Computation Offloading Game for Multi-Channel Wireless Sensor Networks

Heng-Cheng Hu and Pi-Chung Wang *

Abstract: Computation offloading for wireless sensor devices is critical to improve energy efficiency and maintain service delay requirements. However, simultaneous offloadings may cause high interferences to decrease the upload rate and cause additional transmission delay. It is thus intuitive to distribute wireless sensor devices in different channels, but the problem of multi-channel computation offloading is NP-hard. In order to solve this problem efficiently, we formulate the computation offloading decision problem as a decision-making game. Then, we apply the game theory to address the problem of allowing wireless sensor devices to make offloading decisions based on their own interests. In the game theory, not only are the data size of wireless sensor devices and their computation capability considered but the channel gain of each wireless sensor device is also included to improve the transmission rate. The consideration could evenly distribute wireless sensor devices to different channels. We prove that the proposed offloading game is a potential game, where the Nash equilibrium exists in each game after all device states converge. Finally, we extensively evaluate the performance of the proposed algorithm based on simulations. The simulation results demonstrate that our algorithm can reduce the number of iterations to achieve Nash equilibrium by 16%. Moreover, it improves the utilization of each channel to effectively increase the number of successful offloadings and lower the energy consumption of wireless sensor devices.

Keywords: wireless sensor devices; computation offloading; game theory; channel gain; Nash equilibrium

1. Introduction

Wireless sensor devices (WSDs) have emerged in various applications of smart cities, remote healthcare, unmanned aerial vehicle (UAV) and smart homes [1–5], where WSDs generate and transmit remote sensing data to sink nodes. Various types of modern sensor devices may generate a great volume of data [6,7]. For example, mobile devices capable of sensing and computing may act as sensors to participate in crowdsensing for collecting video, images, or voice [8]. These WSDs share data and extract information for operations of common interest. While both energy and computation capabilities of WSDs are usually limited, some environmental monitoring tasks may require low response latency [9,10]. Although the computation capabilities of WSDs are increasing, the energy consumption of WSDs is still problematic for computation-intensive tasks. Therefore, effectively performing computation-intensive tasks is an important challenge for WSDs.

To overcome this challenge, edge computing is a promising and effective approach [11,12]. With edge computing, WSDs could offload their computation tasks which are locally computed originally to remote edge servers with higher computation capabilities through wireless networks. Therefore, the offloading could reduce the energy consumption of WSDs without increasing processing delay.

Although edge computing could increase the efficiency of processing computation tasks for WSDs, the rapidly increasing WSDs may cause radio communication interference to result in longer communication delays when these WSDs offload their tasks to the edge nodes.
server simultaneously. Currently, multi-channel communication has been applied to several applications of WSDs to enhance the performance of both transmission throughput and energy efficiency [2]. The communication interference can thus be avoided by allocating one channel to each WSD. Unfortunately, this approach is not feasible because the spectrum resources are scarce and expensive. An efficient computation offload policy based on limited spectrum resources is thus desired.

In this paper, we develop an efficient solution for the computational offloading problem based on a multi-channel computation offloading game. Game theory has been widely used to design a decentralized mechanism for decision-making problems. By considering each WSD as a selfish agent, each agent can decide to either perform the computation locally or by the remote edge server according to their own interests. The previous proposals based on game theory [13–16] only consider computation capabilities and data size for the decision-making. Because of the selfish behavior in game theory, the dynamic channel assignment may result in poor overall performance. For example, when the channel gain of a WSD is particularly large as compared to the other WSDs, the WSD will occupy a channel. In order to avoid this situation, our scheme considers the channel resource allocation problem based on the channel gain of each WSD for the decision-making problem of computation offloading.

The rest of the paper is organized as follows. Section 2 describes related work. In Section 3, we introduce the system model of multi-WSD offloading. Then, we formulate the proposed centralized mechanism and decentralized mechanism for the multi-WSD computation offload problem in Section 4. In Section 5, we introduce the proposed algorithm based on game theory and prove the existence of Nash equilibrium. In Section 6, the results of the simulation are presented according to the proposed mechanism. Finally, the conclusion of this paper is provided in Section 7.

2. Related Works

Nowadays, the research on the computation offloading decision problem of multiple wireless devices can be divided into two types, namely centralized and decentralized computation offloading. The mechanisms of centralized computation offloading request all devices to send their statistics data before offloading to edge servers, where the statistics data includes data size, computing capacity, and received signal strength indicator (RSSI). After receiving the data, the edge server yields the optimal solution of resource allocation. A previous work presented a system model of multiple basestations with built-in edge servers to serve wireless devices [17]. This work also considers communication interference and presents an approach based on the genetic algorithm to solve the decision-making problem of energy-efficient offloading. The computation tasks could be divided into subtasks for parallel execution [18]. It is also possible to employ neighboring devices for cooperative partial offloading [19].

In the case that a single device transmits multiple independent computation requests to an edge server, another work determines the order of offloading tasks according to the delay and energy requirements [20]. Zhang et al. proposed a mechanism of energy-efficient offloading for multiple devices [21]. With the decision-making results, the maximum energy consumption of the overall system can be minimized under the time limit of devices. The tradeoff between delay and energy consumption can also be addressed by an iterative search algorithm [22]. This algorithm yields the optimal solution in multi-device environments. Kan et al. formulated the multi-device resource-allocation offloading decision problem and proposed a heuristic algorithm to solve the cost minimization problem [23]. With the technique of energy harvesting, the computation offloading problem for wireless-powered WSD networks is also investigated [24–29].

