Article

Mobility-Aware Offloading Decision for Multi-Access Edge Computing in 5G Networks

Saeid Jahandar [1] , **Lida Kouhalvandi [2]** , **Ibraheem Shayea [1,*]** , **Mustafa Ergen [1]** , **Marwan Hadri Azmi [3]** and **Hafizal Mohamad [4]**

1   Electronics and Communication Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical University (ITU), Istanbul 34467, Turkey; bonab18@itu.edu.tr (S.J.); mustafaergen@gmail.com (M.E.)
2   Department of Electrical and Electronics Engineering, Dogus University, Istanbul 34775, Turkey; lida.kouhalvandi@ieee.org
3   School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia; hadri@utm.my
4   Faculty of Engineering and Built Environment , Universiti Sains Islam Malaysia, Nilai 71800, Malaysia; hafizal@usim.edu.my
*   Correspondence: shayea@itu.edu.tr

Abstract: Multi-access edge computing (MEC) is a key technology in the fifth generation (5G) of mobile networks. MEC optimizes communication and computation resources by hosting the application process close to the user equipment (UE) in network edges. The key characteristics of MEC are its ultra-low latency response and real-time applications in emerging 5G networks. However, one of the main challenges in MEC-enabled 5G networks is that MEC servers are distributed within the ultra-dense network. Hence, it is an issue to manage user mobility within ultra-dense MEC coverage, which causes frequent handover. In this study, our purposed algorithms include the handover cost while having optimum offloading decisions. The contribution of this research is to choose optimum parameters in optimization function while considering handover, delay, and energy costs. In this study, it assumed that the upcoming future tasks are unknown and online task offloading (TO) decisions are considered. Generally, two scenarios are considered. In the first one, called the online UE-BS algorithm, the users have both user-side and base station-side (BS) information. Because the BS information is available, it is possible to calculate the optimum BS for offloading and there would be no handover. However, in the second one, called the BS-learning algorithm, the users only have user-side information. This means the users need to learn time and energy costs throughout the observation and select optimum BS based on it. In the results section, we compare our proposed algorithm with recently published literature. Additionally, to evaluate the performance it is compared with the optimum offline solution and two baseline scenarios. The simulation results indicate that the proposed methods outperform the overall system performance.

Keywords: fifth generation (5G); sixth generation (6G); handover (HO); multi-access edge computing (MEC); mobility management; task offloading (TO)

1. Introduction

The technological evolution is currently increasing with the creation of new and up-to-date technology that obtains information from carried data. Based on Cisco's new annual Internet report, 5G will support more than 10% of the world's mobile connections by 2023. There will be nearly 30 billion devices/connections by 2023—5% of those will be mobile [1].

The emergence of the modern multi-access edge computing (MEC) technology is a key discovery in the fifth generation (5G) and future sixth generation (6G) mobile networks because it optimises communication and computation resources effectively [2–6]. This process

employs edge resources near the user equipment (UE) to reduce latency while enhancing reliability and stability [7,8]. The offloading task sends computational tasks/data to the MEC server for processing [9,10]. Once the MEC executes the received task, it is responsible for sending the results back to the appropriate user. Edge computing has several advantages, such as decreased end-to-end latency, increased multimedia bandwidth, computation-rich resources, and enhanced flexibility. The drawback with most UEs is resource limitation due to computation requirements, even though new and improved hardware is available. However, MEC could solve this issue by hosting the computational tasks.

This study proposes two algorithms for choosing the best parameters in the optimisation function. Online algorithms are used to enhance the performance evaluation. The handover (HO) time cost is also considered in the formulation of the optimisation function while making optimum offloading decisions. The two algorithms presented in this work assume that the upcoming future tasks are unknown, and online task offloading (TO) decisions are therefore considered. The proposed methods are not model-based and can be implemented in any mobility scenario. In the first algorithm, called the online UE-base station (BS) algorithm, the users have both the user-side and BS-side information. In the second algorithm, BS-learning, users only have user-side information. The overall simulations are conducted using the MATLAB tool, and are verified in the case where the BS-side information is known. Our proposed method exhibits near to optimum performance. For the scenario based on observation and learning, the results reveal a decrease in performance due to additional costs, such as the HO time cost. The energy performance is also compared with other related papers, proving that in terms of the energy budget, our two proposed methods exhibit near to the optimum solution.

The remainder of this paper is organised as follows: A comprehensive literature review with the motivation of this work is provided in Section 2. Section 3 provides the system model analysis. Section 4 presents the problem formulation and the optimisation problem. Section 5 discusses the proposed online task offloading decision algorithms, problem formulation, and proposed online algorithms. Section 6 displays the simulation setting and numerical results. Finally, Section 7 includes the conclusion of this paper and additionally future tasks that can be employed for improving this manuscript.

2. Literature Review and Motivation of This Work

2.1. Literature Review

Recently, various studies have been published in the MEC domain, and this section provides a comprehensive review of these publications.

The mobility management that has been widely used in various applications such as IoT generates realistic mobility patterns [11,12]. In [13], a mobility-aware hybrid flow rule cache scheme is presented for tackling the problem of forwarding node. From another point of view, edge computing can provide the solutions for the cloud limitations in the current communication systems [14].

Latency factor is an important specification in a vehicular network that results in delays, generated from the high mobility of vehicles [15,16]. Data traffic is expanding with the increase in massive connectivity. The favourable characteristics of MEC technology are suitable for emerging 5G networks as they include ultra-low latency response and real-time applications [17,18]. It is expected that mobile broadband will increase to 8 billion by 2025. In 2017, the European Telecommunications Standards Institute (ETSI) changed the name 'mobile edge computing' to 'multi-access edge computing' for addressing non-cellular operators. MEC servers can now be deployed with a radio access network (RAN), BSs, Wi-Fi access points, and fixed connections [19]. The 3GPP includes MEC technology in the 5G network with the technical specification of 3GPP TS 23.051. Figure 1 presents a comprehensive summary of MEC servers with RAN, BSs (e.g., 4G, 5G, and 6G), Wi-Fi access points, and fixed connections.
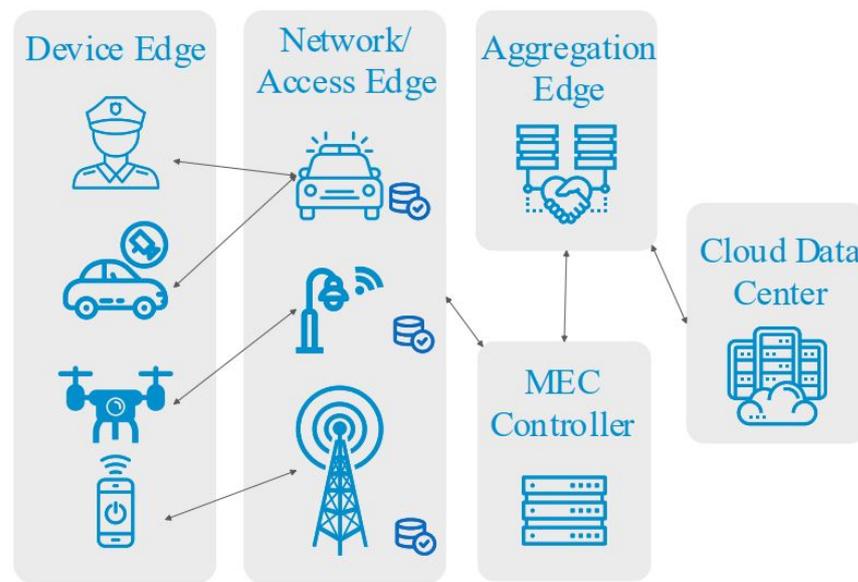
Figure 1. From cloud to network edges.

