloud server is represented by  $T_c$ , and the processing delay of the task in the cloud server can be expressed as

$$T_i^{fc} = \frac{K_i}{f_i^c}. \quad (10)$$

The size of the returned data after processing is considerably smaller than that before processing. Hence, the transmission delay of the result is negligible. When the cloud server processes the task of  $i$ , the total delay of the entire process can be expressed as

$$T_i^C = T_{i,m}^{comm} + T_c + T_i^{fc}. \quad (11)$$

To sum up, the total delay of user terminal  $i$  in task processing can be expressed as

$$T_i = x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C. \quad (12)$$

The total energy consumption can be expressed as

$$E_i = x_i E_i^L + \sum_{m=1}^M y_{i,m} E_{i,m}^{comm} + z_i E_{i,m}^{comm}. \quad (13)$$

### 3. User Energy Efficiency Fairness Resource Allocation Algorithm

First, this section formulates the resource allocation and task unloading optimization problems based on the maximum-minimum criterion and then solves them using the CEEF and ADMM algorithms, respectively.

**3.1. Problem Reformulation.** The maximum-minimum criterion is an effective means to ensure fairness for all users. Hence, it was used to construct the joint resource allocation and task unloading optimization problem. The unloading decision vector for  $i$  is expressed as  $\psi_i = \{x_i, y_{i,1}, y_{i,2}, \dots, y_{i,M}, z_i\}$ , and the joint optimization problem is expressed as follows:

$$\max_{\psi_i, f_{i,m}} \min \frac{\sum_{m=1}^M y_{i,m} R_{i,m} + z_i R_{i,m}}{x_i E_i^L + \sum_{m=1}^M y_{i,m} E_{i,m}^{comm} + z_i E_{i,m}^{comm}}, \quad (14)$$

$$s.t. x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C \leq T_i^{\max}, \quad (15)$$

$$\sum_{i=1}^I f_{i,m} \leq F_m, \quad (16)$$

$$x_i + \sum_{m=1}^M y_{i,m} + z_i = 1, \quad (17)$$

$$x_i, y_{i,m}, z_i \in \{0, 1\}. \quad (18)$$

Equation (14) represents the objective function of joint resource allocation and task unloading, equation (15) represents the user terminal requirements for the delay in the task unloading process, equation (16) shows that the computing resources of the base station cannot exceed the maximum computing capacity, and equations (17) and (18) indicate that only one node can be selected for computing the required tasks.

**3.2. Solutions Using the CEEF Algorithm.** The user energy efficiency fairness optimization problem based on the maximum and minimum criteria is a mixed-integer non-convex fractional programming problem. First, the problem is transformed into an equivalent mixed-integer nonconvex subtraction optimization problem using the generalized fractional theory. Let the variable  $Q$  represent the value of the optimization problem, that is, the maximum and minimum energy efficiency. Using the generalized fraction theory and introducing a relaxation variable  $\theta$ , the problem can be transformed into the following:

$$\max_{\psi_i, f_{i,m}} \theta, \quad (19)$$

$$s.t. x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C \leq T_i^{\max}, \quad (20)$$

$$\sum_{i=1}^I f_{i,m} \leq F_m, \quad (21)$$

$$x_i + \sum_{m=1}^M y_{i,m} + z_i = 1, \quad (22)$$

$$x_i, y_{i,m}, z_i \in \{0, 1\}, \quad (23)$$

$$\sum_{m=1}^M y_{i,m} R_{i,m} + z_i R_{i,m} - Q E_i^{\text{total}} \geq \theta, \quad (24)$$

where

$$E_i^{\text{total}} = x_i E_i^L + \sum_{m=1}^M y_{i,m} E_{i,m}^{comm} + z_i E_{i,m}^{comm}. \quad (25)$$

As  $x_i, y_{i,m}, z_i$  are binary variables, it can be relaxed to  $x_i, y_{i,m}, z_i \in [0, 1]$ . The optimization problem is further modified as follows:

$$\max_{\psi_i, f_{i,m}} \theta, \quad (26)$$

$$\text{s.t. } x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C \leq T_i^{\max}, \quad (27)$$

$$\sum_{i=1}^I f_{i,m} \leq F_m, \quad (28)$$

$$x_i + \sum_{m=1}^M y_{i,m} + z_i = 1, \quad (29)$$

$$x_i, y_{i,m}, z_i \in [0, 1], \quad (30)$$

$$\sum_{m=1}^M y_{i,m} R_{i,m} + z_i R_{i,m} - Q E_i^{\text{total}} \geq \theta. \quad (31)$$