The decentralized offloading mechanisms allow each device to make decisions in a distributed manner. Each device can thus decide the most appropriate decision based on its own interests. The offloading decision of each device will affect the decisions of other devices and vice versa. Game theory is one of the most commonly used methods to
solve decentralized decision problems. Chen et al. considered the scenario that a single basestation serves multiple devices and proposed a mechanism based on game theory to solve the problem of efficient computation offloading for edge cloud [13]. They also prove the existence of Nash equilibrium. Guo et al. proposed a scenario that multiple devices offload to multiple edge servers [14]. They studied the collaborative computation offloading problem among edge servers. To support the idea of centralized cloud and multi-access edge computing, Guo and Liu proposed a general architecture that devices can choose not only local or edge server but also cloud data centers for computation offloading [15]. Meskar et al. designed a competition game in order to reduce the energy consumption of a device for computation tasks [16]. In this competition game, devices choose the decision with the minimum energy consumption under time constraints according to the decisions of other devices. Yuan et al. extended edge computing to UAV and proposed a Stackelberg game approach for the heterogenous computation offloading decision problem [5]. Recently, techniques of machine learning have been employed for the offloading decision problem [4,30,31].

In the previously mentioned algorithms of decentralized offloading based on game theory, devices make decisions based on their benefits without considering overall system performance. The selfishness may give some devices poor transmission bandwidth and degrade the overall performance of computation offloading. To avoid such situations, we consider the transmission interferences among devices of simultaneously offloading computation tasks and develop a system to achieve balanced offloading performance for all devices.

3. System Model

We present the system model of this work in Figure 1. We consider that a set of \( n \) WSDs, \( N = \{ \text{wsd}_1, \text{wsd}_2, \ldots, \text{wsd}_n \} \), are distributed in an area, where each WSD needs to complete a computationally intensive task within a limited time period. These WSDs are connected to one basestation through multiple wireless channels. In addition, each basestation is connected to an edge server, which is connected to a power outlet. The edge server has much higher computation capability than WSDs and allows WSDs to offload their tasks. Similar to the previous studies on offloading of cloud computing and edge computing [13–17,20–23,32–38], we consider a static scenario, where the states of WSDs do not change during the computation offloading.

![Figure 1. An Illustration of Computation Offloading for WSDs in a Multichannel Wireless Sensor Network.](image)

3.1. Communication Model

Next, the communication model of edge computing is introduced. Each basestation has a set of \( m \) wireless channels denoted as \( M = \{1, 2, 3, \ldots, m\} \). Because each channel
may suffer from wireless interferences, we use OFDMA to share spectrum resources to multiple WSDs [10]. We make the channel resources of the system orthogonal to each other, but the number of channel resources is not enough to allocate one channel for each WSD. We express \(0 \leq a_i \leq |M|\) as the computation offloading decision of \(wsd_i\). Specifically, if \(a_i > 0\), it means that \(wsd_i\) decides to offload its task through wireless channel \(a_i\); if \(a_i = 0\), it means that \(wsd_i\) chooses to compute its task locally. Based on the decision variables \(A = (a_1, a_2, \ldots, a_n)\) of each WSD, we can calculate the uplink data rate \(r_i\) of \(wsd_i\) that decides to offload the task to the edge server as follows [39].

\[
r_i = W \log_2 \left(1 + \frac{p_i h_{i,s}}{\omega_0 + \sum_{j \in N: a_j = a_i} p_j h_{j,s}}\right),
\]

where \(W\) is the channel bandwidth, and \(p_i\) is the transmission power of \(wsd_i\); \(h_{i,s}\) denotes the channel gain between \(wsd_i\) and the basestation based on the path loss and shadowing, \(\omega_0\) denotes the background noise power.

3.2. Computation Model

Then, we introduce the computation model for both local and edge computing. We assume that each \(wsd_i\) has a task \(J_i = (B_i, D_i)\) to be calculated. Each task can be calculated locally or offloaded to an edge server according to the WSD's decision. \(B_i\) indicates the data size of the task \(J_i\) and \(D_i\) denotes the number of CPU cycles required to complete task \(J_i\). Then, we show the processing time and energy consumption required for local computing and edge computing.

**Local Computing**

If \(wsd_i\) decides to calculate task \(J_i\) locally, the delay and energy consumption are described below. Let \(f_l^i\) be the local computation capability (i.e., CPU cycles per second) of \(wsd_i\). The delay time required for local computing is defined as

\[
T_l^i = \frac{D_i}{f_l^i},
\]

The energy consumption of \(wsd_i\) for the computation is given as

\[
E_l^i = \gamma_i D_i,
\]

where \(\gamma_i\) is the consumed energy per CPU cycle. In order to ensure the completion of computation tasks, we assume that local computing can always meet the delay requirements of computation tasks with higher energy consumption.

**Edge Computing**

We further consider the delay and energy consumption if \(wsd_i\) chooses to offload its task to the edge server through the basestation. The difference from the local calculation is that the computation offloading requires extra time and energy consumption for transmitting the input data of the task to the edge server. We calculate the transmission time of the offloading input data by using the following equation:

\[
T_{off}^i = \frac{B_i}{r_i}.
\]

The transmission energy of \(wsd_i\) is expressed as

\[
E_{off}^i = T_{off}^i p_i = \frac{B_i}{r_i} p_i.
\]

After the transmission, the edge server performs the computation task \(J_i\). Let \(f_m^i\) be the computational capability (i.e., CPU cycles per second) of the edge server allocated for \(wsd_i\).
Here, we assume that edge servers always have sufficient resources to fulfill the computing requirements of all WSDs. The delay time required by \( wsd_i \) to calculate on the edge server is expressed as

\[
T^e_i = \frac{D_i}{f_{im}}
\]  

(6)

We do not consider the energy consumption of edge servers, because these servers are wire-powered. Thus, the time delay and energy consumption for the edge computation can be expressed as

\[
T^m_i = T^off_i + T^c_i = B_i + \frac{D_i}{f_{im}}
\]  

(7)

and

\[
E^m_i = E^off_i = \frac{B_i}{r_i} p_i
\]  

(8)

According to Equations (3) and (8), we can obtain the energy consumption for each \( wsd_i \) as

\[
E_i = \begin{cases} 
E^l_i, & \text{if } a_i = 0 \\
E^m_i, & \text{if } a_i > 0
\end{cases}
\]  

(9)

Similar to other works [13–16,40–43], we ignore the time of transmitting the results from the edge server to WSDs because the results are usually much smaller than the input data.