The MEC technology can be employed in various applications such as video analytics, location services, and data caching, which allows for flexible and rapid innovative applications. It can also be used in an autonomous vehicular network, sharing information with roadside units (RSU) and pedestrians without involving any cloud servers [20–22]. The development of MEC at the network edge is critical. The devices and sensors of the IoT require significant storage, computation resources, and suitable bandwidth, as the ever-increasing data have caused limited computational resources [23].

The MEC technology has two main challenges [4,9]. The first challenge is appropriately allocating MEC resources, as multiple users are within the coverage region of the MEC server. Once the computational resource is allocated to a user, it will then decide whether to offload the task or execute it locally. Multiple BSs are available for a single user. Choosing the optimum BS for offloading is the current research topic in various MEC-related studies [24–28]. The most common solution is to develop a utility cost function that determines the optimum solution based on the optimisation methods. Because resource allocation, communication, computation, and user mobility must all be considered, joint optimisation problems and analysis should be accomplished. Most joint problems are non-convex and NP-hard [29–31].

The second main challenge in MEC-enabled 5G networks is that MEC servers are distributed within ultra-dense networks [3,32]. Managing user mobility within the small-scale coverage of the MEC server is a significant issue [33–36]. In the recently published literature, MEC often neglects user mobility and assumes they are constant due to complexity [37–39]. Tables 1 and 2 present the summary of mobility management and the related studies in this field in a comprehensive view, respectively. Table 2 demonstrates that our proposed method, in comparison with other reported methods including machine learning approach, is advanced as energy consumption of UEs, handover delay, and task offloading are concurrently considered. In simple words, what is lacking in the previous studies is concurrently considering these parameters. Our method leads the way of designers in mobility-aware offloading decisions.

Table 1. Summary of mobility management challenges in MEC.

Ref.	Topic	Challenges
[40]	Pre-configuration	- Pre-allocation: To enable MEC systems for reducing end-to-end delay during high mobility; - Reallocation group: To share application information among MEC server for HO scenario.
[4,41]	HO prediction	- To choose essentially an optimum server sequentially as vehicles move; - To study the HO in urban areas due to the higher uncertainty.
[42,43]	Densification	- To manage the mobility of users in an ultra dense network (UDN); - To provide small-scale coverage results: frequent HOs, HO process power consumption, and radio resource constraint; - To prepare real-time information for long-term optimisation.
[44]	Network diversity	- To offload tasks to MEC servers over 2G, 3G, 4G, 5G, WLAN, or and overlapping mobile WiMAX networks by UEs. (These various types of networks causes overlapping).
[44,45]	Self optimization	- To make HO decision in MEC, where the HO control parameter needs to be assigned optimally; - To optimise both HO and task offloading in MEC result conflicts in optimisation function.

Table 2. Summary of related studies on handover decision optimisation in MEC.

Ref.	Topic	HO	TO	Limits	Contribution
[41]	Task offloading	Yes	Yes	The energy consumption of UEs are not considered.	- Task offloading decision was formulated as MDP to minimize delay considering handover, migration, communication and computation; - Addressed the uncertainty of transition probabilities.
[4]	Joint optimisation	Yes	Yes	- Only one handover is considered for each UE; - Not applicable for 3D random mobility (limited for 1D road)	- Mobility-aware computation offloading was proposed in MEC-based vehicular networks.
[24,46]	Joint optimisation	No	Yes	- No handover; - No mobility model.	Service caching placement and computation offloading is considered.
[25,29]	Proactive network association	No	Yes	- No handover; - No mobility model; - No UE energy constraint.	- Online proactive caching is considered; - Based on the MDP and Lyapunov optimisation, a two-stage online decision algorithm for proactive network association was innovated.
[9]	Task offloading	No	Yes	Mobility model has no handover cost (i.e., dedicated as a future work)	- Mobility-aware multi-user offloading optimisation for MEC; - UEs energy consumption is included.
[30]	Handover management	Yes	Yes	The system model only reduces number of handovers No handover delay or cost is optimized.	Existing work focused on cloudlet placement and user-to-cloudlet association problem.

Table 2. *Cont.*

Ref.	Topic	HO	TO	Limits	Contribution
[43]	Ultra-dense network	Yes	Yes	The parameter in optimization function leads into sub-optimum MEC and lower performance	Minimized average delay subjected to communication, computation, and handover under the limited energy budget of users.
[31]	Ultra-dense network	No	Yes	No handover (mentioned as challenge but not considered).	Computation offloading for multi-access MEC in UDN is investigated
[8]	Fog computing	No	Yes	Mobility model has no handover cost (mentioned as challenge but not considered)	Task offloading and migration schemes are studied in fog computing
[33]	Fog computing	Yes	Yes	The energy consumption of UEs is not considered (i.e., dedicated as a future work)	- Offloading delay and handover cost are considered as performance metrics; - The RSU considered the targeted node based on vehicle mobility and dynamic computation resources.
[47]	Deep reinforcement learning task scheduling	No	Yes	The tasks are completed before handover (i.e., only controls HO) No handover cost	- DRL is used for offloading scheduling in VEC; - The trade-off between task latency and energy consumption is considered by scheduling tasks to wait in the queue.
[48]	Mobility management using reinforcement learning	Yes	Yes	The energy consumption of UEs is not considered.	An online RL was proposed to optimize handover decisions by predicting user movement trajectory and periodic characteristics of the number of users.
[49]	Edge autonomous energy management	No	No	Only energy managed	An RL-based droplet framework is used Droplets learn energy consumption statistics of the devices.
Our work	Mobility-aware offloading decision	Yes	Yes	In the offloading process, the BSs are considered to be all ON.	- Concurrently considering HO delay, UE energy, and TO. - Optimized parameters are used to increase overall performance. - User-centric approach learns BS-side information without prior knowledge.