The objective function equation (26) is a linear function, and equations (28)–(31) are linear constraints. The total delay of the user terminal using equation (27) is expressed as

$$\begin{aligned} & x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C \\ &= x_i T_i^L + \sum_{m=1}^M y_{i,m} \frac{L_i}{R_{i,m}} + \sum_{m=1}^M y_{i,m} \frac{K_i}{f_{i,m}} \\ &+ \sum_{m=1}^M y_{i,m} T_{k,m} + z_i T_i^C. \end{aligned} \quad (32)$$

The product term  $y_{i,m}/f_{i,m}$  couples the variables  $y_{i,m}$  and  $f_{i,m}$ . Additional variables  $c_{i,m} = 1/f_{i,m}$  and  $V_{i,m} = y_{i,m} c_{i,m}$  are introduced to solve the aforementioned equation. Therefore, the problem is described as follows:

$$\max_{\psi_i, c_{i,m}, V_{i,m}} \theta, \quad (33)$$

$$\text{s.t. } x_i T_i^L + \sum_{m=1}^M y_{i,m} T_{i,m} + z_i T_i^C \leq T_i^{\max}, \quad (34)$$

$$\sum_{i=1}^I \frac{1}{c_{i,m}} \leq F_m, \quad (35)$$

$$x_i + \sum_{m=1}^M y_{i,m} + z_i = 1, \quad (36)$$

$$x_i, y_{i,m}, z_i \in [0, 1], \quad (37)$$

$$\sum_{m=1}^M y_{i,m} R_{i,m} + z_i R_{i,m} - Q E_i^{\text{total}} \geq \theta, \quad (38)$$

$$V_{i,m} \geq \frac{y_{i,m}}{F_m}, \quad (39)$$

$$V_{i,m} \leq \frac{y_{i,m}}{F_{\min}}, \quad (40)$$

where  $F_{\min}$  is a normal number close to 0. In order to solve the aforementioned problem, the CEEF algorithm is proposed. Its steps are shown in Algorithm 1.

As the unloading decision variable is substituted into a continuous variable,  $x_i, y_{i,m}, z_i$  need to be mapped to a binary variable after the optimal solution is obtained. The recovery scheme is as follows:

$$x_i = \begin{cases} 1, & \text{if } i = \operatorname{argmax}_{i \in I} x_i \\ 0, & \text{else} \end{cases} \quad (41)$$

All the other variables were also recovered similarly.

**3.3. Solutions Using the ADMM Algorithm.** The ADMM algorithm is a standard method to solve large-scale optimization problems. Combined with CEEF, it can effectively achieve user energy efficiency fairness.

First, the optimization variables  $x_i, y_{i,m}$ , and  $z_i$  are copied and redefined as  $\hat{x}_i, \hat{y}_{i,m}$ , and  $\hat{z}_i$  to form the following equation:

$$\begin{cases} x_i = \hat{x}_i \\ y_{i,m} = \hat{y}_{i,m} \\ z_i = \hat{z}_i \end{cases} \quad (42)$$

Equation (33) can be equivalently modified as follows:

$$\max_{\hat{x}_i, \hat{y}_{i,m}, \hat{z}_i, c_{i,m}, V_{i,m}} \theta, \quad (43)$$

$$\text{s.t. } \hat{x}_i T_i^L + \sum_{m=1}^M \hat{y}_{i,m} T_{i,m} + \hat{z}_i T_i^C \leq T_i^{\max}, \quad (44)$$

$$\sum_{i=1}^I \frac{1}{c_{i,m}} \leq F_m, \quad (45)$$

$$\hat{x}_i + \sum_{m=1}^M \hat{y}_{i,m} + \hat{z}_i = 1, \quad (46)$$

$$\hat{x}_i, \hat{y}_{i,m}, \hat{z}_i \in [0, 1], \quad (47)$$

$$\sum_{m=1}^M \hat{y}_{i,m} R_{i,m} + \hat{z}_i R_{i,m} - Q \hat{E}_i^{\text{total}} \geq \theta, \quad (48)$$

$$V_{i,m} \geq \frac{\hat{y}_{i,m}}{F_m}, \quad (49)$$

$$V_{i,m} \leq \frac{\hat{y}_{i,m}}{F_{\min}}. \quad (50)$$

Therefore, the augmented Lagrangian function from equation (43) can be expressed as

