sensors

MDPI

Article

Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing

Yan-Yun Huang and Pi-Chung Wang *

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan
* Correspondence: pcwang@nchu.edu.tw; Tel.: +886-4-2284-0497

Abstract: Mobile devices may use mobile edge computing to improve energy efficiency and responsiveness by offloading computation tasks to edge servers. However, the transmissions of mobile devices may result in interference that decreases the upload rate and prolongs transmission delay. Clustering has been shown as an effective approach to improve the transmission efficiency for dense devices, but there is no distributed algorithm for the optimization of clustering and computation offloading. In this work, we study the optimization problem of computation offloading to minimize the energy consumption of mobile devices in mobile edge computing by adaptively clustering devices to improve the transmission efficiency. To address the optimization problem in a distributed manner, the decision problem of clustering and computation offloading for mobile devices is formulated as a potential game. We introduce the construction of the potential game and show the existence of Nash equilibrium in the game with a finite enhancement ability. Then, we propose a distributed algorithm of clustering and computation offloading based on game theory. We conducted a simulation to evaluate the proposed algorithm. The numerical results from our simulation show that our algorithm can improve offloading efficiency for mobile devices in mobile edge computing by improving transmission efficiency. By offloading more tasks to edge servers, both the energy efficiency of mobile devices and the responsiveness of computation-intensive applications can be improved simultaneously.

Keywords: computation offloading; clustering; game theory; Nash equilibrium; mobile edge computing

Citation: Huang, Y.-Y.; Wang, P.-C. Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing. *Sensors* **2023**, *23*, 1155. https://doi.org/10.3390/s23031155

Academic Editor: Paolo Gastaldo

Received: 19 December 2022
Revised: 15 January 2023
Accepted: 16 January 2023
Published: 19 January 2023

1. Introduction

Various computation-intensive mobile applications, such as online gaming, machine learning, and virtual/augmented reality, have been developed. Some of these applications may have a delay-constraint requirement, but satisfying the requests of these applications from mobile devices (MDs) is difficult because these devices have only restricted resources [1]. Mobile edge computing has emerged as a crucial 5G technology. MEC servers are deployed at basestations in the close proximity to MDs to provide the capability of storage and computation for MDs [2]. This technology provides the benefits of improving transmission quality and efficient network operation [3].

Although mobile edge computing could improve the application performance for MDs, the simultaneous transmissions of MDs may degrade channel quality. As a result, the transmission performance is degraded, prolonging the response latency [4]. Although multi-channel communication can be applied to improve transmission performance by assigning MDs to different channels [5], the spectrum resources may still not be enough to accommodate all MDs [6].

To ease the communication overhead for MEC, it is possible to generate clusters of MDs, where only the cluster head of each cluster communicates with the basestation. In each cluster, the cluster members with computation offloadings transmit their data to the cluster head and the cluster head transmits data to the basestation on behalf of all MDs in the cluster. With the MD clustering, the transmission performance can be improved

by allocating spectrum resources to cluster heads. However, the optimization problem of clustering and computation offloading for MDs has not been addressed in a distributed manner in the previous literature.

In this paper, we model the problem of clustering and computation offloading for MDs as a competitive game, where each MD attempts to minimize its overhead by independently adjusting its clustering and offloading decision until a Nash equilibrium is reach. As compared with a centralized approach, the competitive game enables a distributed model for better flexibility and scalability. We summarize the contributions of this work as below:

1. *Clustering and computation offloading for MDs*: To cope with the communication overhead caused by the interference among MDs with computation offloading, we propose a system model that clusters MDs, where only cluster heads can communicate with the basestation. Cluster members can forward their requests through their cluster heads to reduce the number of concurrent transmissions and improve the transmission performance.
2. *Formulation of a distributed competitive game*: Based on the system model, a potential game where each MD selfishly selects a decision to minimize its overhead is formulated. For the proposed game, we show the existence of Nash equilibrium, where no MD can achieve better performance by altering its strategy. With the existence of a Nash equilibrium, the game will converge within limited number of iterations.
3. *An algorithm of the distributed clustering and computation offloading game*: We developed an efficient algorithm, namely, distributed clustering and computation offloading (DCCO). Since the algorithm is performed in a distributed manner, the decision overhead is shared by all MDs within a network. Our algorithm is evaluated to show the performance improvement in terms of the number of successful computation offloadings, energy consumption and response delay. The convergence performance of the algorithm is also investigated.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, we present the system model of clustering and computation offloading. Then, we formulate the problem of clustering and computation offloading for MDs as a competitive game in Section 4, where the existence of Nash equilibrium is demonstrated. In Section 5, we present the proposed DCCO algorithm and explore the convergence of the DCCO game. To evaluate the performance of the proposed algorithm, we conducted experiments upon different algorithms and show their numerical results in Section 6. Finally, the conclusions of this paper are provided in Section 7.

2. Related Works

Researchers have exploited the technology of mobile edge computing for Internet of Things (IoT) and in the 5G framework for computation offloading. You et al. studied a near-optimal design with a threshold-based structure for network resource allocation in a multi-user mobile-edge computing offloading (MECO) system based on TDMA/OFDMA [7]. Malik et al. proposed parallel execution for computation tasks [8]. Guan et al. employed neighboring devices for cooperative partial offloading [9]. Zaman et al. incorporated machine learning for mobility prediction to improve resource efficiency [10,11]. The tradeoff between delay and energy consumption can also be addressed by an iterative search algorithm [12]. This algorithm yields the optimal solution in multi-device environments. The above algorithms are performed in a centralized manner for better optimization, but they may suffer from the cost of gathering statistical data from all MDs.

There are also implementations operated in a decentralized manner, where mobile devices make offloading decisions based on their own interests without forwarding statistics data to the MEC server. Tran et al. improved the performance gain of computation offloading for MEC with multiple servers [13]. Dinh et al. optimized the decisions of task allocation and CPU frequency with an offloading framework to minimize both total execution latency and energy consumption [14]. Bi et al. maximized the total computa-

tion rate of all wireless devices based on wireless power transfer for energy-harvesting wireless devices with a computation offloading policy [15]. Neto et al. presented a user-level online offloading framework [16] in which each device is equipped with a decision engine to minimize the remote-execution overhead. Chen et al. developed an online peer offloading structure by using the Lyapunov technique to handle spatially uneven computation workloads and prevent long computation latency in overloaded small-cell basestations [17]. Mazouzi et al. focused on computation offloading over a heterogeneous cloudlet environment for mobile devices whose tasks have different energy and latency constraints [18]. They proposed a heuristic approach of distributed linear relaxation based on the Lagrangian decomposition method.

There have also been studies applying game theory to solve the problem of computation offloading in a distributed framework. Yang et al. took a small cell network with multiple users and multiple MEC servers into account to design a threshold-based game-theoretic approach [19]. Guo et al. introduced a hybrid fiber-wireless (FiWi) network to support a system of integrating a centralized cloud and multi-access edge computing [20]. To address the collaborative computation offloading problem and overcome the drawback of centralized management, an approximation greedy strategy and a game theory as a distributed scheme was developed. He et al. investigated the edge-user-allocation problem from the perspectives of application vendors in edge computing, where users can make their own allocation decisions [21].

According to the previous literature [22], we are aware that one factor affecting the performance of computation offloading is the communication between mobile devices and MEC servers. Channel management was considered in the previous studies of communication. Li et al. investigated the problems of radio and computational resources to improve spectrum efficiency for both static and dynamic tasks from mobile devices for vehicular edge computing [23]. Ning et al. proposed a hybrid computation offloading structure for real-time interactive systems [24]. They formulated a joint problem of task offloading, sub-channel assignment and power allocation. Alsen et al. introduced an MEC system for unmanned aerial vehicles with the optimization problem of task offloading. They also considered bandwidth allocation for IoT devices and resource allocation in local computation [25]. Tun et al. presented a system of virtualized multi-access edge computing, where network bandwidth and computation resource are sliced [26]. Cheng et al. aimed at a multi-user and multi-MEC server scenario based on OFDMA [27]. They jointly investigated task offloading policies and radio resource allocation. In this work, we also use OFDMA for the communications between MDs and basestations.