The conventional MEC-based offloading methods are not powerful enough, and advanced optimisation methods are required. Xuefei presents a method, the Markov decision process (MDP), for reducing the amount of delay in high mobility vehicular networks [41]. Another method of joint optimisation is presented in [50,51] for vehicle random mobility. In [52], the boundless simulation area (BSA) model for configuring the vehicles's mobility is presented, which is based on the Markov chain. For the IoT technology, a proactive network association can be a sufficient solution for reducing latency in MEC systems [53,54]. As a solution for reducing the handover in MEC-based networks, in [55], a region partitioning approach is presented, where this method is evaluated for both random and real traces. In another work [56], the dynamic service placement is employed, where offline algorithms are used for determining the optimal service for minimizing the average cost. To support the energy-constrained devices such as IoT sensors, the wireless powered MEC networks are integrated with simultaneous wireless information and power transmission (SWIPT) techniques in [57]. In another work [58], the

task offloading and wireless power transformation (WPT) are jointly optimized. Moreover, the authors proposed a resource allocation scheme to minimize energy consumption.

There are important limitations in the previous summarized literature in that these works do not include HO cost in the optimisation function. Additionally, either energy factor or delay is considered in these methods where the optimum parameter cannot be selected effectively.

2.2. Motivation of This Work

This work considers user mobility while having optimum offloading decisions. To support service continuity, the host MEC should reallocate the task to the target MEC server. During the task offloading, if users move from the coverage region of a base station, it will cause handover that leads to additional cost. Hence, our problem formulation includes the handover cost. Below is a summary of the two employed algorithms.

In the first algorithm, calculating the optimum BS for offloading is possible because the BS information is available. No handover will be presented, making the performance close to the optimal offline solution. However, we must ensure that users' total energy consumption will not exceed its budget as the energy budget is limited.

In the second algorithm, the BS-side information is not available. This means that the channel gains and the available computing CPUs are unknown. Therefore, users must learn time and energy costs through observations. The optimum BS is determined based on these observations. However, sub-optimum BS selection may occur due to the variance of observation. Selecting sub-optimal BS will increase the total cost and number of handovers. Therefore, it is essential to select optimum controlling parameters in the optimisation function.

The novelty and contribution of this manuscript can be summarized as follows:

- Our proposed method is beneficial for the designers in the field of mobility management in MEC, as this approach considers three important parameters—energy consumption of UEs, handover delay, and task offloading—concurrently. This jointly consideration is missing in the reported literature;
- The proposed method is intelligent enough for finding the optimal value of alpha presented in Equation (18), resulting in average time and total energy costs near to the optimum offline solution;
- In our proposed methods, the offloading decision is user-centric, which is decided on the UE-side. Moreover, Algorithm 2 initially has neither base station nor network information. In this case, the learning process is used with optimized steps to reduce overall delay and improve energy performance.

3. System Model

The MEC network consists of the UEs and N BSs equipped with the MEC server. The BSs are distributed on a finite two dimensional (2D) regular grid network, each with a supporting radius of R (as shown in Figure 2). Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the index set of MEC servers, and l_n represents the location of BS n equipped with the MEC server. The mobile UEs create a total M task for offloading into MEC servers, denoted by the set of $\mathcal{M} = \{1, 2, \ldots, M\}$. Let l_m signify the location of task m created by a user with mobility. Due to the densified deployment of BSs, it is assumed that each task m can be served by multiple BSs. The set of admissible BSs to location l_m can be represented as

$$\mathcal{A}(m) = \{n| \parallel l_n - l_m \parallel \leq R, \forall n \in \mathcal{N}\} \tag{1}$$
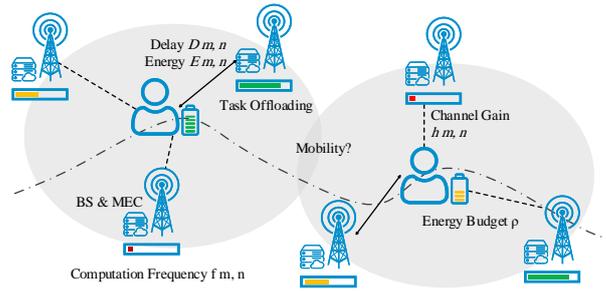
Figure 2. System model of mobility-aware task offloading in MEC enabled 5G network.

The users decide which candidate BSs $\mathcal{A}(m) \in \mathcal{N}$ are appropriate to offload computational tasks based on the utility cost. It is also assumed that only one BS is responsible for computing each task m. Hence, users will choose one BS station from $\mathcal{A}(m)$ for each task m. Due to large task sizes, each task m can be divided into many sub-tasks. An online learning approach is considered to manage the user mobility. Any random mobility model can be applicable and suitable. The system model has been divided into three sub-models: computation task model, network model, and mobility model. The explanations of each sub-model follow.

3.1. Computation Task Model

Here, it is supposed that a UE generates task m to offload into the MEC server. The computational tasks can be parameterized as triplets in (2):

$$\chi \triangleq [d_m, c_m, t_m] \tag{2}$$

where d_m specifies the total data size of task m (in bits). The computation intensity c_m refers to the number of required CPUs to accomplish computing one-bit data of a task. The t_m indicates the computation deadline time allowed for executing task m (in seconds). As the size of the computed result is generally small, it is omitted from the equation. To consider large input data sizes, d_m can be divided into k_m number of equal-sized sub-tasks. Therefore, $d_m = k_m d_{sub}$, and d_{sub} denote to each sub-task data size.