**Initialization:**  
 Set  $t = 1$ ,  $Q^{(0)} = 0$ ,  $\varepsilon = 0.01$ ,  $t_{\max} = 25$ .  
 Substitute  $Q^{(0)} = 0$  into the problem equation (33) to obtain an initial feasible solution  $\Pi^{(0)}$ .

**Iteration:**  
**while**  $Q^{(t)} - Q^{(t-1)} > \varepsilon$  and  $t \leq t_{\max}$  **do**  
 (1) Substitute  $Q^{(t-1)}$  into the problem equation (31) and update feasible solution  $\Pi^{(t)}$ ;  
 (2) Solve the problem equation (15) by substituting the feasible solution  $\Pi^{(t)}$  and update  $Q^{(t)}$ ;  
 $t = t + 1$ .  
**end while**

ALGORITHM 1: CEEF algorithm.

$$\begin{aligned}
 L_\rho(\hat{x}_i, \hat{y}_{i,m}, \hat{z}_i, \lambda_i, \mu_{i,m}, \gamma_i) &= \theta + \lambda_i (\hat{x}_i - x_i) \\
 &+ \sum_{m=1}^M \mu_{i,m} (\hat{y}_{i,m} - y_{i,m}) \\
 &+ \gamma_i (\hat{z}_i - z_i) + \frac{\rho}{2} (\hat{x}_i - x_i)^2 \quad (51) \\
 &+ \frac{\rho}{2} \sum_{m=1}^M (\hat{y}_{i,m} - y_{i,m})^2 \\
 &+ \frac{\rho}{2} (\hat{z}_i - z_i)^2,
 \end{aligned}$$

where  $\lambda_i$ ,  $\mu_{i,m}$ , and  $\gamma_i$  represent the Lagrange multipliers and  $\rho > 0$  represents the augmented Lagrange parameter, which is a constant. This constant controls the convergence of the ADMM algorithm iteration. To facilitate the solution of the target variable, the Lagrange multiplier coefficients are simplified. Hence, the augmented Lagrange function can be equivalently transformed into the following:

$$\begin{aligned}
 L_\rho(\hat{x}_i, \hat{y}_{i,m}, \hat{z}_i, \lambda_i, \mu_{i,m}, \gamma_i) &= \theta + \frac{\rho}{2} \left( \hat{x}_i - x_i + \frac{\lambda_i}{\rho} \right)^2 + \frac{\rho}{2} \sum_{m=1}^M \left( \hat{y}_{i,m} - y_{i,m} + \frac{\mu_{i,m}}{\rho} \right)^2 \\
 &+ \frac{\rho}{2} \left( \hat{z}_i - z_i + \frac{\gamma_i}{\rho} \right)^2. \quad (52)
 \end{aligned}$$

When using ADMM to solve the problem, the variables need to be updated through multiple iterations. Let

$$e^{(t)} = (\hat{x}_i^{(t)}, \hat{y}_{i,m}^{(t)}, \hat{z}_i^{(t)}), \quad (53)$$

$$s^{(t)} = (x_i^{(t)}, y_{i,m}^{(t)}, z_i^{(t)}), \quad (54)$$

$$d^{(t)} = (\lambda_i^{(t)}, \mu_{i,m}^{(t)}, \gamma_i^{(t)}). \quad (55)$$

represent the value of the  $t^{\text{th}}$  iteration of the variables. The first step to solve the equations is to update the local variable, which is expressed as follows:

$$e^{t+1} = \arg_e \max L_\rho(e, s^{(t)}, d^{(t)}). \quad (56)$$

This problem can be decomposed into  $m$  parallel sub-problems, where each subproblem is solved separately using convex optimization.