Clustering is a technique to categorize entities into different groups, and the entities of each group share a certain level of similarity [28]. Numerous approaches have been proposed to address the optimization of clustering [29,30]. Although the technique of clustering is applied to wireless transmission, previous studies of computation offloading rarely exploited the concept of clustering to improve the performance of offloading based on game theory. Hong et al. declared a robust optimization algorithm of energy and power with fault tolerance to overcome diverse electrical power and computing strength [31]. Attiah et al. implemented a game-theoretical clustering framework to allow for the differentiation of a cluster head's obligation between each node to manage energy usage [32]. Afsar et al. aimed at the issue of energy shortage by proposing the splitting network and game-theory-based clustering algorithm [33]. Loomba et al. presented the development of an energy-efficient stochastic leader-selection algorithm [34], which uses the distance between mobile devices and the basestation to select the best cluster head as offloading agent. The algorithm also improves the energy usage of the interaction between mobile handsets and an application server. Bouet et al. used a MEC-clustering algorithm to consolidate edge communications [35]. Du et al. introduced a fast and self-adaptive clustering algorithm to obtain a accurate density threshold by optimizing the sum of squared errors formula with limited iterations [36]. He et al. offloaded computational tasks in MEC-based ultra dense networks from the core network by the perception of AP-clustering [37].

To our knowledge, there is no distributed clustering algorithm for MEC computation offloading. We are thus motivated to combine the decision problems of clustering and computation offloading as a competitive game and propose an efficient algorithm.

3. System Model

In this section, we show the model for the distributed clustering and computation offloading game. In the model, a set of mobile devices expressed as $N = \{1, 2 \ldots, n\}$ are randomly distributed in a cell. These mobile devices have computation tasks, which must be completed with a limited delay. Each basestation is connected to a MEC server with computational capabilities, as shown in Figure 1. The computation tasks which cannot be accommodated by MEC servers will be forwarded to remote cloud datacenter, but the computation tasks offloaded to the datacenter may suffer from long latency. In this work, we consider a static scenario where the states of mobile devices do not change during the operation of computation offloading.

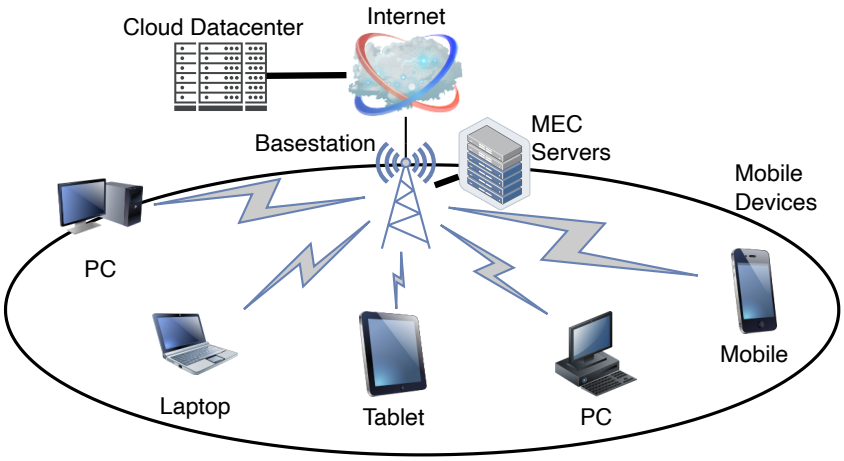

Figure 1. An illustration of mobile edge computing.

We further propose a clustering approach which incorporates short-distance communication, e.g., Wi-Fi in WLAN or dedicated short-range communications (DSRC) in VANET, to minimize communication overheads. As shown in the left of Figure 2, mobile devices are randomly distributed in a cell, where each device has a unique identifier. Within a range of short-distance communication, a mobile device can select a nearby MD as its cluster head, where the cluster heads are denoted by solid circles in the middle of Figure 2. The cluster heads collect data from their members of the same cluster and transmit the data to the basestation. As a result, the cluster members do not directly communicate with the basestation to reduce interference.

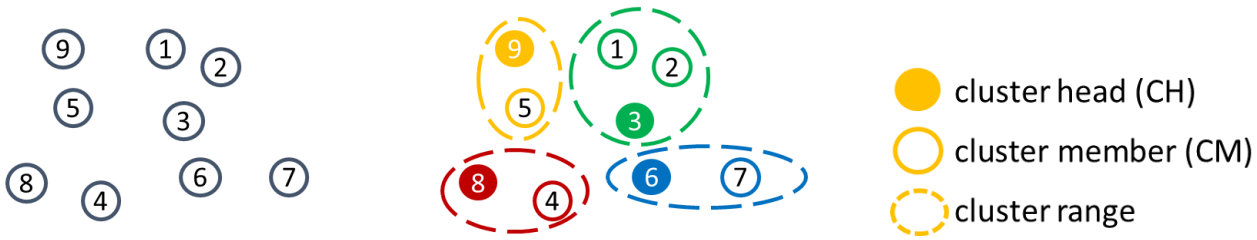

Figure 2. An example of MD clustering.

The node clustering generates a set of clusters denoted as $CL = \{cl_1, cl_2 \ldots, cl_t\}$, $1 \leq t \leq N$. With the clusters, a cluster head could achieve a superior transmission rate to the basestation and shorten the transmission latency. In this work, we neglect the

occurrence of collisions. If the collision ratios are too high to enable clustering, our scheme can still opt for direct communications between mobile devices and the basestation. We also assume that data from the computation task of each MD would be encrypted before the transmission to avoid the concern of privacy breaches.

In the following subsections, we describe the system model in detail. Section 3.1 describes the communication model. The computation model is introduced in Section 3.2. The notation used in both subsections is listed in Table 1.

Table 1. Notation.

Symbol	Definition
N	The set of mobile devices
CL	The set of clusters
M	The set of channels
OFF	The offloading decisions of all MDs
off_n	The offloading decision of MD n
C	The clustering decisions of all MDs
c_n	The clustering decision of MD n
J_n	The computation task of MD n
B_n	The data size for J_n
D_n	The required CPU cycles for J_n
T_n	The computation time for J_n
E_n	The energy consumption for J_n
p_n	The transmission power of MD n
r_n	The transmission rate of MD n

3.1. Communication Model

In this section, we describe the proposed communication model for mobile edge computing. Each basestation has a set of wireless channels represented as $M = \{1, 2 \ldots, m\}$. MDs share these channels by using OFDMA to assign a subset of channel resources. We denote $off_n \in \{0\} \cup M, c_n \in \{0\} \cup CL$ as the channel and clustering selection of each MD n, respectively. With the decision profiles, $OFF = \{off_1, off_2 \ldots, off_n\}$ and $C = \{c_1, c_2 \ldots, c_n\}$, of all mobile devices, we can compute the uplink data transmission rate of MD n which opts to offload via the MD clustering approach by using the following formula:

$$r_n(off, c) = Wlog_2(1 + \frac{p_n g_{n,s}}{\omega_0 + \sum_{m \in N: off_m = off_n, c_m \neq c_n, m \neq n} p_m g_{m,s}}) \quad (1)$$

W is the channel bandwidth, and p_n is the transmission power of MD n. $g_{n,s}$ denotes the channel gain between the MD n and the basestation s according to the path loss and shadowing. ω_0 denotes the background noise power.

From Equation (1), we can recognize that mobile devices may incur heavy interference when numerous devices select the same wireless channel. By clustering mobile devices to reduce excessive transmissions in a wireless channel, the uplink transmission rate can be improved.

3.2. Computation Model

Next, we present the computation model. We assume that each mobile device n has a computational task, $J_n = \{B_n, D_n\}$, to be executed either locally on MD n, or offloaded to MEC server based on its decision. B_n indicates the data size of computation input to offload, and D_n denotes the total number of CPU cycles required to accomplish J_n. Then,

we can calculate the computation overhead in terms of delay and energy consumption for local computing and edge computing.

3.2.1. Local Computing

Let f_n^l be the computational capability of MD n, i.e., the number of CPU cycles per second. If MD n calculates its task J_n locally, the delay is $T_n^l = \frac{D_n}{f_n^l}$, and the energy consumption can be formulated as:

$$E_n^l = D_n \varepsilon_n^l, \tag{2}$$

where ε_n^l is the coefficient of the consumed energy per CPU cycle.