Edge computing can reduce the energy consumption of WSDs by offloading computation tasks, but not all WSDs can offload their tasks because of poor channel quality. According to Equation (1), it is necessary to lower channel interference to achieve a higher transmission rate. However, the channel interference is mainly tied to the WSDs sharing the same channel. In order to consider the overall transmission performance, we consider the channel gain of each WSD to reduce channel interferences. Moreover, because a centralized approach may suffer from high system overhead and considerable latency owing to acquiring information from all WSDs, we employ a decentralized approach for the computation offloading decision problem.

3.3. Channel Gain

Our main idea is to make WSDs with similar channel gains offload their tasks through the same wireless channel. It is based on the observation that mixing both low-channel-gain and high-channel-gain WSDs in the same channel may result in huge transmission interference to keep these WSDs from successful offloading. Accordingly, we can categorize WSDs into different sets according to their channel gains.

Since we assume that there are \( m \) wireless channels and \( n \) WSDs, we can use the k-means to divide WSDs into \( m \) clusters according to their channel gains. The cluster centers are denoted as \( \{ h^1_{\text{channel}}, h^2_{\text{channel}}, h^3_{\text{channel}}, \ldots, h^m_{\text{channel}} \} \). They are also equivalent to the channel’s qualities. We use the difference between each WSD’s channel gain and each channel’s quality by calculating the distance from the cluster center based on the following equation:

\[
\arg \min_{h^j_{\text{channel}}} |h_{i,s} - h^j_{\text{channel}}|
\]  

(10)

Therefore, we can define the difference between \( wsd_i \) and channel \( j \)’s quality as

\[
diff_i(j) = |h_{i,s} - h^j_{\text{channel}}|
\]  

(11)

According to the above formula, WSDs with similar channel gains will choose the same channel to offload their tasks.
4. Problem Formulation

In this section, we separately consider the problem of centralized computing offloading and decentralized computing offloading.

4.1. Centralized Edge Computing

In a centralized edge-computing system, we can formulate the energy optimization problem as follows:

\[
\begin{align*}
\min & \sum_{1 \leq i \leq n} E_i, \\
\text{s.t.} & \ a_i \in \{0, 1, 2, 3, \ldots, M\}, \\
& \ T(a_i) \leq T_{i, \text{max}}.
\end{align*}
\]

\(T(a_i)\) is the time spent for \(wsd_i\) if the WSD chooses the channel \(a_i\) for task offloading. \(T_{i, \text{max}}\) denotes the maximum allowable delay for \(wsd_i\).

WSDs transmit their statistics to the edge server to make system-wide decisions in a centralized manner. Although the centralized optimization can minimize overall energy consumption, it is problematic to enforce WSDs to comply with the remote decisions.

4.2. Decentralized Edge Computing

In a decentralized edge-computing system, we can formulate the energy optimization problem for each WSD in the following equation:

\[
\begin{align*}
\min & \ E_i(a_i, a_{-i}), \\
\text{s.t.} & \ a_i \in \{0, 1, 2, 3, \ldots, M\}, \\
& \ T(a_i) \leq T_{i, \text{max}},
\end{align*}
\]

where \(a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)\) be the set of decisions of the other WSDs except \(wsd_i\).

In this system, all WSDs make decisions according to their own interests. Therefore, the decision of each WSD could be selfish for the whole system. Each WSD considers the current decision state of other WSDs before making the best decision for themselves. In this paper, we consider the decision-making problem of distributed computation offloading among the WSDs and present a feasible algorithm.

5. Multi-Channel Computation Offloading Game

In this section, we implement the multi-channel computation offloading game for mobile edge computing.

5.1. Game Formulation

Let \(-a_i\) be the other decisions that can meet time constraints but are not chosen by \(wsd_i\). Then, given \(a_{-i}\), \(wsd_i\) would like to set its decision variable \(a_i\) as the solution of the following equation:

\[
\begin{align*}
\min & E_i(a_i, a_{-i}), \\
\text{subject to} & \ a_i \in \{0, 1, 2, 3, \ldots, M\}, \\
& \ T(a_i) \leq T_{i, \text{max}}, \\
& \ \text{diff}(a_i) I_{\{a_i \neq 0\}} \leq \text{diff}(-a_i) I_{\{-a_i \neq 0\}}.
\end{align*}
\]

\(I_{\{X\}}\) is an indicator function, where \(X\) is a conditional expression. If \(X\) is true, \(I_{\{X\}}\) is equal to 1; otherwise \(I_{\{X\}}\) is equal to 0.
In order to ease the explanation of the problem, we can set the energy consumption function of the problem as the following equation:

\[ E_i(a_i, a_{-i}) = \begin{cases} 
E^l_i, & \text{if } a_i = 0, \\
E^m_i, & \text{if } a_i > 0 \text{ and } T_i \leq T^\text{max}_i, \\
\infty, & \text{if } a_i > 0 \text{ and } T_i > T^\text{max}_i.
\]  

Then, we formulate the optimization problem in Equation (14) as a strategic game \( \Gamma = (N, \{A_i\}_{1 \leq i \leq n}, \{E_i\}_{1 \leq i \leq n}) \), where the set of wsd \( i \) is the set of the rational players, \( A_i \) is the strategy set of player \( i \), and the energy consumption function \( E_i(a_i, a_{-i}) \) of wsd \( i \) is the cost function to be minimized by player \( i \). The game \( \Gamma \) is the multi-channel computation offloading game. In the following, we introduce the concept of Nash equilibrium.

**Definition 1.** A strategy set \( a^* = (a^*_i, \ldots, a^*_n) \) is a Nash equilibrium for the channel selection computation offloading game when no WSD can reduce its cost by changing its decision, i.e.,

\[ E_i(a^*_i, a^*_{-i}) < E_i(a_i, a^*_{-i}), \forall a_i \in A_i, 1 \leq i \leq n. \]  

Therefore, according to the definition of Nash equilibrium, we can know that when the game reaches equilibrium, each WSD is in the situation of choosing their own best decision and the game has reached its end.