As the UEs decide to offload the computational task, the MEC server is responsible for resource allocation based on available resource constraints. The computation resource inside the MEC server is indicated based on the CPU cycle frequency. Let $F(n)$ represent the total computing constraint of the MEC server n, which indicates the maximum available CPU cycle frequency of the MEC server. The computation frequency $f_{m,n}$ (denoted as CPU frequency) is the total resource allocated by MEC server n into the UE with requesting task m for computation. Accordingly, the time cost for executing a sub-task of m by the MEC server n is given as (3):

$$t_{m,n}^e = d_{sub} \times \frac{c_m}{f_{m,n}} \tag{3}$$

where d_{sub} specifies the sub-task data size. Once all sub-tasks are computed by MEC server m, which takes a total duration of $T_{m,n}^{tot} = \sum_{k=1}^{k_m} t_{m,n}^e$, it deallocates resources so other UEs could use deallocated CPU for their tasks.

3.2. Network Model

In this study, it is assumed that channel bandwidth ω is equally allocated among tasks. According to [59], the data transmission rate from the UE with task m to the BS n is obtained by (4):

$$r_{m,n} = \omega \times \log_2(1 + \frac{p_m h_{m,n}}{\mathcal{N}_{\sigma^2}}) \tag{4}$$

where $\mathcal{N}_{\sigma^2}$ represents the noise power, p_m is the transmission power of UE m, and $h_{m,n}$ denotes the wireless channel gain from UE with task m to BS n at location l_m, respectively. It is assumed that users are constant during task offloading. As each task m can be divided into multiples sub-tasks, the channel gain $h_{m,n}$ will remain constant. The wireless channel gain $h_{m,n}$ is described as (5):

$$h_{m,n}[dB] = 127 + 30 \times \log d[Km] \tag{5}$$

where $d = \| l_n - l_m \|$ refers to the distance between task m at location l_m and BS n (in kilometers). The channel model is as suggested in [60], which is for ultra-dense heterogeneous networks, and the fading follows a Rayleigh distribution.

Additionally, the time and energy costs of offloading a sub-task of m to BS n, respectively, can be presented using (6) and (7):

$$t^o_{m,n} = \frac{d_{sub}}{r_{m,n}} \tag{6}$$

$$e^o_{m,n} = p_m t^o_{m,n} \tag{7}$$

where p_m specifies power required to transmit m. The p_m is assumed as given for each task, and it depends on many parameters, including antenna gain.

3.3. Mobility Model

The large data size of tasks can be divided into multiple sub-tasks. Each sub-task can be executed in various BSs. Changing the BSs while computing sub-tasks of m can result in HO delay cost.

Let τ^h_m denote HO delay of a one-time BS switch. Considering the multiple sub-tasks, the sequence of BS is represented by $\eta_m = \{\eta^1, \eta^2, \ldots, \eta^{k_m}\}$. According to [43], the HO delay for all sub-tasks of m can be calculated as follows:

$$t^h_{m,n} = \tau^h_m \sum_{k=2}^{k_m} \mathbb{H}\{x\} \tag{8}$$

$$\mathbb{H}\{x\} = \begin{cases} 1 & \eta^k \neq \eta^{k-1}, \quad \eta^k \in \mathcal{A}(m) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

4. Problem Formulation

In the mobile system, the UE experience during offloading decisions is determined by both latency and energy budget. A utility function must be designed as a trade-off between the time and energy budgets to make the appropriate decision when the BS is to offload and perform HO due to the UE mobility. The total time cost for task m can be calculated as the sum of execution time, task offloading time, and HO time, as in (10):

$$T^{tot}_{m,\eta} = \sum_{k=1}^{k_m} (t^e_{m,\eta} + t^o_{m,\eta}) + t^h_{m,\eta}. \tag{10}$$

The total time cost can be written as (11):

$$T^{tot}_{m,\eta} = \sum_{k=1}^{k_m} (t^e_{m,\eta} + t^o_{m,\eta} + \tau^h_m \mathbb{H}\{x\}). \tag{11}$$

This study only considers the task offloading energy cost of UEs. The total energy consumption of UE for offloading task m is calculated by

$$E^{tot}_{m,\eta} = \sum_{k=1}^{k_m} e^o_{m,\eta}. \tag{12}$$

The decision-making process cannot predict the UE trajectory after the computation deadline time t_m while considering UE mobility during task offloading. Therefore, the total time cost needs to be less than the time constraint t_m presented as

$$T_{m,\eta}^{tot} \leq t_m, \forall m \in M \tag{13}$$

Eventually, the UE's limited energy budget will restrict the sum of the energy cost M tasks to be less than energy constraint ρ. The total UE energy is constrained as below:

$$\sum_{m=1}^{M} E_{m,\eta}^{tot} \leq \rho. \tag{14}$$

As previously noted in Equation (1), all candidate BS η for task m should be within coverage area with a radius of R, denoted as (15):

$$\eta_m^k \in \mathcal{A}(m), \forall k \in \{1, 2, \ldots, k_m\}. \tag{15}$$

Our target is to determine the optimal offloading decision to minimize the total time cost within a limited energy budget for all tasks M. The problem is formulated as follows (16):

$$\text{GP:} \min_{\eta} \frac{1}{M} \sum_{m=1}^{M} T_{m,\eta}^{tot} \tag{16}$$

where $\eta \in \{\eta_1, \eta_2, \ldots, \eta_M\}$ is the variable vectors to be optimised. Constraint (13) ensures that the task will be executed on time (subject to the computation deadline time t_m). Constraint (14) guarantees that the energy consumption is below the energy budget ρ. Additionally, constraint (15) chooses all the candidate BSs that can serve task m.

Due to the non-linearity of the optimisation problem presented in (16) and the complexity of other variables, the GP is a mixed-integer non-linear programming (MINLP) problem [61]. An online algorithm is proposed to address the UEs' trajectory during task offloading.

5. Online Task Offloading Decision Algorithm

This section presents two mobility-aware online task offloading decisions for MEC, based on the Lyapunov optimisation and without the knowledge of future tasks. The proposed online algorithms are then compared with the offline optimal solution.