3.2.2. Edge Computing

The main difference between edge and local computing is the additional overheads for the computation offloading of a MD. The overhead includes the offloading delay and energy consumption for transmitting data to the MEC server. The delay and energy consumption for the offloading are defined as T_n^{off} and E_n^{off}, where $E_n^{off} = T_n^{off} p_n$. After MD n finishes the transmission, the MEC server performs the computation task J_n remotely. Thus, the delay for executing task J_n is T_n^e. We ignore the energy consumption of the MEC server.

There are two approaches to offload the task of each MD, where a MD can either directly offload its task to the MEC server or upload its data through the cluster head. In the later case, the cluster head transmits the data of all cluster members based on the following equation:

$$B_n^{cl_n} = \sum_{n \in cl_n} B_n \tag{3}$$

Let f^e be the computational capability of the MEC server. If MD n offloads its task J_n to the MEC server through the basestation, the total delay and the energy consumption can be expressed as:

$$T_n^e = T_n^{off} + T_n^{exe} = \frac{B_n}{r_n(off, c)} + \frac{D_n}{f^e} \tag{4}$$

$$E_n^e = E_n^c = \frac{p_n B_n}{r_n(off, c)} \tag{5}$$

The possible values of B_n are listed in Equation (6), and those of p_n are listed in Equation (7).

$$B_n = \begin{cases} B_n, & \text{MD } n \text{ as an offloader} \\ B_n^{cl_n}, & \text{MD } n \text{ as a cluster head} \\ 0, & \text{MD } n \text{ as a cluster member} \end{cases} \tag{6}$$

$$p_n = \begin{cases} p_n^e, & \text{MD } n \text{ as an offloader or a cluster head} \\ p_n^w, & \text{MD } n \text{ as a cluster member} \end{cases} \tag{7}$$

We further combine Equations (2) and (5) to obtain computation overhead of clustering-based offloading for each MD:

$$E_n = \begin{cases} E_n^l, & \text{if } c_n = 0 \text{ and } off_n = 0 \\ E_n^e, & \text{if } c_n = 0 \text{ and } off_n > 0 \\ E_n^c, & \text{if } c_n > 0 \text{ and } off_n > 0 \end{cases} \tag{8}$$

We do not consider the downlink transmission of results in the model, because the result size is usually much smaller than the input data [19].

4. Problem Formulation

Next, we formulate the clustering and computation offloading problem for mobile edge computing based on the proposed communication and computation model.

4.1. Game Formulation

We denote $s_n = \{off_n, c_n\}$ as the channel and clustering decisions of the mobile device n. Let $s_{-n} = \{s_1, \ldots, s_{n-1}, s_{n+1}, \ldots, s_N\}$ be the decisions of the other MDs and $-s_n$ be decisions not chosen by mobile device n. Given the decisions of the other MDs, s_{-n}, mobile device n chooses the decision s_n as the solution of the following equation:

$$\begin{aligned} \text{OMIN} \quad &: \quad min \sum_{n \in N} E_n(s_n, s_{-n}) \\ C_1 \quad &: \quad off_n \in \{0, 1, 2 \ldots, M\}, \\ C_2 \quad &: \quad c_n \in \{0, cl_1, cl_2 \ldots, cl_t\}, \\ C_3 \quad &: \quad D_n(s_n) I_{\{off_n \neq 0, c_n \neq 0\}} \leq D_n(-s_n) I_{\{off_n \neq 0, c_n \neq 0\}}, \end{aligned} \tag{9}$$

where C_1 and C_2 are defined in Section 3.1. $I_{\{S\}}$ is used to indicate whether a decision variable is true. For example, if S is true, $I_{\{S\}} = 1$; otherwise, $I_{\{S\}} = 0$.

According to Equation (8), we can set the problem of computation energy consumption as the following equation:

$$E_n(s_n, s_{-n}) = \begin{cases} E_n^l, & \text{if } c_n = 0 \text{ and } off_n = 0 \\ E_n^e, & \text{if } c_n = 0 \text{ and } off_n > 0 \\ E_n^c, & \text{if } c_n > 0 \text{ and } off_n > 0 \end{cases} \tag{10}$$

Then, we formulate the overhead minimization problem (OMIN) to be a game of strategy, $\Gamma = (N, \{S_n\}_{n \in N}, \{E_n\}_{n \in N})$, where the players are the mobile devices $\in N$. S_n is the set of strategies including offloading off_n and clustering decision c_n for player n. The energy consumption $E_n(s_n, s_{-n})$ of each MD is the expense function to be minimized for player n. Consequently, we regard the game as the multi-MD clustering and computation offloading game and show the existence of a Nash equilibrium in the game.

Definition 1. *A set of strategies $s^\star = \{s_1^\star, \ldots, s_n^\star\}$ reaches a Nash equilibrium for the clustering and computation offloading game with multiple MDs where no MD can additionally reduce its energy consumption by adjusting its strategy in the equilibrium $s^\star$; i.e.,*

$$E_n(s_n^\star, s_{-n}^\star) < E_n(s_n, s_{-n}^\star), \forall s_n \in S_n, n \in N. \tag{11}$$

Based on the above definition, when the game of clustering and computation offloading reaches a Nash equilibrium, each MD has a mutually satisfactory solution.

4.2. Nash Equilibrium of the DCCO Game

Next, we show the presence of Nash equilibrium for the clustering and computation offloading (DCCO) game.

Definition 2. *It can be inferred from a potential game that a potential function $\Phi(s)$ is needed so that $\forall n \in N, s_n, s_n^\star \in S_n, s_{-n} \in S_{-n}$:*

$$\begin{aligned} & E_n(s_n^\star, s_{-n}) - E_n(s_n, s_{-n}) < 0 \\ \Longrightarrow\ & \Phi_n(s_n^\star, s_{-n}) - \Phi_n(s_n, s_{-n}) < 0. \end{aligned} \tag{12}$$

Lemma 1. *Given a set of strategies with offloading and clustering decisions $s = \{off, c\}$, edge computing with clustering is favorable if its received interference from other MDs, $\tau_n(s) = \tau_n(\text{off}, c) = \sum_{i \in N \setminus \{n\}: \text{off}_i = \text{off}_n, c_i \neq c_n} p_i g_{i,s}$, in the channel, satisfies that $\tau_n(s) \leq \Theta_n$, where the threshold is expressed as*

$$\Theta_n = \frac{p_n g_{n,s}}{\frac{p_n B_n}{2^{WE_n^l} - 1}} - \omega_0$$

Proof. According to Equations (2) and (5), if we anticipate the edge computing with clustering is advantageous compared with local computation for MD n, the following circumstance, $E_n^c(s) \leq E_n^l(s)$, must occur; i.e., $\frac{B_n}{r_n(s)} p_n \leq E_n^l$. Therefore, we can derive the following equation:

$$r_n(s) \geq \frac{B_n}{E_n^l} p_n.$$

Based on Equation (1), we can get that

$$\sum_{i \in N \setminus \{n\}: off_i = off_n, c_i \neq c_n} p_i g_{i,s} \leq \frac{p_n g_{n,s}}{\frac{p_n B_n}{2^{WE_n^l} - 1}} - \omega_0. \tag{13}$$

□

The condition after '\' indicates the exception cases.

Through Lemma 1, we notice that if a MD receives low interference in its channel, the MD will prefer to upload its task to the MEC server via its cluster head. In contrast, if the obtained interference from other MDs on the same channel is high, the MD should perform its task locally.