**5.2. Game Theory with WSD’s Channel Gain**

The pseudo-code of the proposed algorithm is listed in Algorithm 1. Initially, this algorithm assumes that the offloading decisions of all WSDs are local computation, i.e., \( a_i = 0 \), for all WSDs. Moreover, each WSD will transmit a pilot signal to the basestation first. After the basestation receives pilot signals from all WSDs, the basestation returns all the received interference information to all WSDs. After each WSD receives the interference information, each device computes the best solution for their DOPT problems in multiple iterations. If an offloading decision of a WSD remains the same, the WSD will not update its decision to the edge server. If the new decision is different from the previous one, the WSD will send an update request message to the edge server to compete for validating its updated decision. This means that for wsd \( i \), the cost of updating the decision will be lower than that of maintaining the decision of the previous iteration, where the cost is the energy consumption of wsd \( i \).

Then, after the edge server receives the update requests from WSDs, it will randomly select one of the WSDs that transmit update requests. The update request of the selected WSD is validated and a reply of update permission is returned to the WSD. The WSDs whose decisions are not selected by the edge server will not receive the reply message from the edge server and will retain the same decision as the previous iteration. In other words, these WSDs will not update their decisions.

After the selected device updates its decision, the next iteration is performed to generate new decisions by repeating the above actions until no WSD requests for updating their decisions. Namely, the decisions of all WSDs reach the Nash equilibrium. Reaching the Nash Equilibrium means that no WSD can obtain a lower cost by changing its decision. Therefore, all the WSD decisions in this iteration form the solution to our multi-channel computation offloading problem.


Algorithm 1 Game Theory with WSD’s Channel Gain

Initialization: The initial computation decisions for all WSDs are \( a_i(0) = 0 \)
1: repeat for each WSD, wsd, and each decision time slot \( t \) in parallel:
2: Send a pilot signal to the basestation on the selected channel \( a_i(t) \)
3: Receive the interference information on all channel from the basestation
4: \( \Delta_i(t) \leftarrow \text{compute the best response solution for (DOPT)} \)
5: if \( \Delta_i(t) \neq a_i(t-1) \) then
6: Send an update request message to the edge server to compete for an update decision
7: if there is an update permission message received from the edge server then
8: choose the decision \( a_i(t+1) = \Delta_i(t) \) for next time slot
9: else choose the original decision \( a_i(t+1) = a_i(t) \) for next time slot
10: end if
11: else choose the original decision \( a_i(t+1) = a_i(t) \) for next time slot
12: end if
13: until no WSD transmits any update to the edge server

5.3. Convergence Analysis

Next, we discuss the convergence of the algorithm, which means that we have to prove the existence of Nash equilibrium for the multi-channel computation offloading game. Before that, we first introduce a convenient property that is helpful for proving the existence of Nash equilibrium.

Definition 2. A game is said to be a potential game if there is a potential function \( \Phi(a) \) in the game such that \( \forall wsd_i \in N, a_{-i} \in A_{-i}, a_i, a'_i \in A_i, \)
\[
E_i(a'_i, a_{-i}) - E_i(a_i, a_{-i}) < 0
\]
\[
\Rightarrow \Phi(a'_i, a_{-i}) - \Phi(a_i, a_{-i}) < 0. \tag{17}
\]

The potential function is a useful tool for analyzing the equilibrium property of a game, because one of the properties of a potential game is that it must have a Nash equilibrium and the finite improvement property [43]. Next, we show that the multi-channel computation offloading game is a potential game.

Lemma 1. Given a computation offloading strategy set \( a \), the edge server is beneficial for wsd, if the interference \( \mu_i(a) = \sum_{wsd_j \in N \setminus \{wsd_i\}, a_j = a} p_j h_{ij,s} \) received on the selected wireless channel \( \mu_i > 0 \) satisfies that \( \mu_i(a) \leq Q_i \), with the threshold
\[
Q_i = \frac{p_i h_{i,s}}{B_i} - \bar{\omega}_0.
\]

Proof. According to Equations (3) and (5), if we want the edge server to be beneficial as compared to local computing for wsd, we have the condition \( E_i^m(a) \leq E_i^l \), i.e., \( \frac{B_i}{r_i p_i} \leq E_i \). Therefore, we can derive the following equation,
\[
r_i \geq \frac{B_i}{E_i p_i}.
\]
According to Equation (1), we can obtain that
\[ \sum_{w_{sd} \in N \setminus \{w_{sd}\}, a_j = a_i} p_i h_{i,s} \leq \frac{p_i h_{i,s}}{p_i B_i} - \bar{\omega}_0 \cdot \frac{2 W E_i}{1}. \]

From Lemma 1, we observe that if the interference received by WSD on the channel is low enough, it is beneficial for the WSD to offload the work to the edge server for computing. Conversely, when the interference on the channel is too high for the WSD, the device should perform local computation.

**Theorem 1.** The channel selection computation offloading game is a potential game that must have Nash Equilibrium and the finite improvement property.

**Proof.** We first construct the potential equation for the channel selection computation offloading game as
\[
\Phi(a) = \frac{1}{2} \sum_{i \in A} \sum_{j \in A \setminus \{i\}} p_i h_i p_j h_j I_{\{a_i = a_j\} I_{\{a_i > 0\}}} + p_i h_i T_i I_{\{a_i = 0\}}.
\] (18)

Then, Equation (18) can be equivalently written as the following formula:
\[
\frac{1}{2} \sum_{j \in A \setminus \{k\}} p_k h_k p_j h_j I_{\{a_j = a_k\} I_{\{a_k > 0\}}} + \frac{1}{2} \sum_{i \in A \setminus \{k\}} p_i h_i p_k h_k I_{\{a_k = a_j\} I_{\{a_k > 0\}}} + \frac{1}{2} \sum_{i \in A \setminus \{k\}} \sum_{j \in A \setminus \{i,k\}} p_i h_i p_j h_j I_{\{a_j = a_k\} I_{\{a_k > 0\}}} + p_k h_k Q_k I_{\{a_k = 0\}} + \sum_{i \in A \setminus \{k\}} p_i h_i Q_i I_{\{a_j = 0\}}.
\] (19)