5.1. Mobility-Aware UE-BS Algorithm

This algorithm simultaneously has UEs with both the UE-side state information and the BS-side information. The BS remains constant during the offloading of one task, as previously noted. Having both sides' state information helps the UE to select the optimal BS for offloading and avoids any HOs. Within the UE-BS scenario, all sub-tasks of m will be served by a single optimum BS η_m^{opt}. The main challenge for solving GP in an online algorithm is that without having future task information, $m + 1, m + 2, \ldots, M$, the limited energy budget for current tasks will be spent and nothing will remain for upcoming tasks. To overcome this issue, the solution is to define an energy queue and store the used energy budget. By exceeding a specific amount of energy budget for task m, future tasks will be allocated to another candidate BS $\eta \in \mathcal{A}$. According to [43], the energy queue is obtained as (17):

$$\beta_{m+1} = \max\{\beta_m + E_{m,\eta_m^{opt}}^{tot} - \rho/M, 0\} \tag{17}$$

where β_0 is equal to zero. By using the Lyapunov optimisation, we can define the optimisation problem as a trade-off between time cost and energy cost of task offloading, represented as (18):

$$Z_m^{tot} = \alpha T_{m,n}^{tot} + \beta_m E_{m,n}^{tot} \tag{18}$$

where α is a control parameter to adjust the trade-off between energy and time cost. No handover will occur as we have both sides' information in the UE-BS scenario and we selected the appropriate BS for the entire time of processing task m. The total time cost can be reduced to $T_{m,\eta}^{tot} = \sum_{k=1}^{k_m}(t_{m,\eta}^e + t_{m,\eta}^o)$. As all sub-tasks will be served by a single optimum BS η_m^{opt}, the optimisation function in (18), can be simplified as

$$z_m = \alpha\left(t_{m,\eta_m^{opt}}^e + t_{m,\eta_m^{opt}}^o\right) + \beta_m e_{m,\eta_m^{opt}}^o. \tag{19}$$

To minimize the cost z_m in Equation (19) for each task m, the online UE-BS algorithm presented in Algorithm 1 is proposed.

Algorithm 1: Mobility-aware online UE-BS algorithm

Input: $\mathcal{A}(m)$, d_m, c_m, $f_{m,n}$, $h_{m,n}$ and α
1: **if** $m = fJ + 1, \forall f = 0, 1, \ldots, F-1$ **then**
2: $\quad \beta_m \leftarrow 0$
3: **end if**
4: Choose η_m^* subject to (13), (15) by solving
$$\min_{n \in \mathcal{A}(m)} \alpha\left(t_{m,\eta_m^{opt}}^e + t_{m,\eta_m^{opt}}^o\right) + \beta_m e_{m,\eta_m^{opt}}^o$$
5: Update β_m according to (17).

Because there is no handover within this scenario, there is no need to divide tasks into sub-tasks. Therefore, according to the optimization function in line 4, the Algorithm 1 complexity only relies on the number of candidate BS $\mathcal{A}(m)$, which is $\mathcal{O}(|\mathcal{A}(m)|)$.

5.2. Mobility-Aware BS Learning Algorithm

In this algorithm, the UEs only have the UE-side state information. UEs are required to learn the BS state information to make offloading decisions. Unlike the UE-BS algorithm where all UEs stick to a single and optimum BS for all sub-tasks of m, the learning process causes sub-optimal BS selection. Choosing sub-optimal BS will cause additional cost as well as HOs during the learning process.

One solution for learning optimal BS is to offload all sub-tasks of m to every candidate BS $\eta_m \in \mathcal{A}(m)$ and to observe the total energy and time costs. Let $\tilde{t}_{m,n}$ and $\tilde{e}_{m,n}$ represent the observed time cost and energy cost, respectively. Based on Equation (18), the observed optimisation function is

$$\tilde{z}_{m,n} = \alpha\tilde{t}_{m,n} + \beta_m\tilde{e}_{m,n} \tag{20}$$

where $\tilde{z}_{m,n}$ is a noisy version of $z_{m,n}$ with a specified variance. The main challenge in the BS learning scenario is that UEs may offload their sub-tasks into sub-optimal BSs due to the variance of $\tilde{z}$. The UE will try to offload their tasks into all BSs to learn the best optimum BS. However, offloading many tasks is not practical and may cause frequent HOs. A possible solution is to assign a stop parameter for the learning process. Therefore, the algorithm will only apply to the first k_s sub-task, and the remaining sub-tasks will be offloaded to pre-determined optimal BS. The second challenge is choosing the stop learning parameter. The large k_s may cause frequent HOs and may increase costs, whereas the small k_s will lead to the selection of sub-optimal BS. The proposed Algorithm 2 presents the online UE-BS algorithm for minimising the cost $\tilde{z}_{m,n}$ in Equation (20) for each task m.

Algorithm 2: Mobility-aware online BS learning algorithm

Input: $\mathcal{A}(m)$, d_m, c_m and α.

1: **if** $m = fJ + 1, \forall f = 0, 1, \ldots, F - 1$ **then**
2:　　$\beta_m \leftarrow 0$
3: **end if**
4: **for** $k = 1, \ldots, |\mathcal{A}(m)|$ **do**
5:　　connect o each BS $n \in \mathcal{A}(m)$ once.
6:　　Update $\bar{z}_{m,n,k} = \alpha \tilde{t}_{m,n} + \beta_m \tilde{e}_{m,n}$.
7:　　Update $\theta_{m,n,k} = 1$.
8: **end for**
9: **for** $|\mathcal{A}(m)| + 1, \ldots, k_s$ **do**
10:　　Connect to $\eta_m^k = \arg\min_n \{ \bar{z}_{m,\eta_m^k,k} - \beta \sqrt{\frac{2 \ln k}{\theta_{m,n,k}}} \}$
11:　　Observe $\tilde{t}_{m,\eta_m^k}$ and $\tilde{e}_{m,\eta_m^k}$.
12:　　$\bar{z}_{m,\eta_m^k,k} \leftarrow \frac{\theta_{m,\eta_m^k,k} \bar{z}_{m,\eta_m^k,k} + \alpha \tilde{t}_{m,\eta_m^k} + \beta_m \tilde{e}_{m,\eta_m^k}}{\theta_{m,\eta_m^k,k} + 1}$.
13:　　$\theta_{m,\eta_m^k,k} \leftarrow \theta_{m,\eta_m^k,k} + 1$.
14: **end for**
15: **for** $k_s + 1, \ldots, k_m$ **do**
16:　　Connect to $\eta_m^{k_s}, \forall k \in \{k_s + 1, k_s + 2, \ldots, k_m\}$
17: **end for**
18: Update β_m according to (17).

In Algorithm 2, the task is divided into k_m number of sub-tasks. Therefore Algorithm 2 has service continuity despite handover because of dividing the task into small-sized sub-tasks. However, these division introduces additional complexity, as seen in Algorithm 2. As noted above, the algorithm only applies into k_s number of tasks, which results in computation complexity of $\mathcal{O}(k_s|\mathcal{A}(m)|)$. This means the small learning step k_s leads to lower complexity. However, as noted above, the lower the k_s means sub-optimal BSs may be selected. Therefore, there is a trade-off between having optimum BS and lower computation complexity in the algorithm.