The second step is updating the global variable  $s$  using the following formula:

$$x_i^{(t+1)} = \operatorname{argmax}_{x_i} \frac{\rho}{2} \left( \hat{x}_i^{(t+1)} - x_i + \frac{\lambda_i^{(t)}}{\rho} \right)^2, \quad (57)$$

$$y_{i,m}^{(t+1)} = \operatorname{argmax}_{y_{i,m}} \frac{\rho}{2} \sum_{m=1}^M \left( \hat{y}_{i,m}^{(t+1)} - y_{i,m} + \frac{\mu_{i,m}^{(t)}}{\rho} \right)^2, \quad (58)$$

$$z_i^{(t+1)} = \operatorname{argmax}_{z_i} \frac{\rho}{2} \left( \hat{z}_i^{(t+1)} - z_i + \frac{\gamma_i^{(t)}}{\rho} \right)^2. \quad (59)$$

The aforementioned problem is an unconstrained quadratic convex problem. After deriving the variable, the following equation is obtained:

$$\rho(\hat{x}_i^{(t+1)} - x_i + \lambda_i^{(t)}) = 0, \quad (60)$$

$$\rho \sum_{m=1}^M (\hat{y}_{i,m}^{(t)} - y_{i,m} + \mu_{i,m}^{(t)}) = 0, \quad (61)$$

$$\rho(\hat{z}_i^{(t+1)} - z_i + \gamma_i^{(t)}) = 0. \quad (62)$$

The global variable optimal solution is obtained as follows:

$$x_i^{(t+1)} = \hat{x}_i^{(t+1)} + \lambda_i^{(t)}, \quad (63)$$

$$y_{i,m}^{(t+1)} = \frac{1}{M} \sum_{m=1}^M (\hat{y}_{i,m}^{(t+1)} + \mu_{i,m}^{(t)}), \quad (64)$$

$$z_i^{(t+1)} = \hat{z}_i^{(t+1)} + \gamma_i^{(t)}. \quad (65)$$

The third step is updating the Lagrange multiplier as follows:

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \hat{x}_i^{(t+1)} - x_i^{(t+1)}, \quad (66)$$

$$\mu_{i,m}^{(t+1)} = \mu_{i,m}^{(t)} + \hat{y}_{i,m}^{(t+1)} - y_{i,m}^{(t+1)}, \quad (67)$$

$$\gamma_i^{(t+1)} = \gamma_i^{(t)} + \hat{z}_i^{(t+1)} - z_i^{(t+1)}. \quad (68)$$

**Initialization:**  
Initialize the feasible set  $x_i^{(t)}, y_{i,m}^{(t)}, z_i^{(t)}$ ,  
set the stopping criteria values  $\varepsilon_{pri}$  and  $\varepsilon_{dual}$ ,  
scaling Lagrange multipliers vectors  $\lambda_i^{(t)}, \mu_{i,m}^{(t)}, \gamma_i^{(t)}$ , set  $t = 0$ ;  
**while**  $\|x_i^{(t+1)} - x_i^{(t)}\|_2 \geq \varepsilon_{dual}, \|y_{i,m}^{(t+1)} - y_{i,m}^{(t)}\|_2 \geq \varepsilon_{dual}, \|z_i^{(t+1)} - z_i^{(t)}\|_2 \geq \varepsilon_{dual}$  **do**  
(1) Update local variables  $\hat{x}_i^{(t+1)}, \hat{y}_{i,m}^{(t+1)}, \hat{z}_i^{(t+1)}$  by solving problem equation (56);  
(2) Update global variables  $x_i^{(t+1)}, y_{i,m}^{(t+1)}, z_i^{(t+1)}$  by equations (63)–(65);  
(3) Update multipliers  $\lambda_i^{(t+1)}, \mu_{i,m}^{(t+1)}, \gamma_i^{(t+1)}$  by equations (66)–(68);  
 $t = t + 1$ .  
**end while**

ALGORITHM 2: ADMM algorithm.

The fourth step verifies the iteration termination condition, expressed as follows:

$$\|\hat{x}_i^{(t+1)} - x_i^{(t+1)}\|_2 \leq \varepsilon_{pri}, \quad (69)$$

$$\|\hat{y}_{i,m}^{(t+1)} - y_{i,m}^{(t+1)}\|_2 \leq \varepsilon_{pri}, \quad (70)$$

$$\|\hat{z}_i^{(t+1)} - z_i^{(t+1)}\|_2 \leq \varepsilon_{pri}, \quad (71)$$

$$\|x_i^{(t+1)} - x_i^{(t)}\|_2 \leq \varepsilon_{dual}, \quad (72)$$

$$\|y_{i,m}^{(t+1)} - y_{i,m}^{(t)}\|_2 \leq \varepsilon_{dual}, \quad (73)$$

$$\|z_i^{(t+1)} - z_i^{(t)}\|_2 \leq \varepsilon_{dual}, \quad (74)$$

where  $\varepsilon_{pri} > 0$  and  $\varepsilon_{dual} > 0$  represent the iteration stop threshold under feasible conditions.