Then, because
\[
\frac{1}{2} \sum_{i \in A \setminus \{k\}} \sum_{j \in A \setminus \{i,k\}} p_i h_i p_j h_j I_{\{a_j = a_k\} I_{\{a_k > 0\}}} + \sum_{i \in A \setminus \{k\}} p_i h_i Q_i I_{\{a_j = 0\}}
\] (20)
is independent of \(w_{sd_k}\)'s strategies \(a_k\) and
\[
\sum_{j \in A \setminus \{k\}} p_k h_k p_j h_j I_{\{a_j = a_k\} I_{\{a_k > 0\}}} = \sum_{i \in A \setminus \{k\}} p_i h_i p_k h_k I_{\{a_k = a_j\} I_{\{a_k > 0\}}}.
\] (21)

we can use Equations (19)–(21) to derive the following result:
\[
\Phi(a) = \sum_{j \in A \setminus \{k\}} p_j h_j p_k h_k I_{\{a_j = a_k\} I_{\{a_k > 0\}}} + p_k h_k Q_k I_{\{a_k = 0\}} + \Xi(a_{A \setminus \{k\}}),
\] (22)
where $\Xi(a_{A_i}\{k\})$ is the Equation (20). Since $wsd_k$ update decision $a_k$ cannot change $\Xi(a_{A_i}\{k\})$, we can omit it in the rest of the proof.

Next, we assume that when $wsd_k$ decides to update its decision to reduce its cost function, it will make $E_k(a_k', a_{-k}) < E_i(a_k, a_{-k})$. According to the definition of potential game, the update should also cause $\Phi(a_k', a_{-k}) < \Phi(a_i, a_{-i})$ in the potential function. In order to prove the case, we consider three cases, namely case (1): $a_k > 0, a_k' > 0$, case (2): $a_k = 0, a_k' > 0$, and case (3): $a_k > 0, a_k' = 0$.

The first case occurs when $wsd_k$'s decision is updated from the wireless channel $a_k > 0$ to the wireless channel $a_k' > 0$. According to Equation (1), because the function $w\log_2 x$ is monotonically increasing for $x$ and the condition $E_k(a_k', a_{-k}) < E_i(a_k, a_{-k})$ is known, we can obtain the result of this inequality:

$$\sum_{i \in \mathcal{N}\backslash\{k\}; a_i = a_k'} p_i h_i < \sum_{i \in \mathcal{N}\backslash\{k\}; a_i = a_k} p_i h_i.$$  \hfill (23)

Next, we can know the following result according to Equations (22) and (23), that is,

$$\Phi(a_k, a_{-k}) - \Phi(a_k', a_{-k}) = p_k h_k \sum_{i \in \mathcal{A}\backslash\{k\}} p_i h_i I_{\{a_i = a_i \}} I_{\{a_i > 0\}}$$
$$- p_k h_k \sum_{i \in \mathcal{A}\backslash\{k\}} p_i h_i I_{\{a_i' = a_i \}} I_{\{a_i > 0\}} > 0.$$  \hfill (24)

In the second case, it means that the decision of $wsd_k$ is updated from a decision of local computation, $a_k = 0$, to edge computing using a wireless channel $a_k' > 0$. We know that if $wsd_k$ selects a wireless channel $a_k' > 0$ for offloading, the interference on the wireless channel $a_k'$ must be lower than the threshold value of interference, i.e., $\sum_{i \in \mathcal{N}\backslash\{wsd_k\}; a_i = a_k'} p_i h_i < Q_k$ and $E_k(a_k', a_{-k}) < E_i(a_k, a_{-k})$. Accordingly, we can derive the following equation:

$$\Phi(a_k, a_{-k}) - \Phi(a_k', a_{-k}) = p_k h_k Q_k I_{\{a_k = 0\}}$$
$$- p_k h_k \sum_{i \in \mathcal{A}\backslash\{k\}} p_i h_i I_{\{a_i' = a_i \}} I_{\{a_i > 0\}} > 0.$$  \hfill (25)

In the third case, the decision of $wsd_k$ is updated from offloading with the wireless channel $a_k > 0$ to local computation, $a_k' = 0$. This case could happen under two situations. One is $E_k(a_k', a_{-k}) < E_i(a_k, a_{-k})$, the other is $T_k(a_k, a_{-k}) > T^\alpha_k$. For the former situation, because $\sum_{i \in \mathcal{N}\backslash\{wsd_k\}; a_i = a_k'} p_i h_i > Q_k$, we can obtain the following equation:

$$\Phi(a_k, a_{-k}) - \Phi(a_k', a_{-k}) = p_k h_k \sum_{i \in \mathcal{A}\backslash\{k\}} p_i h_i I_{\{a_i = a_i \}} I_{\{a_i > 0\}}$$
$$- p_k h_k Q_k I_{\{a_k = 0\}} > 0.$$  \hfill (26)

For the latter situation, we can derive that $E_i(a_k, a_{-k}) \equiv \infty$ according to Equation (15). Thus, we can also conclude that $E_k(a_k', a_{-k}) < E_i(a_k, a_{-k})$ and $\sum_{i \in \mathcal{N}\backslash\{wsd_k\}; a_i = a_k'} p_i h_i \equiv \infty > Q_k$. This also implies that there will be the same result.

Based on the results of the above three possible updates of decision, we can demonstrate that the multi-channel computation offloading game is a potential game that can reach Nash Equilibrium. □
6. Simulation Results

In this section, we show the simulation results of the proposed decentralized multi-channel computing offloading algorithm. Section 6.1 describes the simulation parameters and Section 6.2 presents the simulation results.