6. Simulations Results and Discussion

This section provides the simulation experiments for evaluating the proposed methods. Table 3 summarises the main simulation parameters with their values. The coverage region of each MEC-equipped BS is considered to be a circle with a radius of 150 m deployed on a regular grid network within an area of 1000×1000 m^2, as shown in Figure 3. The channel bandwidth of BS is set to be $\omega = 20$ MHz. The channel gain from the UE to BS is modelled as $127 + 30 \log d$ (km) dB [60]. All wireless communication parameters are set based on the 3GPP specification [62].

According to [43], the one-second video size should be set to $d_{sub} = 0.62$ Mbits. Let us assume that each video is 60 s to 120 s long, and $k_m \in \{60, \ldots, 120\}$. According to the following definition, $d_m = k_m d_{sub}$, the input data size is uniformly distributed $d_m \sim U(37.2, 74.4)$ MBits. The same consideration holds for computation intensity, which is uniformly distributed within U(500, 1000) cycles/bits. Each MEC server m allocates its available computational CPU resource to its users ($f_{m,n}$), and the total used CPU should not reach the resource constraint of that MEC server (F_n). The MEC server n uniformly distributes the resource constraint $F_n = 20$ GHz to its UEs. For simplicity, it is assumed that the available CPU frequency on all BSs are equal and are set to 25 GHz. To ensure that execution of each sub-task is completed within its latency constraint, t_m is set to be 150 ms. After reaching the total execution time of a task $T_{m,n}^{tot} = \sum_{k=1}^{k_m} t_{m,n}^e$, the MEC server deallocates the resource $f_{m,n}$ and makes it available for reallocation to other users as soon as they request it.
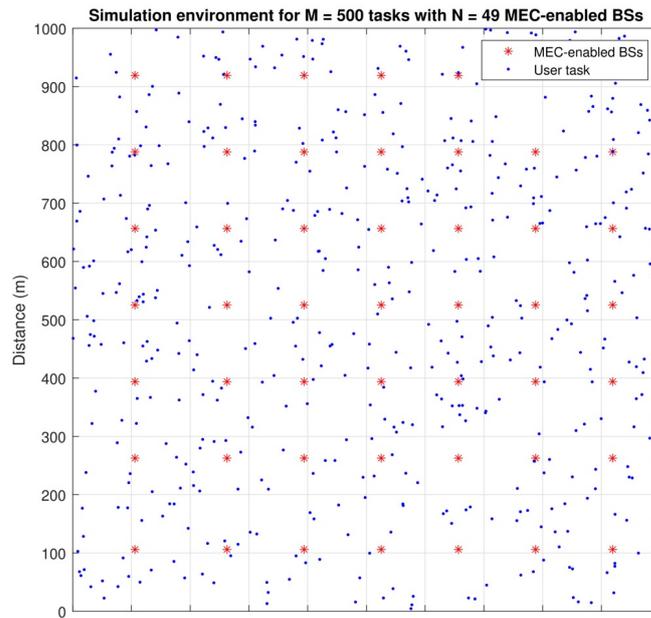
**Figure 3.** The location of BSs and users within simulation region.

Table 3. Employed simulation parameters with corresponding values.

| Parameters	Value
Radius of the BS coverage area R	150 m
Availbe CPU frequency on BS F_n	25 GHz
Channel bandwidth ω	20 MHz
Channel gain from UE to BS $h_{m,n}$	$127 + 30 \log d(km)$ dB
Subtask size	$d_{sub} = 0.62$ Mbits
Input data size of each task d_m	$[37.2, 74.4]$ Mbits
Computation intensity of each task c_m	$[500, 1000]$ cycles/bit
Total available computation CPU for each task m by BS n $f_{m,n}$	$[0, F_n]$
Computation deadline of each task t_m	150 ms
Noise power $\mathcal{N}_{\sigma^2}$	2×10^{-13} W
UE transmission power p_m	0.5 W
One-time handover delay τ_m^h	5 ms
Battery capacity	1 kJ

The number of tasks varies for different simulations. We consider a fixed number of BSs as $N = 49$, where all are active the entire time. It is also assumed that there is at least one BS to provide service to each UE. The transmit power of mobile UEs are $p_m = 0.5$ W, and the noise power is set to be $\mathcal{N}_{\sigma^2} = 2 \times 10^{-13}$ W. The one time HO is $\tau_m^h = 5$ ms, and the battery capacity is $J = 1$ kJ.

This section compares our two proposed methods, UE-BS and BS-learning algorithms, with the optimum offline solution as well as the two common benchmark algorithms known as time greedy and energy greedy. The performance of the proposed methods are evaluated and compared with the related study of [43].

Figure 4 compares the average time cost and total energy cost for all algorithms over different task sizes $\mathcal{M}$. As seen in Figure 4a, both EMM-GSI and the proposed UE-BS algorithms have their time cost close to the optimum offline solution due to possessing BS-side information. However, EMM-LSI and the proposed BS-learning have additional costs due to sub-optimal BS selection and handover cost.
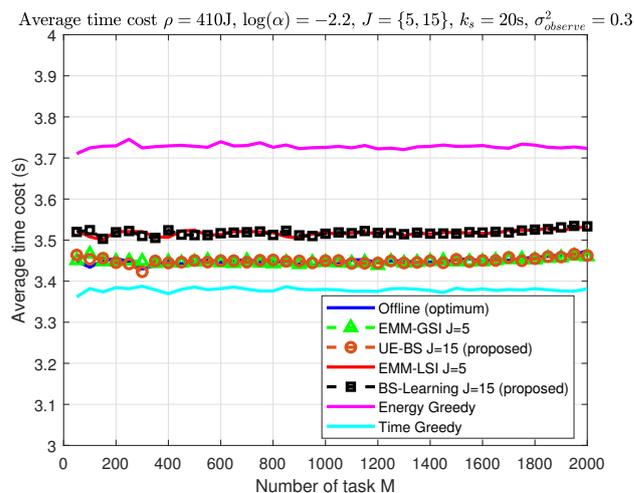
(a)
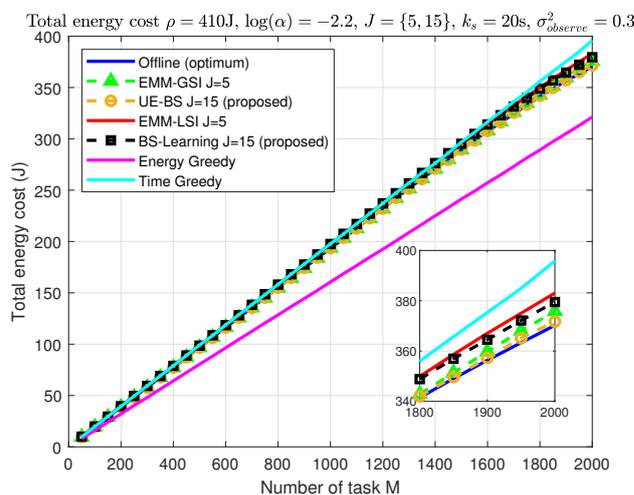
(b)

Figure 4. Performance evaluation of algorithms ($\rho = 410$ J, $\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$ s, $\sigma_{observe}^2 = 0.3$): (**a**) average time cost; (**b**) total energy cost.