Theorem 1. *The clustering and computation offloading game is a possible game to reach a Nash equilibrium with a finite enhancement ability.*

Proof. First, we assume that $s = \{off, c\}$ are the channel and clustering decisions of mobile devices. The potential equation for clustering and computation offloading game can be presented as

$$\Phi(s) = \frac{1}{2} \sum_{i \in S} \sum_{j \in S \setminus \{i\}} p_i g_{i,s} p_j g_{j,s} I_{\{off_i = off_j, c_i \neq c_j\}} I_{\{c_i > 0\}} + \sum_{i \in S} p_i g_{i,s} \Theta_i I_{\{off_i = 0, c_i = 0\}} \tag{14}$$

□

Equation (14) can be expressed as the following:

$$\begin{aligned} \Phi(s) \quad = \quad & \frac{1}{2} \sum_{j \in S \setminus \{k\}} p_k g_{k,s} p_j g_{j,s} I_{\{off_j = off_k, c_j \neq c_k\}} I_{\{c_k > 0\}} \\ & + \frac{1}{2} \sum_{i \in S \setminus \{k\}} p_i g_{i,s} p_k g_{k,s} I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_i > 0\}} \\ & + \frac{1}{2} \sum_{i \in S \setminus \{k\}} \sum_{j \in S \setminus \{i,k\}} p_i g_{i,s} p_j g_{j,s} I_{\{off_i = off_j, c_i \neq c_j\}} I_{\{c_i > 0\}} \\ & + p_k g_{k,s} \Theta_k I_{\{off_k = 0, c_k = 0\}} \\ & + \sum_{i \in S \setminus \{k\}} p_i g_{i,s} \Theta_i I_{\{off_i = 0, c_i = 0\}} \end{aligned} \tag{15}$$

We can get the following equation:

$$\begin{aligned} & \sum_{j \in S \setminus \{k\}} p_k g_{k,s} p_j g_{j,s} I_{\{off_j = off_k, c_j \neq c_k\}} I_{\{c_k > 0\}} \\ = \quad & \sum_{i \in S \setminus \{k\}} p_i g_{i,s} p_k g_{k,s} I_{\{off_k = off_j, c_k \neq c_j\}} I_{\{off_i > 0, c_i > 0\}} \end{aligned} \tag{16}$$

Let $\Xi(s_{S \setminus \{k\}})$ be the following equation:

$$\frac{1}{2} \sum_{i \in S \setminus \{k\}} \sum_{j \in S \setminus \{i,k\}} p_i g_{i,s} p_j g_{j,s} I_{\{off_i = off_j, c_i \neq c_j\}} I_{\{c_i > 0\}} + \sum_{i \in S \setminus \{k\}} p_i g_{i,s} \Theta_i I_{\{off_i = 0, c_i = 0\}}. \tag{17}$$

With Equations (15) and (16), we can obtain the following equation:

$$\Phi(s) = \sum_{j \in S \setminus \{k\}} p_i g_{i,s} p_k g_{k,s} I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_i > 0\}} + p_k g_{k,s} \Theta_k I_{\{off_k = 0, c_k = 0\}} + \Xi(s_{S \setminus \{k\}}), \quad (18)$$

where $\Xi(s_{S \setminus \{k\}})$ is self-reliant for MD k's strategy s_k. With Equation (13), we obtain the following equation:

$$E_k(s) = \sum_{j \in S \setminus \{k\}} p_j g_{j,s} I_{\{off_j = off_k, c_j \neq c_k\}} I_{\{c_k > 0\}} + \Theta_k I_{\{off_k = 0, c_k = 0\}} \quad (19)$$

Then, we assume that MD k will have the situation $E_k(s_k^\star, s_{-k}) < E_k(s_k, s_{-k})$, when it prefers updating its clustering and offloading decisions to cut down its cost. With the potential game's definition, the request of updating would result in the situation $\Phi_k(s_k^\star, s_{-k}) < \Phi_k(s_k, s_{-k})$. We consider three cases—(1) $c_k > 0, c_k^\star > 0$, (2) $c_k = 0, c_k^\star > 0$ and (3) $c_k > 0, c_k^\star = 0$—to analyze whether the convergence can be achieved for a given channel decision off_k.

Case 1 appears when the MD k's clustering and offloading decisions are updated from the clustering $c_k > 0$ toward the other clustering $c_k^\star > 0$. Based on Equation (1), it is obvious that the function $Wlog_2\alpha$ consistently grows by α, and the circumstance $E_k(s_k^\star, s_{-k}) < E_k(s_k, s_{-k})$ is known. We can get the result of the following equation:

$$\sum_{i \in N \setminus \{k\}: off_i = off_k^\star, c_i \neq c_k} p_i g_{i,s} < \sum_{i \in N \setminus \{k\}: off_i = off_k, c_i \neq c_k} p_i g_{i,s} \quad (20)$$

With Equations (18) and (20), we can get:

$$\begin{aligned} \Phi_k(s_k, s_{-k}) - \Phi_k(s_k^\star, s_{-k}) &= p_k g_{k,s} \sum_{i \in S \setminus \{k\}} p_i g_{i,s} I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \\ &\quad - p_k g_{k,s} \sum_{i \in S \setminus \{k\}} p_i g_{i,s} I_{\{off_k^\star = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} > 0 \end{aligned} \quad (21)$$

Case 2 occurs when MD k's decision is updated from a clustering decision $c_k = 0$, i.e., no clustering, to another cluster $c_k^\star > 0$. The interference in the cluster $c_k^\star > 0$ should be smaller than the threshold of interference, i.e., $\sum_{i \in N \setminus \{n\}: off_i = off_k^\star, c_i \neq c_k} p_i g_{i,s} < \Theta_k$ and $E_k(s_k^\star, s_{-k}) < E_k(s_k, s_{-k})$. The following equation can be yielded:

$$\begin{aligned} \Phi_k(s_k, s_{-k}) - \Phi_k(s_k^\star, s_{-k}) &= p_k g_{k,s} \Theta_k I_{\{off_k = 0, c_k = 0\}} \\ &\quad - p_k g_{k,s} \sum_{i \in S \setminus \{k\}} p_i g_{i,s} I_{\{off_k^\star = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} > 0 \end{aligned} \quad (22)$$

The last case takes place when MD k's clustering and offloading decisions are updated with the cluster $c_k > 0$ toward the other clustering decision $c_k^\star = 0$, i.e., no clustering. As for $E_k(s_k^\star, s_{-k}) < E_k(s_k, s_{-k})$ and $\sum_{i \in N \setminus \{n\}: off_i = off_k^\star, c_i \neq c_k} p_i g_{i,s} > \Theta_k$, we can generate the following equation:

$$\begin{aligned} \Phi_k(s_k, s_{-k}) - \Phi_k(s_k^\star, s_{-k}) &= p_k g_{k,s} \sum_{i \in S \setminus \{k\}} p_i g_{i,s} I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \\ &\quad - p_k g_{k,s} \Theta_k I_{\{off_k^\star = 0, c_k = 0\}} > 0 \end{aligned} \quad (23)$$

Based on the analysis for three cases of updating decisions, we can observe that a potential game of clustering and computation offloading can be formed to reach Nash equilibrium.

5. Distributed Clustering and Computation Offloading Game

In this section, we develop an efficient distributed clustering and computation offloading game among multiple MDs to achieve Nash equilibrium and analyze the convergence of the DCCO game.

5.1. Algorithm

The proposed algorithm is conducted in an iterative manner, as listed in Algorithm 1. Initially, the decisions of all MDs are local computation and without joining any clustering; i.e., $off_n = 0, c_n = 0$. In order to examine the interference information of different channels, each MD, say n, may transmit its offloading and clustering decisions to the basestation through the channel access. Then, the basestation will return the information of wireless channels and the locations of nearby mobile devices to MD n. The MD n can thus measure the interference of different channels to avoid poor transmission quality among MDs of the same channel in the previous iteration. The locations of nearby MDs can be used to identify the neighboring nodes. Once a MD n receives the feedback from the basestation, it will calculate the best solution, $\Delta_n(t)$, based on the OMIN problem, where each MD n independently selects its decisions of offloading off_n and clustering c_n. If the decisions of current iteration are different to those of the previous one, i.e., $\Delta_n(t) \neq \Delta_n(t-1)$, the MD n transmits a request to the MEC server to compete for the permission to update its decision. Otherwise, the MD n does not forward any message to the MEC server to maintain its decision in this iteration; i.e., $off_n(t+1) = off_n(t), c_n(t+1) = c_n(t)$.

Algorithm 1 DCCO game.