6.1. Simulation Parameters

In our scenario, we consider a basestation whose coverage is 100 m [14]. There are [30–50] WSDs, namely N = [30,50], randomly distributed within the coverage [13]. There are five available channels, namely \( M = 5 \) [13], where the bandwidth of channel, \( W \), is 10MHz [44]. The transmission power \( p \) is 100 mW [13] and the background noise is \( \omega_0 = -100 \text{ dbm} \) [13]. The channel gain is defined as \( h_{i,s} = l_{i,s}^{-\alpha} \), where \( l_{i,s} \) is the distance from \( wsd_i \) to the basestation. We set the path loss factor as \( \alpha = 4 \) [13].

The data generated by WSDs could be of various types and sizes. We randomly generate data size, \( B = [100, 1000] \times 10^3 \text{ bits} \) [23]. The number of required CPU cycles for computation tasks is also randomly selected, where \( D = [100, 1000] \times 10^6 \text{ cycles} \) [23]. The computation capacity of a WSD, \( f^i \), is randomly set within \( [1.5, 2.5] \times 10^9 \text{ cycles/sec} \) [23] and the consumed energy per CPU cycle is \( \gamma = 0.5 \text{ J/gigacycle} \) [17]. The edge computation capacity \( f^m \) is assigned to each WSD that computes on the edge server as \( 10 \times 10^9 \text{ cycles/sec} \) [13]. The maximum tolerance time \( T_{\text{max}}^i \) for \( wsd_i \) is \([0.7, 1]\) sec.

6.2. Simulation Results

In order to compare the performance of our algorithm, we also implement several mechanisms as listed below.

Local: All WSDs compute their tasks locally.

Random: The offloading decisions of all WSDs are random.

Greedy: A WSD with a larger difference between edge and local computation has a higher priority for computation offloading.

Game Theory (GT): This approach uses game theory for decision-making without considering channel gains of WSDs.

Game Theory with Channel Gain (GTCC): GTCC minimizes the energy consumption of WSDs by performing the game theory, where the communication quality of each channel is considered. (\(-a_i\) be the other decisions that can meet time constraints but not chosen by \( wsd_i \).)

Channel Pre-Allocation+Game Theory (CPA+GT): CPA+GT assigns channels for WSDs based on their channel gains initially. Then, it minimizes energy consumption by applying the game theory. (\(-a_i\) be the other decisions that are not chosen by \( wsd_i \).)

We first show the effectiveness of the proposed approach of channel selection with 40 WSDs and five channels. Figure 2 shows the data rate of each WSD in descending order of their signal strengths. The data size of each WSD is \( 500 \times 10^3 \text{ bits} \). We can observe that as compared to random channel selection, the selection based on channel quality can improve the data rate of most WSDs. Moreover, the approach can avoid a low data rate because there are only WSDs of similar channel gains in the same channel. By considering channel quality, the transmission rate of most WSDs can be improved to increase the number of successful offloadings.
Figure 2. Average Data Rate for WSDs with and without the Proposed Channel Selection Approach.

The number of offloadings for different numbers of WSDs is shown in Figure 3. The figure shows the difference among different approaches based on game theory with and without considering channel gain. The figure shows that if there are more channel resources for WSDs, there are more opportunities for GTCG to offload tasks to the edge server because the WSDs of GTCG can change their decisions among the channels. However, when the channel resources are scarce, the channels that GTCG can select are reduced for the WSDs with lower channel-gain values to degrade the number of offloadings. CPA+GT also does not have the problem that WSDs rob channels from WSDs with lower channel gain by changing the channel selection. It can maintain transmission performance even when the channel resources are reduced. The performance of the random decision could outperform GT because GT could allow a WSD to occupy one single channel. If the approach of the random decision could evenly allocate WSDs in different channels, the random-decision approach could achieve better performance with respect to the number of offloadings.

Figure 3. Average Number of Offloadings for Different Number of WSDs.

Next, we show the overall energy consumption of WSDs in Figure 4. By offloading computation tasks to edge servers, the energy consumption of WSDs can be improved. Since GTCG and CPA+GT can offload more tasks to the edge server as shown in Figure 3, they consume less energy as compared to the other approaches.
Figure 4. Average Energy Consumption for Different Number of WSDs.

Figure 5 shows the average latency for completing tasks. We note that all tasks can be accomplished within their time constraints since each task can always be computed locally. Both GTCG and CPA+GT have longer latency because each offloading requires additional transmission latency. Therefore, the average latency could be decreased with more local-computation tasks. The latency of CPA+GT may decrease with more WSDs because the percentage of WSDs with successful offloadings decreases. The difference between GTCG/CPA+GT and the other approaches is about 111 milliseconds or less.

Figure 6 shows the number of offloadings with different channel bandwidths. As the per-channel bandwidth increases, GTCG can always have a greater number of offloaded tasks as compared to CPA+GT because the additional resources cannot significantly affect the decisions of CPA+GT. Instead, GTCG allows WSDs to change their channel selections. Since the average available resources for each WSD increase, when there is a WSD to occupy another channel, the impact on the WSDs of the occupied channel would be less significant. As for other methods that do not consider channel gain, although the number has increased, it is still far from GTCG.
Figure 6. Average Number of Offloadings with Different Bandwidth.

In Figure 7, we increase the number of channels to show the average number of offloadings. Likewise, when the available resources increase, GTCG can always offload more tasks. In fact, GTCG could offload all tasks with eight channels. On the contrary, when the number of channels for CPA+GT is increased to more than 10, it is still impossible to achieve offloading for all WSDs. GTCG provides the best offloading performance among all compared algorithms.

Figure 7. Average Number of Offloadings with Different Number of Channels

We further demonstrate the problem of a single WSD occupying a wireless channel, as mentioned earlier. Figure 8 shows the number of channels selected by the WSD in each approach. Among them, we can see that random decision, greedy and GT have one channel occupied by only one WSD to cause unfair channel allocation. The imbalance channel allocation for WSDs can be avoided by considering the channel quality.
In addition to showing the number of WSDs in each channel, we also present the standard deviation for the number of WSDs in each channel for different approaches in Table 1. Both GTCG and CPA+GT have smaller values of standard deviation as compared to the other approaches. We can thus conclude that the decisions of channel selection from the proposed approaches are better balanced. CPA+GT has a smaller value of standard deviation than GTCG because it does not allow each WSD to change their channel selections.