For the higher number of task size in Figure 4b, the proposed methods exhibit slightly better performance compared to existing studies that have energy cost near optimum for the offline solution. The proposed methods maintain energy consumption below the energy budget ρ while achieving relatively low time cost.

Figure 5 displays the trade-off between the average time cost and the total energy cost for α from 10^{-4} to 10. The α is the controlling parameter in the optimisation solution. We must therefore ensure it is optimally selected. The interception points indicate the optimum α values, as presented in Figure 5.

In Figure 5a, the BS-side information is presented. In Figure 5b, the proposed algorithm learns the total cost throughout observations. As can be seen, the interception points are almost the same, varying between $[-2.3, -2.1]$. To conveniently evaluate all algorithms in the same condition, the controlling parameter is set to be $\log(\alpha) = -2.2$. Due to the HO and learning process in Figure 5b, the time and energy costs are slightly higher than those in Figure 5a.
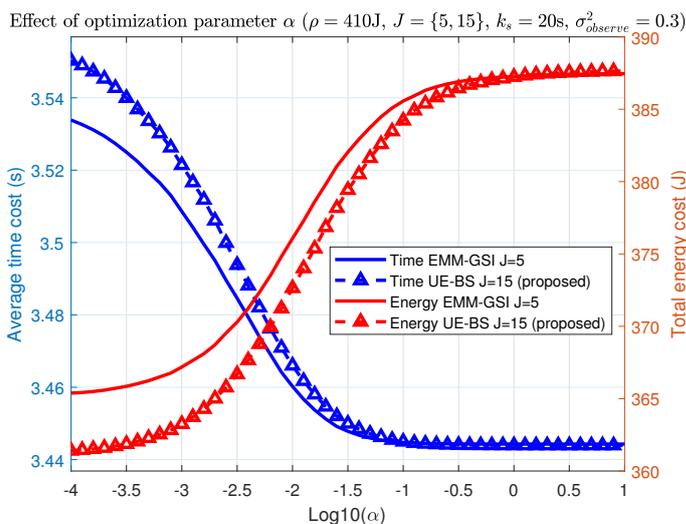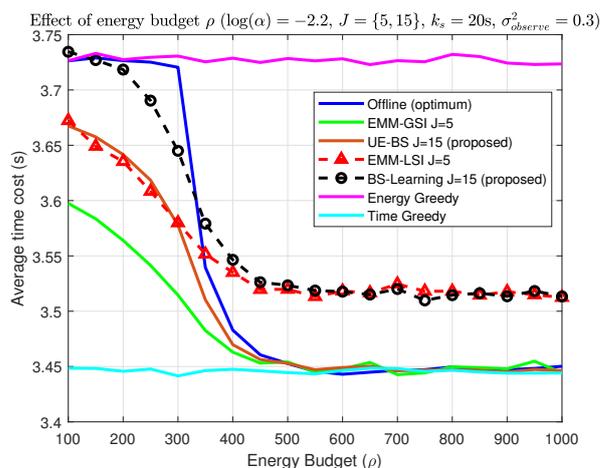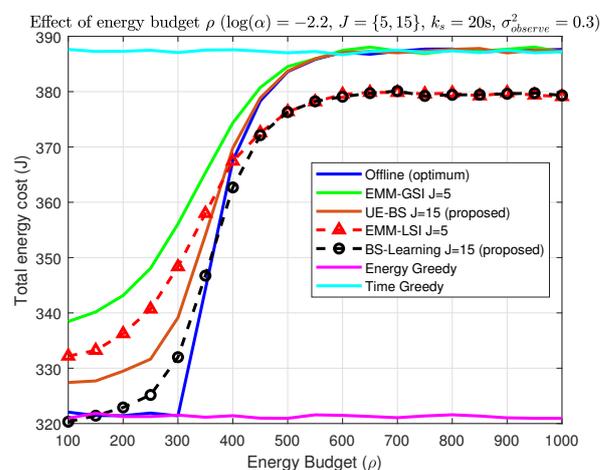
(**a**)
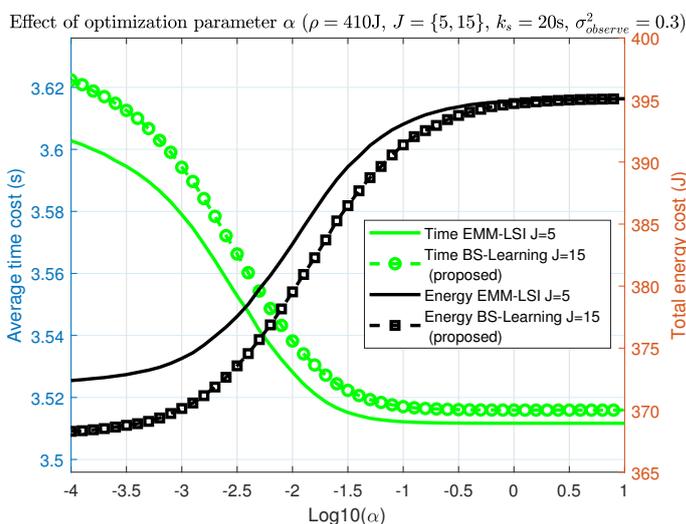
(**b**)

Figure 5. Effect of optimization parameter α ($\rho = 410$ J, $J = \{5, 15\}$, $k_s = 20$ s, $\sigma^2_{observe} = 0.3$): (**a**) online UE-BS algorithm; (**b**) online BS-learning algorithm.

Figure 6 reveals the performance of the algorithms for various energy budgets ρ. The figures indicate that by increasing the energy budget ρ, all algorithms will achieve similar performance as the time greedy algorithm. Users will have enough energy budget to neglect cost and stick to the BS with the lowest average time cost. It must be noted that if users do not have BS-side information they will experience performance loss (see Figure 6).