Initialize:
offloading decision $off_n(0) = 0$;
clustering decision $c_n(0) = 0$.
while update decision request $\neq \emptyset$ **do**
  for each iteration t **do**
    for each MD n **do**
      Send a wireless signal toward basestation.
      Receive information of channels and near MDs.
      $\Delta_n(t) \longleftarrow$ compute best solution by (OMIN).
    end for
    if $\Delta_n(t) \neq \Delta_n(t-1)$ **then**
      Send a request to update decision to MEC server.
      if a permission of updating decision received from the MEC server **then**
        Update offloading and clustering decisions
        $off_n(t+1), c_n(t+1) = \Delta_n(t)$.
      else
        Keep offloading and clustering decisions
        $off_n(t+1) = off_n(t), c_n(t+1) = c_n(t)$.
      end if
    else
      Keep offloading and clustering decisions
      $off_n(t+1) = off_n(t), c_n(t+1) = c_n(t)$.
    end if
  end for
end while

When the MEC server receives update requests from MDs, the MEC server will randomly select and accept one of those requests and inform all MDs. Only the selected MD can update its decision, i.e., $off_n(t+1), c_n(t+1) = \Delta_n(t)$, and the other MDs keep their decisions unchanged as in the previous iteration—i.e., $off_n(t+1) = off_n(t), c_n(t+1) = c_n(t)$. The game will execute continuously until no update messages are transmitted to the MEC server. In other words, no mobile device can further reduce energy consumption by updat-

ing its decision. Therefore, the MEC server with the basestation will commit the decisions in the last iteration of the DCCO game, and each MD will perform its computational task based on its clustering and offloading decisions.

We note that it is possible that an MD may disconnect before convergence. In this case, the MD will no longer receive any updates to yield the final decision for computation offloading. The other MDs will ignore this MD and proceed to reach Nash equilibrium.

5.2. Convergence Analysis

Owing to Theorem 1, the algorithm of DCCO game will retain a Nash equilibrium within the limited iterations. In this subsection, we analyze the computational complexity of the DCCO algorithm.

Assume F iterations are required to finish the algorithm of the DCCO game for N mobile devices; the overall complexity of the DCCO game is $\mathcal{O}(FN \log N)$. We let $\Theta_{max} = max_{n \in N} \Theta_n$, $\Lambda_n = p_n g_{n,s}$, $\Lambda_{max} = max_{n \in N} \Lambda_n$ and $\Lambda_{min} = min_{n \in N} \Lambda_n$ and consider F iterations to reach convergence for the DCCO game. The following result can be obtained.

Theorem 2. *DCCO game will end within at most* $\frac{\Lambda_{max}^2}{2\Lambda_{min}} N^2 + \frac{\Theta_{max}\Lambda_{max}}{\Lambda_{min}} N$ *iterations when* Θ_n *and* Λ_n *are both positive integers,* $\forall n \in N$*. That is,* $F \leq \frac{\Lambda_{max}^2}{2\Lambda_{min}} N^2 + \frac{\Theta_{max}\Lambda_{max}}{\Lambda_{min}} N$.

Proof. Based on Equation (14), we can obtain:

$$0 \leq \Phi(s) \leq \frac{1}{2} \sum_{i \in N} \sum_{j \in N} \Lambda_{max}^2 + \sum_{i \in N} \Lambda_{max} \Theta_{max} = \frac{1}{2} \Lambda_{max}^2 N^2 + \Lambda_{max} \Theta_{max} N \tag{24}$$

□

We assume that a MD $k \in N$ receives permission from the MEC server to update its current clustering decision from c_k to another clustering decision $c_k^\star$, in order to yield a decline in its expense function. According to Definition 2, the above condition also yields a decline in the potential function as well; i.e.,

$$\Phi_k(s_k^\star, s_{-k}) > \Phi_k(s_k, s_{-k}) + \Lambda_{min}. \tag{25}$$

Now, we discuss each of the three cases presented in Section 4.2: (1) $c_k > 0, c_k^\star > 0$, (2) $c_k = 0, c_k^\star > 0$ and (3) $c_k > 0, c_k^\star = 0$.

Case 1: Based on Equation (21), we can obtain:

$$\begin{aligned} \Phi_k(s_k, s_{-k}) - \Phi_n(s_k^\star, s_{-k}) \quad = \quad & \Lambda_k \Big(\sum_{i \in S \setminus \{k\}} \Lambda_i I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \\ & - \sum_{i \in S \setminus \{k\}} \Lambda_i I_{\{off_k^\star = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \Big) > 0. \end{aligned} \tag{26}$$

Due to the property of integers as Λ_i for each MD in N,

$$\sum_{i \in S \setminus \{k\}} \Lambda_i I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \geq \sum_{i \in S \setminus \{k\}} \Lambda_i I_{\{off_k^\star = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} + 1 \tag{27}$$

As a result, based on Equation (26), we can yield the following equation for Case 1:

$$\Phi_k(s_k, s_{-k}) \geq \Phi_k(s_k^\star, s_{-k}) + \Lambda_k \geq \Phi_k(s_k^\star, s_{-k}) + \Lambda_{min}$$

Case 2: Based on Equation (22), we have the following equation:

$$\Phi_k(s_k, s_{-k}) - \Phi_n(s_k^\star, s_{-k}) = \Lambda_k \Big(\Theta_k - \sum_{i \in S \setminus \{k\}} \Lambda_i I_{\{off_k = off_i, c_k \neq c_i\}} I_{\{c_k > 0\}} \Big) > 0 \tag{28}$$

Thus, based on Equation (28), Case 2 can be held by:

$$\Phi_k(s_k, s_{-k}) \geq \Phi_k(s_k^\star, s_{-k}) + \Lambda_k \geq \Phi_k(s_k^\star, s_{-k}) + \Lambda_{min}$$

Case 3: Through the equivalent statements on Cases 1 and 2, we can demonstrate the following equation for the last case:

$$\Phi_k(s_k, s_{-k}) \geq \Phi_k(s_k^\star, s_{-k}) + \Lambda_{min}$$

Hence, according to Equations (24) and (25) and by applying the potential function into a minimum state, the algorithm of the DCCO game will finish within $\frac{\Lambda_{max}^2}{2\Lambda_{min}} N^2 + \frac{\Theta_{max}\Lambda_{max}}{\Lambda_{min}} N$ iterations, at most.

We consider that transmission power and channel gain of MD n in reality are both positive integers, i.e., $p_n \geq 0, g_{n,s} \geq 0$. Moreover, the condition of $\Theta_n \geq 0$ is non-negative to imply that the probability of MD n can achieve advantageous edge computing as compared with local computing. Our simulation results in Section 6 also demonstrate that the DCCO game can converge rapidly.

6. Simulation Results

In this section, we validate the effectiveness of the proposed algorithm based on simulations. We developed a Python-based simulator to perform our simulations. The simulation scenario included a basestation whose coverage was 100 m. There were 30 to 70 MDs randomly distributed in the cell. There were five channels, namely, $M = 5$; the bandwidth of each channel was 5 MHz. The transmission power p was 150 mWatts. We set the path loss factor as 4. The data generated by MDs were 5000 kbits, and the number of required CPU cycles for computation tasks was 1000×10^8. The computational capacity of the MEC server was 10×10^9 cycles/s, and that of a MD was 1.0×10^9 cycles/s. We summarize the simulation settings in Table 2.

Table 2. Simulation parameters.

Parameters	Value
Data size	5000 kbits
Number of CPU Cycles	1000×10^8 cycles
Number of Channels	5
Channel Bandwidth	5 MHz
Wi-Fi Channel Bandwidth	0.5 MHz
MEC Capacity	10×10^9 cycles/s
MD Capacity	1.0×10^9 cycles/s
MD Transmission Power	150 mWatts
MD Wi-Fi Power	50 mWatts
Noise	10^{-10} mWatts
Path Loss Factor	4
Coverage	50 m

In our simulation, two additional algorithms were used, as listed below.

1. Distributed computation offloading (DCO): The DCO algorithm exploits game theory for offloading decisions.
2. LE-based distributed computation offloading game (L&E DCO): The L&E DCO algorithm uses game theory based on latency and energy consumption [19].
3. Distributed clustering and computation offloading (DCCO): The proposed algorithm employs game theory for clustering and offloading decisions.

First, we reveal the numbers of offloading MDs for the numbers of MDs in the different schemes in Figure 3. The scheme L&E DCO considers the ratio between edge and local computing to outperform the algorithm based on original game theory (DCO). The results

show that it uses about 23% additional offloadings for 30 MDs as compared to DCO. However, our scheme, DCCO, can further increase the number of offloading MDs by 38%. It is apparent that DCCO outperforms both DCO and L&E DCO because of the additional MD clustering. The interference received by each MD is reduced, since the MDs in the same channel are eliminated. DCCO thus enables more MDs to offload tasks to the MEC server.