Table 1. Standard Deviation for the Number of WSDs in Different Channels.

| Method     | Standard Deviation |
|------------|--------------------|
| Random     | 11.02              |
| Greedy     | 12.16              |
| GT         | 11.67              |
| CPA+GT     | 2.56               |
| GTCG       | 3.53               |

We illustrate the difference in WSD fairness between GTCG/CPA+GT and other methods by performing 100 computation tasks and show the percentage of successful offloading for each WSD in descending order of their channel gains in Figure 9. The methods without considering the channel gains of WSDs have imbalanced offloading ratios among WSDs. Moreover, the WSDs with better channel gains may not successfully offload their tasks. However, it is not the case for CTCG. When a WSD cannot acquire sufficient resources on the original channel, it may still offload its task by changing its channel selection. Although GTCG/CPA+GT cannot keep WSDs with different channel gains having the same offloading ratios, they do increase the offloading ratios for WSDs with low channel gains.
Next, we consider the case of unevenly distributed WSD locations, where 25% of WSDs are close to the basestation, and 75% of the WSDs are located at the border of the basestation coverage. Figure 10 shows that the performance of GTCG and CPA+GT is not affected by the uneven WSD distribution. In addition, GTCG always has better performance than CPA+GT because the channel pre-allocation of CPA+GT may limit the offloading decisions of some WSDs.

Figure 11 presents an iterative process of three methods based on game theory. We observe that GTCG and CPA+GT reach Nash equilibrium within 32 and 37 iterations, respectively. The GT takes 44 iterations to reach convergence with severe oscillation of energy consumption. The oscillation occurs with some WSD and with better quality occupies a channel to make the original WSDs of the same channel suffer from high transmission interferences. These WSDs may change their channels to result in low convergence. CPA+GT converges faster than GTCG because CPA+GT assigns channels for each WSD initially. As a result, each update of offloading decision only affects the WSDs in the same channel. Although GTCG has a slightly slower convergence performance than CPA+GT, it achieves better offloading performance as a reasonable tradeoff.
7. Conclusions

In this paper, we consider the efficient computation offloading problem of multiple WSDs. As the number of WSDs increases, we consider the computation offloading problem in a multi-channel wireless sensor network for better transmission performance. However, the transmission performance of WSDs in the same channel is mainly affected by the WSDs with high channel gains. Accordingly, we propose a channel-assignment approach for WSDs. Our approach can arrange WSDs with similar channel gains in the same channel to effectively improve transmission performance. Then, we formulate a decentralized computational offloading decision problem and propose an algorithm based on game theory. We further prove the existence of Nash equilibrium for our algorithm. The simulation results show that our algorithm can effectively increase the number of offloadings by jointly considering the channel gain of each WSD for channel selection to minimize the energy consumption of WSDs. Our algorithm also has fast convergence performance. Our future work attempts to further increase the number of successful offloadings by employing heterogenous transmission techniques.

Author Contributions: Conceptualization, H.-C.H. and P.-C.W.; methodology, H.-C.H.; software, H.-C.H.; validation, P.-C.W.; formal analysis, H.-C.H.; investigation, H.-C.H.; resources, P.-C.W.; data curation, H.-C.H.; writing—original draft preparation, H.-C.H.; writing—review and editing, P.-C.W.; visualization, H.-C.H.; supervision, H.-C.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, Taiwan, grant number 111-2221-E-005-045.