The energy efficiency fairness algorithm based on the ADMM algorithm described here is shown in Algorithm 2.

As the unloading decision variable is modified into a continuous variable,  $x_i, y_{i,m}, z_i$  need to be mapped to a binary variable after the optimal solution is obtained. The recovery scheme is as follows:

$$x_i = \begin{cases} 1, & \text{if } i = \operatorname{argmax}_{i \in I} x_i \\ 0, & \text{else} \end{cases} \quad (75)$$

All the other variables were recovered in the same way.

**Complexity Analysis:** when using CEFF algorithm to solve the problem, the algorithm complexity is  $O(I^2(M+1))$ . For the proposed ADMM-based distributed algorithm, in the local variable update, the computational complexity is  $O(I^3)$ . In the global variable update, the corresponding calculation complexity is  $O(I(M+1))$ . In addition, the complexity of updating Lagrange multiplier is  $O(I)$ . Therefore, the computational complexity of each iteration of algorithm is  $O(I^3) + O(I(M+1)) + O(I) \approx O(I^3)$ .

## 4. Simulation Results

In this section, the proposed algorithms are compared by simulation experiments. The simulation is composed of 3 base stations and 27 random users. The radius covered by

each base station is 500 m, and the users are randomly distributed in the overlapping area covered by the base station. The transmission link bandwidth in the system is 15 MHz, noise power is  $\sigma^2 = -143$  dBW, path loss model is  $h = 128.1 + 37.6 \log_{10}(d)$ , and  $d$  represents the distance between the MEC server and user terminal. The local computing capacity of the user terminal, base station, and cloud server is 0.6 Gcycles/s, 5 Gcycles/s, and 10 Gcycles/s, respectively. To achieve a performance comparison, a noncooperative scheme is simulated in addition to the CEFF and ADMM [29]. In the noncooperative scheme, computing tasks could be processed by the user terminal, MEC server, or cloud server, but the cooperation between MEC servers is not considered. Computing tasks cannot be forwarded to other MEC servers, and the maximum-minimum user energy efficiency cannot be achieved by optimizing resource allocation.

Figure 2 shows the convergence process of the CEFF algorithm, ADMM algorithm, and noncooperative algorithm. The abscissa represents the number of iterations, and the ordinate represents the energy efficiency. It can be seen from the simulation graph that with an increase in the number of iterations, the three schemes stabilize at their unique value, thereby verifying the convergence of the algorithms. The performance of CEFF and ADMM algorithms is 30.76% higher than that of the noncooperative scheme. This is because the computing tasks in the noncooperative scheme cannot be forwarded among MEC servers, thereby increasing energy consumption and decreasing energy efficiency.

Figure 3 shows the impact of different task calculations on user energy efficiency performed by the CEFF, ADMM, and noncooperative algorithms when the transmission power of the user terminal is 0.2 W and 0.5 W. The abscissa represents computations required for the task, and the ordinate represents energy efficiency. The energy efficiency of the three schemes gradually decreases with an increase in computations. As the number of calculations increases, the energy consumption increases, thereby decreasing energy efficiency. The figure also shows that the energy efficiency trend of the CEFF and ADMM algorithms is consistent and greater than that of the noncooperative scheme. We can hence conclude that the performance of the cooperative schemes is better than that of noncooperative scheme under varied computational loads.