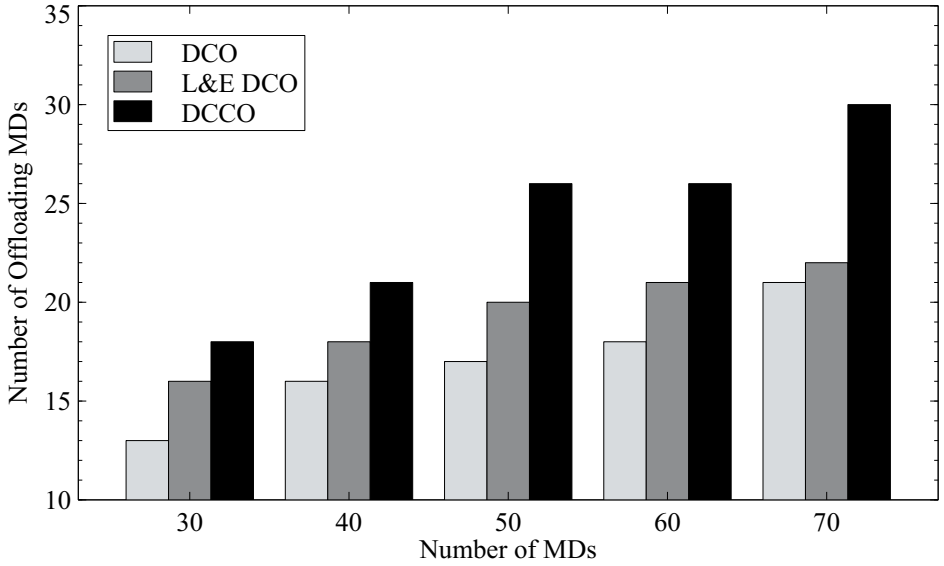

Figure 3. The numbers of offloading MDs for various algorithms.

We further show the numbers of cluster heads and members in Figure 4. The results show that the proposed algorithm can significantly increase the number of clustered MDs. When there are more MDs, more clusters are also generated to reduce the number of MDs directly communicating with the basestation.

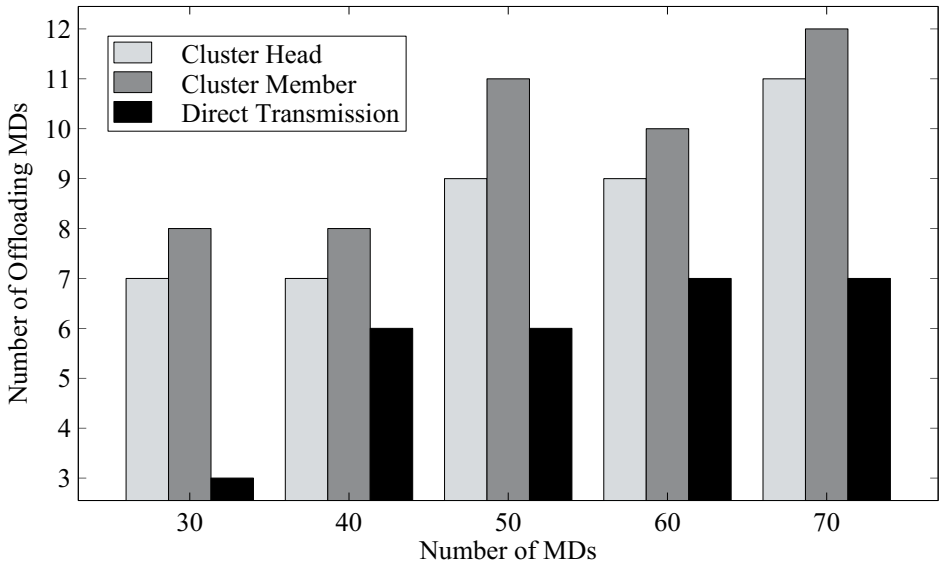

Figure 4. The numbers of cluster heads and members in each channel with 30 MDs.

We also show the number of MDs in each channel for the scenario with five channels and 30 MDs in Figure 5. The numbers of cluster heads and members for the proposed DCCO algorithm are also depicted. The results suggest that the additional offloading MDs can be achieved by MD clustering. Since the cluster members do not communicate with the basestation directly, the interference received by the mobile devices can be reduced to improve the transmission performance.

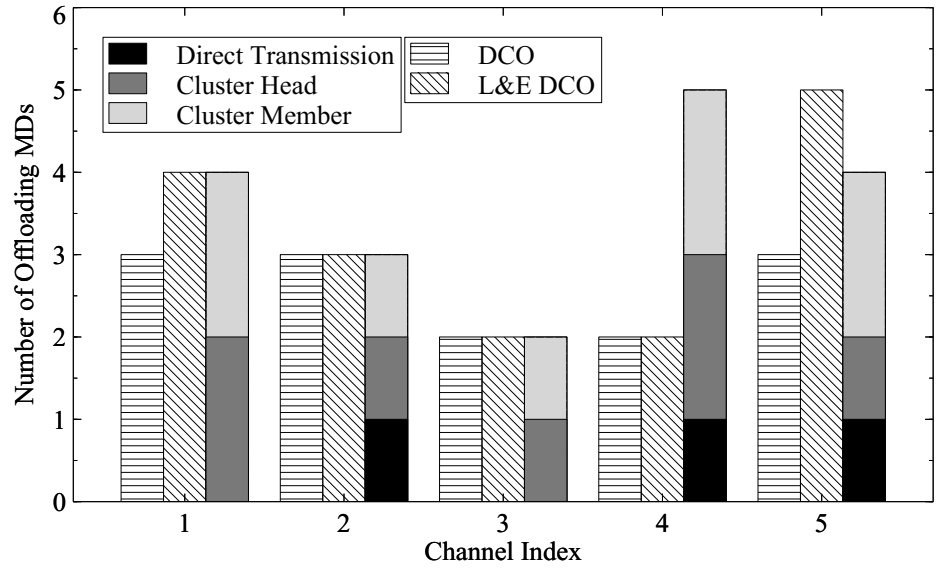

Figure 5. The distribution of 30 MDs in each channel.

Then, we indicate the energy usage in average for a mobile device to execute the task in Figure 6. The energy consumption includes that for both transmission and execution, where mesh bars denote the average energy consumption for data transmission. L&E DCO has similar energy consumption to DCO. In particular, L&E DCO consumes more energy for data transmission. As compared with the other two schemes, our scheme, DCCO, can reduce energy consumption by about 30% for 30 MDs by offloading more tasks to the MEC server.

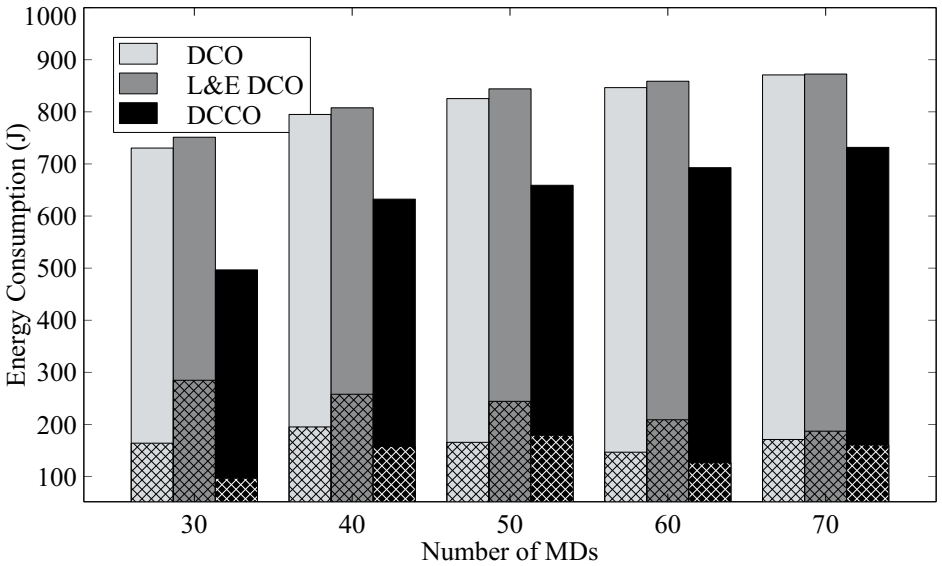

Figure 6. The average energy consumption with different algorithms.