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. Shahhosseini, S.; Anzanpour, A.; Azimi, I.; Labbaf, S.; Seo, D.; Lim, S.S.; Liljeberg, P.; Dutt, N.; Rahmani, A.M. Exploring computation offloading in IoT systems. *Inf. Syst.* 2022, 107, 101860. [CrossRef]
2. 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]
3. Soyata, T.; Muraleedharan, R.; Funai, C.; Kwon, M.; Heinzelman, W. Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In Proceedings of the IEEE symposium on computers and communications (ISCC), Cappadocia, Turkey, 1–4 July 2012; pp. 59–66.
4. Chen, X.; Liu, G. Federated Deep Reinforcement Learning-Based Task Offloading and Resource Allocation for Smart Cities in a Mobile Edge Network. *Sensors* 2022, 22, 4738. [CrossRef] [PubMed]
5. Yuan, X.; Xie, Z.; Tan, X. Computation Offloading in UAV-Enabled Edge Computing: A Stackelberg Game Approach. Sensors 2022, 22, 3854. [CrossRef]
6. Kang, J.; Eom, D.S. Offloading and transmission strategies for IoT edge devices and networks. Sensors 2019, 19, 835. [CrossRef]
7. Zhou, S.; Le, D.V.; Tan, R.; Yang, J.Q.; Ho, D. Configuration-Adaptive Wireless Visual Sensing System with Deep Reinforcement Learning. IEEE Trans. Mob. Comput. 2022, accepted. [CrossRef]
8. Zhang, C.; Zhao, M.; Zhu, L.; Wu, T.; Liu, X. Enabling Efficient and Strong Privacy-Preserving Truth Discovery in Mobile Crowdsensing. IEEE Trans. Inf. Forensics Secur. 2022, 17, 3569–3591. [CrossRef]
9. Chalhoub, G.; Deux, M.C.; Rmili, B.; Misson, M. Multi-channel wireless sensor network for Heavy-Lift Launch Vehicles. Acta Astronaut. 2019, 158, 68–75. [CrossRef]
10. Chen, L.; Yao, M.; Wu, Y.; Wu, J. EECDN: Energy-Efficient Cooperative DNN Edge Inference in Wireless Sensor Networks. ACM Trans. Internet Technol. 2022, Accepted. [CrossRef]
11. 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]
12. Kumar, K.; Lu, Y.H. Cloud computing for mobile users: Can offloading computation save energy? Computer 2010, 43, 51–56. [CrossRef]
13. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient multi-user computation offloading for multi-edge cloud computing. IEEE/ACM Trans. Netw. 2015, 24, 2795–2808. [CrossRef]
14. Guo, H.; Liu, J.; Zhang, J. Efficient computation offloading for multi-access edge computing in 5G HetNets. In Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
15. Guo, H.; Liu, J. Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks. IEEE Trans. Veh. Technol. 2018, 67, 4514–4526. [CrossRef]
16. Meskar, E.; Todd, T.D.; Zhao, D.; Karakostas, G. Energy efficient offloading for competing users on a shared communication channel. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 3192–3197.
17. Guo, F.; Zhang, H.; Ji, H.; Li, X.; Leung, V.C. Energy Efficient Computation Offloading for Multi-Access MEC Enabled Small Cell Networks. In Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018.
18. Malik, U.M.; Javed, M.A.; Frdna, J.; Rozhon, J.; Khan, W.U. Efficient Matching-Based Parallel Task Offloading in IoT Networks. Sensors 2022, 22, 6906. [CrossRef]
19. 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]
20. Mao, Y.; Zhang, J.; Letaief, K.B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
21. Zhang, K.; Mao, Y.; Leng, S.; Zhao, Q.; Li, L.; Peng, X.; Pan, L.; Maharjan, S.; Zhang, Y. Energy-efficiency for offloading mobile edge computing in 5G heterogeneous networks. IEEE Access 2016, 4, 5896–5907. [CrossRef]
22. Zhang, J.; Hu, X.; Ning, Z.; Nai, E.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]
23. Kan, T.Y.; Chiang, Y.; Wei, H.Y. Task offloading and resource allocation in mobile-edge computing system. In Proceedings of the 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–4.
24. Wang, L.; Shao, H.; Li, J.; Wen, X.; Lu, Z. Optimal Multi-User Computation Offloading Strategy for Wireless Powered Sensor Networks. IEEE Access 2020, 8, 35150–35160. [CrossRef]
25. Huang, L.; Bi, S.; Zhang, Y.J.A. Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks. IEEE Trans. Mob. Comput. 2020, 19, 2581–2593. [CrossRef]
26. Wang, C.; Lu, W.; Peng, S.; Qu, Y.; Wang, G.; Yu, S. Modeling on Energy Efficiency Computation Offloading Using Probabilistic Action Generating. IEEE Internet Things J. 2022, 9, 20681–20692. [CrossRef]
27. Garcia, C.E.; Camana, M.R.; Koo, I. Particle Swarm Optimization-Based Secure Computation Efficiency Maximization in a Power Beacon-Assisted Wireless-Powered Mobile Edge Computing NOMA System. Energies 2020, 13, 5540. [CrossRef]
28. Tarchi, D.; Bozorgchenani, A.; Gebremeskel, M.D. Zero-Energy Computation Offloading with Simultaneous Wireless Information and Power Transfer for Two-Hop 6G Fog Networks. Energies 2022, 15, 1632. [CrossRef]
29. Park, L.; Lee, C.; Na, W.; Choi, S.; Cho, S. Two-Stage Computation Offloading Scheduling Algorithm for Energy-Harvesting Mobile Edge Computing. Energies 2019, 12, 4367. [CrossRef]
30. Mekala, M.S.; Jolfaei, A.; Srivastava, G.; Zheng, X.; Anvari-Moghaddam, A.; Viswanathan, P. Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems. IEEE Trans. Emerg. Top. Comput. Intell. 2020, 6, 245–254. [CrossRef]
31. Yang, S.; Lee, G.; Huang, L. Deep Learning-Based Dynamic Computation Task Offloading for Mobile Edge Computing Networks. Sensors 2022, 22, 4088. [CrossRef] [PubMed]
32. Barbarossa, S.; Sardellitti, S.; Di Lorenzo, P. Joint allocation of computation and communication resources in multiuser mobile cloud computing. In Proceedings of the IEEE 14th workshop on signal processing advances in wireless communications (SPAWC), Darmstadt, Germany, 16–19 June 2013; pp. 26–30.
33. Barbera, M.V.; Kosta, S.; Mei, A.; Stefa, J. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Turin, Italy, 14–19 April 2013; pp. 1285–1293.
34. Wen, Y.; Zhang, W.; Luo, H. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Orlando, FL, USA, 25–30 March 2012; pp. 2716–2720.
35. Wu, H.; Huang, D.; Bouzefrane, S. Making offloading decisions resistant to network unavailability for mobile cloud collaboration. In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, 20–23 October 2013; pp. 168–177.
36. Chen, X. Decentralized computation offloading game for mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. 2014, 26, 974–983. [CrossRef]
37. Wu, S.L.; Tseng, Y.C.; Lin, C.Y.; Sheu, J.P. A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. Comput. J. 2002, 45, 101–110. [CrossRef]
38. Iosifidis, G.; Gao, L.; Huang, J.; Tassiulas, L. An iterative double auction for mobile data offloading. In Proceedings of the 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), Tsukuba Science City, Japan, 13–17 May 2013; pp. 154–161.
39. Rudenko, A.; Reiher, P.; Popek, G.J.; Kuenning, G.H. Saving portable computer battery power through remote process execution. ACM SIGMOBILE Mob. Comput. Commun. Rev. 1998, 2, 19–26. [CrossRef]
40. Huerta-Canepa, G.; Lee, D. An adaptable application offloading scheme based on application behavior. In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications-Workshops, Okinawa, Japan, 25–28 March 2008; pp. 387–392.
41. Xian, C.; Lu, Y.H.; Li, Z. Adaptive computation offloading for energy conservation on battery-powered systems. In Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS), Hsinchu, Taiwan, 5–7 December 2007; pp. 1–8.
42. Huang, D.; Wang, P.; Niyato, D. A dynamic offloading algorithm for mobile computing. IEEE Trans. Wirel. Commun. 2012, 11, 1991–1995. [CrossRef]
43. Monderer, D.; Shapley, L.S. Potential games. Games Econ. Behav. 1996, 14, 124–143. [CrossRef]
44. 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]