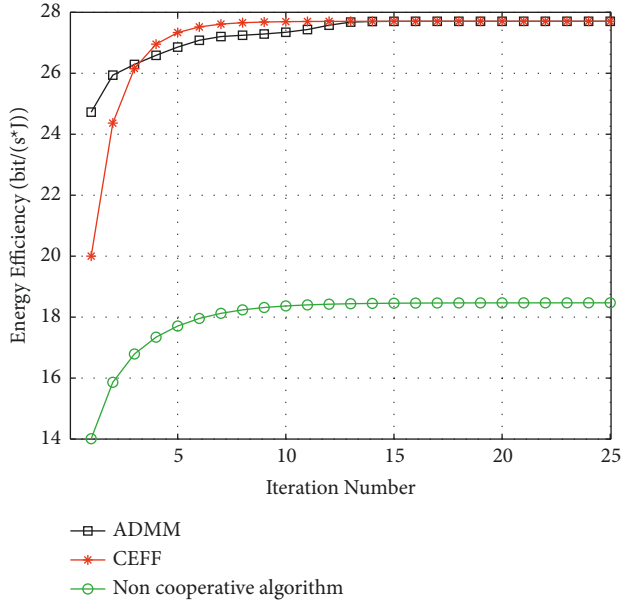


FIGURE 2: Convergence graph of the three algorithms.

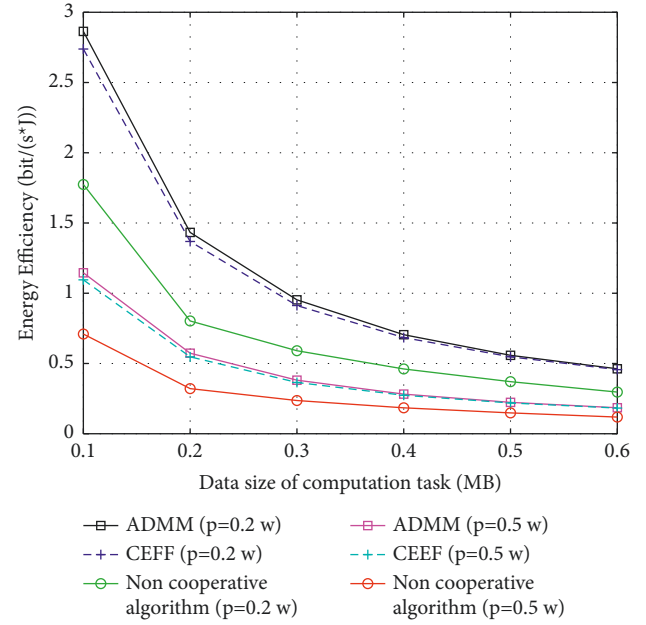


FIGURE 4: Impact of task data size on energy efficiency.

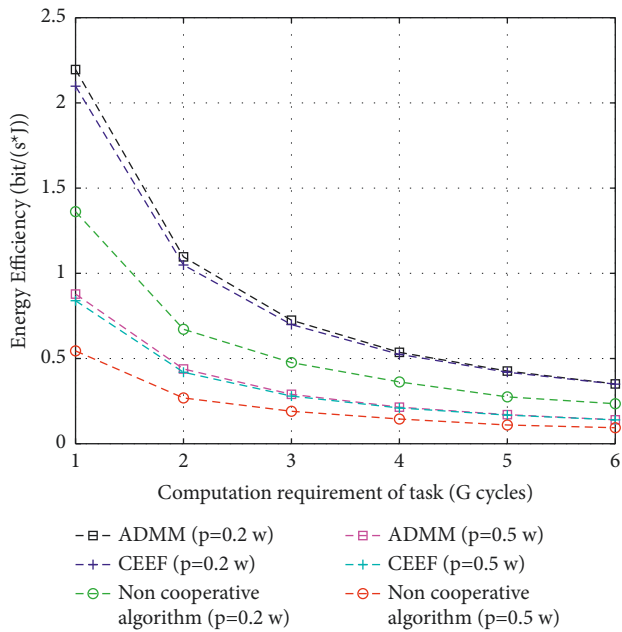


FIGURE 3: Impact of computational requirements for the task on energy efficiency.