Figure 7 shows the average delay for finishing tasks from mobile devices for different numbers of MDs. The delay in the results also includes the latency for both transmission and execution, where the mesh bars depict average transmission delay. In the case of 30 MDs, it can be observed that the proposed algorithm has the shortest delay. With the DCCO algorithm, we can reduce response delay by about 20% as compared to DCO.

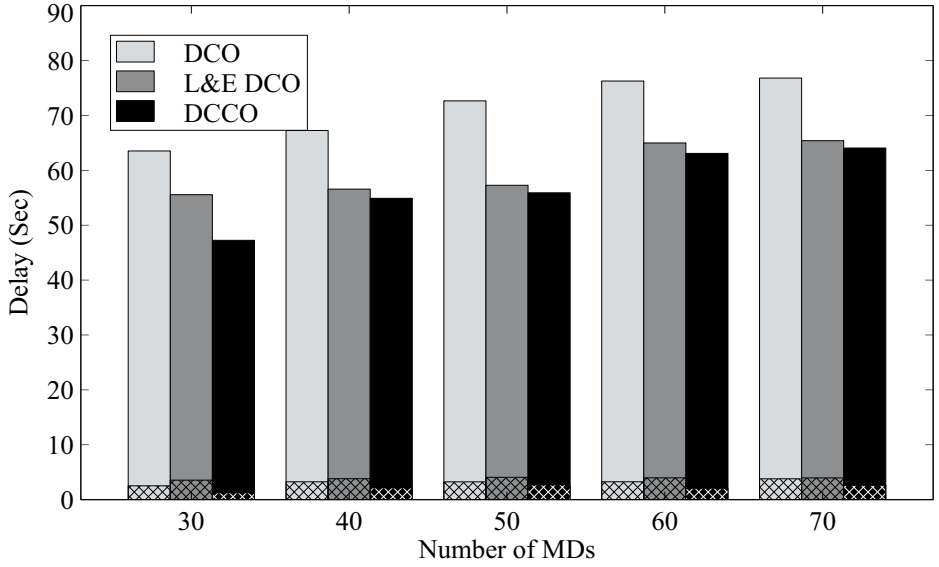

Figure 7. The average delay with different algorithms.

Figure 8 and 9 present the number of iterations for convergence with 30 and 70 MDs, respectively. In both figures, the upper half shows the energy consumption and the lower half shows the number of MD clusters. We observe that within 20 iterations, all schemes can reach Nash equilibrium. Although the proposed DCCO algorithm reaches the Nash equilibrium with more iterations than the other two algorithms, it also reduces the average energy consumption by increasing the number of clusters and offloading MDs. Figure 9 shows that about 35 iterations are required to reach Nash equilibrium. The results show that the convergence time is sublinearly related to the number of MDs. As a result, the proposed algorithm provides scalability for scenarios with numerous MDs.

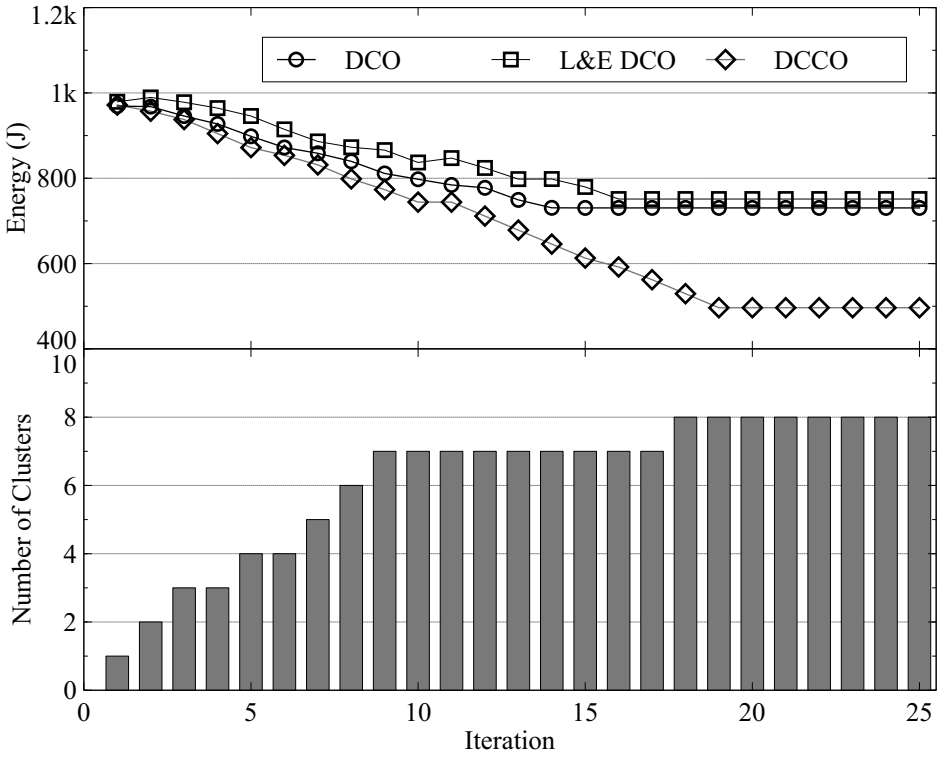

Figure 8. The convergence performance with 30 MDs.

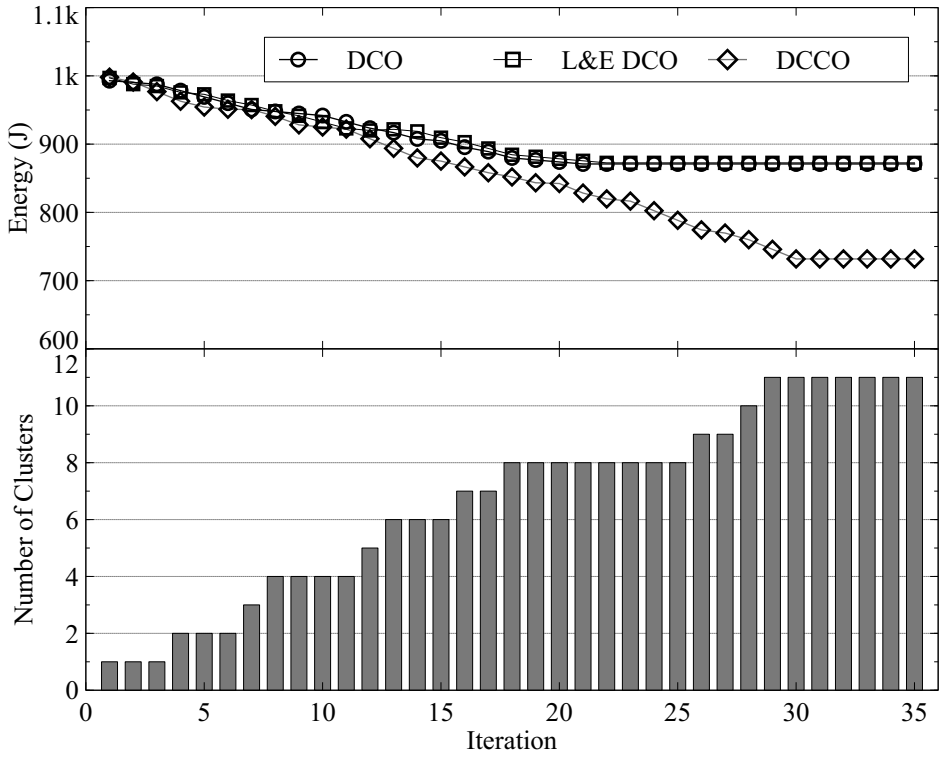

Figure 9. The convergence performance with 70 MDs.

7. Conclusions

In this work, we proposed an algorithm based on game theory to combine clustering and computation offloading to deal with increasing MDs in mobile edge computing. With MD clustering, the number of transmitting nodes in a channel can be reduced to improve the transmission rate because cluster members can forward data through their cluster heads. Accordingly, we formulated the overhead minimization problem as a competitive game and presented an algorithm for the clustering and computation offloading game. We also showed the existence of a Nash equilibrium for the game. In the performance evaluation, we showed that the proposed model for the distributed clustering and computation offloading game can achieve better efficiency of computation offloading than the previous game-theory-based schemes. With our algorithm, the number of offloaded tasks can be increased by up to 36% to lower the energy consumption of mobile devices by 30%. Our algorithm also shortens the latency of computation tasks by 20%. Moreover, our algorithm can effectively converge to yield feasible decisions of clustering and offloading. In our future work, we will attempt to improve the fairness of energy consumption among mobile devices, since mobile devices with poor channel quality may not be able to successfully offload their computation tasks.