Figure 4 shows the impact of different data sizes of a task on user energy efficiency under the CEEF, ADMM, and noncooperative algorithms when the transmission power of the user terminal is 0.2 W and 0.5 W. The abscissa in the figure represents the data size of the task, and the ordinate represents the energy efficiency. An increase in the data size of the task gradually decreases the efficiency of the three algorithms. When the data size of a task increases, the energy consumed by the system for calculations increases, thereby decreasing energy efficiency. The figure also shows that the energy efficiency

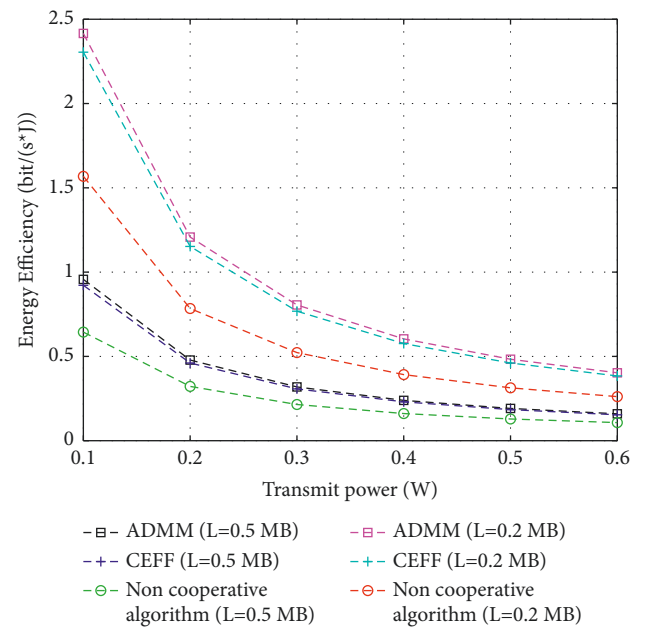


FIGURE 5: Impact of transmission power on energy efficiency.

trend of the CEEF and ADMM algorithms is consistent and greater than that of the noncooperative scheme. We can hence conclude that the performance of the CEEF and ADMM algorithms is better than that of the noncooperative scheme for varied data sizes.

Figure 5 shows the impact of transmission power of the user terminal on user energy efficiency of the CEEF, ADMM, and noncooperative algorithms when the task data size is 0.2 Mb and 0.5 Mb. The abscissa represents the transmission power, and ordinate represents the energy efficiency. With an increase in transmission power, the energy efficiency of the



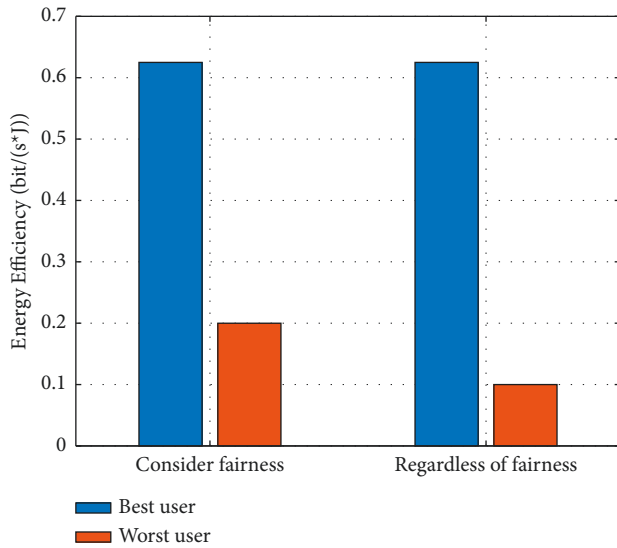


FIGURE 6: Comparison between optimization considering and not considering fairness.

three schemes gradually decreases. When the transmission power gradually increases, the energy consumed by the system for transmission calculation increases more than the transmission rate, resulting in reduced energy efficiency. The figure also shows that the energy efficiency trend of the CEEF and ADMM algorithms is consistent and greater than that of the noncooperative scheme. We can conclude that the CEEF and ADMM algorithms perform better than the noncooperative algorithm when the user transmission power is varied.

Figure 6 shows a comparison between considering and not considering fairness. The difference between the two is that the objective function of optimization is different. The objective function without considering fairness is that the maximum total energy efficiency, that is, the ratio of the total user rate to the total user energy consumption should be the largest. The figure shows that when fairness is not considered, the energy efficiency gap between the best users and the worst users is relatively large. We can conclude that the maximum and minimum user energy efficiency considered in this paper can better ensure the fair access of resources for all users.

## 5. Conclusion

This paper develops a cloud network-edge network model and analyzes the fairness of user energy efficiency based on the maximum-minimum criterion. The user energy efficiency fairness optimization problem proposed in the cloud network-edge network model is a mixed-integer nonconvex fractional programming problem, which is difficult to deal with. Through the generalized fraction theory, relaxation variables, and equivalent replacement of the optimization problem, the problem was modified into a convex optimization problem. Accordingly, CEEF- and ADMM-based energy efficiency fairness algorithms are proposed to dispose of the problem. Simulations are performed to test the proposed algorithm and verify that the proposed algorithm can guarantee user energy efficiency fairness.

## Data Availability

Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The work was supported in part by Subproject of National Key Research and Development Plan in 2020 (Grant No. 2020YFC1511704), Beijing Science and Technology Project (Grant no. Z211100004421009), Beijing Information Science & Technology University (Grant Nos. 2020KYNH212 and No. 2021CGZH302), and in part by the National Natural Science Foundation of China (Grant no. 61971048).

## References

- [1] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [2] K. Peng, V. C. M. Leung, X. Xu, L. Zheng, J. Wang, and Q. Huang, "A survey on mobile edge computing: focusing on service adoption and provision," *Wireless Communications and Mobile Computing*, vol. 2018, p. 1, Article ID 8267838, 2018.
- [3] M. Merluzzi, P. D. Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 342–356, 2020.
- [4] Z. Qu, H. Sun, and M. Zheng, "An efficient quantum image steganography protocol based on improved EMD algorithm," *Quantum Information Processing*, vol. 20, no. 2, pp. 53–29, 2021.
- [5] Z. Qu, Y. Huang, and M. Zheng, "A novel coherence-based quantum steganalysis protocol," *Quantum Information Processing*, vol. 19, no. 10, pp. 362–425, 2020.
- [6] Z. Qu, S. Chen, and X. Wang, "A secure controlled quantum image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 10, pp. 380–425, 2020.
- [7] L. Meng, W. Tan, J. Ma, R. Wang, X. Yin, and Y. Zhang, "Enhancing dynamic ECG heartbeat classification with lightweight transformer model," *Artificial Intelligence in Medicine*, vol. 124, Article ID 102236, 2022.
- [8] W. Tan, P. Liu, X. Li, S. Xu, Y. Chen, and J. Yang, "Segmentation of lung airways based on deep learning methods," *IET Image Processing*, vol. 16, no. 5, pp. 1444–1456, 2022.
- [9] M. Wu, L. Tan, and N. Xiong, "A structure fidelity approach for big data collection in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 248–273, 2014.
- [10] S. Huang, A. Liu, S. Zhang, T. Wang, and N. N. Xiong, "BD-VTE: A novel baseline data based verifiable trust evaluation scheme for smart network systems," *IEEE Transactions on Network Science and Engineering*, vol. 8, pp. 2087–2105, 2020.
- [11] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

- [12] M. Patel, B. Naughton, and C. Chan, "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.
- [13] Z. Xiao, X. Dai, H. Jiang et al., "Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020.
- [14] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7867–7881, 2020.
- [15] J. Dai, Z. Zhang, S. Mao, and D. Liu, "A view synthesis-based 360°VR caching system over MEC-enabled C-ran," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3843–3855, 2020.
- [16] W. Shi, H. Sun, J. Cao, and Q. Zhang, "Edge computing: an emerging computing model for the internet of everything era," *Journal of Computer Research and Development*, vol. 54, no. 5, pp. 907–924, 2017.
- [17] Y. Wu and J. Zheng, "Modeling and analysis of the downlink local delay in MEC-based VANETs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6619–6630, 2020.
- [18] I. Sarrigiannis, K. Ramantas, E. Kartsakli, P. V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Online VNF lifecycle management in an MEC-enabled 5G IoT architecture," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4183–4194, 2020.
- [19] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1634–1644, 2021.
- [20] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4804–4814, 2019.
- [21] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4642–4655, 2018.
- [22] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11098–11112, 2018.
- [23] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2018.
- [24] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2603–2616, 2018.
- [25] 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.
- [26] T. Q. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.
- [27] M. H. Chen, M. Dong, and B. Liang, "Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2868–2881, 2018.
- [28] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.
- [29] Y. Wang, X. Tao, X. Zhang, P. Zhang, and Y. T. Hou, "Co-operative task offloading in three-tier mobile computing networks: an ADMM framework," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2763–2776, 2019.