Author Contributions: Conceptualization, Y.-Y.H. and P.-C.W.; methodology, Y.-Y.H.; software, Y.-Y.H.; validation, Y.-Y.H.; formal analysis, Y.-Y.H.; investigation, Y.-Y.H.; resources, P.-C.W.; data curation, Y.-Y.H.; writing—original draft preparation, Y.-Y.H.; writing—review and editing, P.-C.W.; visualization, Y.-Y.H.; supervision, Y.-Y.H.; project administration, P.-C.W.; funding acquisition, P.-C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Council, grant number NSTC 111-2221-E-005-045.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Al-Habob, A.A.; Dobre, O.A.; Armada, A.G.; Muhaidat, S. Task scheduling for mobile edge computing using genetic algorithm and conflict graphs. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8805–8819. [CrossRef]
2. Maray, M.; Shuja, J. Computation offloading in mobile cloud computing and mobile edge computing: Survey, taxonomy, and open issues. *Mob. Inf. Syst.* **2022**, *2022*, 1121822. [CrossRef]
3. Tran, T.X.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Commun. Mag.* **2017**, *55*, 54–61. [CrossRef]
4. Cardellini, V.; Personé, V.D.N.; Di Valerio, V.; Facchinei, F.; Grassi, V.; Presti, F.L.; Piccialli, V. A game-theoretic approach to computation offloading in mobile cloud computing. *Math. Program.* **2016**, *157*, 421–449. [CrossRef]
5. Alahmadi, H.; Boabdullah, F. A Review of Multi-Channel Medium Access Control Protocols for Wireless Sensor Networks. *Eur. J. Eng. Technol. Res.* **2021**, *6*, 39–53. [CrossRef]
6. Hu, H.C.; Wang, P.C. Computation Offloading Game for Multi-Channel Wireless Sensor Networks. *Sensors* **2022**, *22*, 8718. [CrossRef]
7. You, C.; Huang, K.; Chae, H.; Kim, B.H. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wirel. Commun.* **2016**, *16*, 1397–1411. [CrossRef]
8. Malik, U.M.; Javed, M.A.; Frnda, J.; Rozhon, J.; Khan, W.U. Efficient Matching-Based Parallel Task Offloading in IoT Networks. *Sensors* **2022**, *22*, 6906. [CrossRef]
9. Guan, X.; Lv, T.; Lin, Z.; Huang, P.; Zeng, J. D2D-Assisted Multi-User Cooperative Partial Offloading in MEC Based on Deep Reinforcement Learning. *Sensors* **2022**, *22*, 7004. [CrossRef]
10. Zaman, S.K.u.; Jehangiri, A.I.; Maqsood, T.; Haq, N.u.; Umar, A.I.; Shuja, J.; Ahmad, Z.; Dhaou, I.B.; Alsharekh, M.F. LiMPO: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Clust. Comput.* **2022**, 1–19.
11. Zaman, S.K.u.; Jehangiri, A.I.; Maqsood, T.; Umar, A.I.; Khan, M.A.; Jhanjhi, N.Z.; Shorfuzzaman, M.; Masud, M. COME-UP: Computation Offloading in Mobile Edge Computing with LSTM Based User Direction Prediction. *Appl. Sci.* **2022**, *12*, 3312. [CrossRef]
12. Zhang, J.; Hu, X.; Ning, Z.; Ngai, E.C.H.; Zhou, L.; Wei, J.; Cheng, J.; Hu, B. Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. *IEEE Internet Things J.* **2017**, *5*, 2633–2645. [CrossRef]
13. Tran, T.X.; Pompili, D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Trans. Veh. Technol.* **2018**, *68*, 856–868. [CrossRef]
14. Dinh, T.Q.; Tang, J.; La, Q.D.; Quek, T.Q. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Trans. Commun.* **2017**, *65*, 3571–3584.
15. Bi, S.; Zhang, Y.J. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4177–4190. [CrossRef]
16. Neto, J.L.D.; Yu, S.Y.; Macedo, D.F.; Nogueira, J.M.S.; Langar, R.; Secci, S. ULOOF: A user level online offloading framework for mobile edge computing. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2660–2674. [CrossRef]
17. Chen, L.; Zhou, S.; Xu, J. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1619–1632. [CrossRef]
18. Mazouzi, H.; Achir, N.; Boussetta, K. Dm2-ecop: An efficient computation offloading policy for multi-user multi-cloudlet mobile edge computing environment. *ACM Trans. Internet Technol.* **2019**, *19*, 1–24. [CrossRef]
19. Yang, L.; Zhang, H.; Li, X.; Ji, H.; Leung, V.C. A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2762–2773. [CrossRef]
20. Guo, H.; Liu, J. Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4514–4526. [CrossRef]
21. He, Q.; Cui, G.; Zhang, X.; Chen, F.; Deng, S.; Jin, H.; Li, Y.; Yang, Y. A game-theoretical approach for user allocation in edge computing environment. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *31*, 515–529. [CrossRef]
22. Miettinen, A.P.; Nurminen, J.K. Energy efficiency of mobile clients in cloud computing. *HotCloud* **2010**, *10*, 19.
23. Li, S.; Lin, S.; Cai, L.; Li, W.; Zhu, G. Joint Resource Allocation and Computation Offloading With Time-Varying Fading Channel in Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3384–3398. [CrossRef]
24. Ning, Z.; Wang, X.; Rodrigues, J.J.; Xia, F. Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3058–3067. [CrossRef]
25. Alsenwi, M.; Tun, Y.K.; Pandey, S.R.; Ei, N.N.; Hong, C.S. UAV-Assisted Multi-Access Edge Computing System: An Energy-Efficient Resource Management Framework. In Proceedings of the 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 7–10 January 2020; pp. 214–219.
26. Tun, Y.K.; Alsenwi, M.; Pandey, S.R.; Zaw, C.W.; Hong, C.S. Energy Efficient Multi-Tenant Resource Slicing in Virtualized Multi-Access Edge Computing. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–4.

27. Cheng, K.; Teng, Y.; Sun, W.; Liu, A.; Wang, X. Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
28. Garima; Gulati, H.; Singh, P. Clustering techniques in data mining: A comparison. In Proceedings of the 2015 2nd international conference on computing for sustainable global development (INDIACom), New Delhi, India, 11–13 March 2015; pp. 410–415.
29. Kumar, Y.; Singh, P.K. Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. *Appl. Intell.* **2018**, *48*, 2681–2697. [CrossRef]
30. Kumar, Y.; Singh, P.K. A chaotic teaching learning based optimization algorithm for clustering problems. *Appl. Intell.* **2019**, *49*, 1036–1062. [CrossRef]
31. Hong, Z.; Wang, R.; Song, T.; Shao, Q.; Zhou, L. Energy-efficient and power-optimal topology control with potential game for heterogeneous wireless sensor networks. In Proceedings of the 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), Atlanta, GA, USA, 8–10 October 2016; pp. 533–540.
32. Attiah, A.; Chatterjee, M.; Zou, C.C. A game theoretic approach for energy-efficient clustering in wireless sensor networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
33. Afsar, M.M.; Crump, R.T.; Far, B.H. Energy-efficient coalition formation in sensor networks: A game-theoretic approach. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–6.
34. Loomba, R.; de Frein, R.; Jennings, B. Selecting energy efficient cluster-head trajectories for collaborative mobile sensing. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.
35. Bouet, M.; Conan, V. Mobile edge computing resources optimization: A geo-clustering approach. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 787–796. [CrossRef]
36. Du, H.; Zeng, S.; Dou, T.; Fang, W.; Wang, Y.; Zhang, C. FASTBEE: A Fast and Self-Adaptive Clustering Algorithm Towards to Edge Computing. In Proceedings of the 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Shanghai, China, 22–24 June 2018; pp. 128–133.
37. He, S.; Wang, T.; Wang, S. Mobility-driven user-centric AP clustering in mobile edge computing based ultra dense networks. *Digit. Commun. Netw.* **2019**, *6*, 210–216. [CrossRef]