Therefore, the EMM-LSI and proposed BS-learning algorithms are not going to reach the optimum offline cost level. As a result, EMM-LSI and the proposed BS-learning algorithms will not reach the optimum offline cost level. The next important point is the slope of the lines, as shown in Figure 6. The results indicate that our proposed algorithms are near optimum for the offline solution compared to the algorithms in [43].

Effect of energy budget ρ ($\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$s, $\sigma^2_{observe} = 0.3$)

(**a**)

Effect of energy budget ρ ($\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$s, $\sigma^2_{observe} = 0.3$)

(**b**)

Figure 6. Effect of energy budget ρ ($\log(\alpha) = -2.2$, $J = \{5, 15\}$, $k_s = 20$ s, $\sigma^2_{observe} = 0.3$): (**a**) average time cost; (**b**) total energy cost.

7. Conclusions

Significant research has been accomplished to determine online task offloading decisions in MEC while considering user mobility, and a limited energy budget has been allocated to users. The optimisation function has been developed as a trade-off between the time cost and energy cost, not to exceed the energy budget. The online algorithms are considered as potential solutions as future upcoming tasks are unknown. In this manuscript, two scenarios have been presented. The first scenario is when both the user-side and BS-side information are available. The user will be able to calculate the optimum BS and remain for the entire task processing period. The second scenario is when users have no access to BS information. Unlike the first scenario, users cannot calculate the utility cost and make offloading decisions based on that. Instead, users will offload limited tasks to each BS and observe the total time cost and energy consumption. After determining the optimum BS, the remaining tasks are offloaded to that BS. The BS may change during task offloading due to sub-optimum BS selection in the BS-learning algorithm, resulting in additional costs and HOs. The simulation results indicate that our two proposed algorithms perform slightly better than those in existing literature. Additionally, a performance loss is provided in the second scenario due to HO, and the outcomes are similar to the optimal offline solution.

In future work, it would be better to consider resource allocation in the problem formulation. This requires solving NP-hard and non-convex problems. One potential

solution is to use machine learning (ML) approaches [63,64]. Applying ML approaches in MEC systems will be challenging because ML algorithms are extremely complex and computation resources are in high demand. Dividing learning computations into smaller tasks and distributing them among multiple MEC servers can be a possible solution. The issues would then be to determine the types of computations that can be divided as well as the method to divide while considering MEC resources. Merging the results obtained from different MEC servers for a specific task should also be managed and assessed. Due to UE mobility and frequent handovers among MEC servers, it is quite challenging to integrate outputs from various sub-tasks into a single output when the UE's trajectory and location are dynamic. Future work should employ deep neural network (DNN) with multi-objective functions to optimise HO and computation offloading. Additionally, for the scenario where the UEs are far from network coverage, we can implement a device-to-device (D2D) connection or use MEC servers inside vehicular systems. Furthermore, for simplicity, multipath propagation was not considered here, and future work needs to consider the multipath impact into the problem formulation. Furthermore, the BSs in both algorithms are assumed to be all activated during the offloading process. Recent technologies such as software-defined networking (SDN) require a programmable network that enables developers to switch ON/OFF the BSs. In future work, it would be better to integrate the SDN with the MEC technology. Finally, future work could also analyse the run-time complexity of algorithms and compare it with benchmarks.

Author Contributions: Methodology, Writing, results discussion and Analysis S.J.; Formal analysis, L.K.; Supervision, I.S. and M.E.; Funding acquisition; M.H.A. and H.M. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3GPP	Third Generation Partnership Project
5G	Fifth Generation of Mobile Networks
6G	Sixth Generation of Mobile Networks
BS	Base Station
BSA	Boundless Simulation Area
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
ETSI	European Telecommunications Standards Institute
HO	Handover
IoT	Internet of Things
ML	Machine Learning
MDP	Markov Decision Process
MEC	Multi-access Edge Computing

RAN	Radio Access Network
RSU	Roadside Unit
RL	Reinforcement Learning
SDN	Software-defined Network
TO	Task Offloading
UDN	Ultra Dense Network
UE	User Equipment
VEC	Vehicular Edge Computing

References

1. Cisco. New Cisco Annual Internet Report Forecasts 5G to Support More than 10% of Global Mobile Connections by 2023. Available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 20 March 2022).
2. Lieira, D.D.; Quessada, M.S.; Cristiani, A.L.; Meneguette, R.I. Algorithm for 5G Resource Management Optimization in Edge Computing. *IEEE Lat. Am. Trans.* **2021**, *19*, 1772–1780. [CrossRef]
3. Thananjeyan, S.; Chan, C.A.; Wong, E.; Nirmalathas, A. Mobility-Aware Energy Optimization in Hosts Selection for Computation Offloading in Multi-Access Edge Computing. *IEEE Open J. Commun. Soc.* **2020**, *1*, 1056–1065. [CrossRef]
4. Huy Hoang, V.; Ho, T.M.; Le, L.B. Mobility-Aware Computation Offloading in MEC-Based Vehicular Wireless Networks. *IEEE Commun. Lett.* **2020**, *24*, 466–469. [CrossRef]
5. Tuysuz, M.F.; Aydin, M.E. QoE-Based Mobility-Aware Collaborative Video Streaming on the Edge of 5G. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7115–7125. [CrossRef]
6. Ergen, M.; Inan, F.; Ergen, O.; Shayea, I.; Tuysuz, M.F.; Azizan, A.; Ure, N.K.; Nekovee, M. Edge on wheels with OMNIBUS networking for 6G technology. *IEEE Access* **2020**, *8*, 215928–215942. [CrossRef]
7. Nyanteh, A.O.; Li, M.; Abbod, M.F.; Al-Raweshidy, H. CloudSimHypervisor: Modeling and Simulating Network Slicing in Software-Defined Cloud Networks. *IEEE Access* **2021**, *9*, 72484–72498. [CrossRef]
8. Wang, D.; Liu, Z.; Wang, X.; Lan, Y. Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks. *IEEE Access* **2019**, *7*, 43356–43368. [CrossRef]
9. Zhan, W.; Luo, C.; Min, G.; Wang, C.; Zhu, Q.; Duan, H. Mobility-Aware Multi-User Offloading Optimization for Mobile Edge Computing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3341–3356. [CrossRef]
10. Liao, Z.; Ma, Y.; Huang, J.; Wang, J.; Wang, J. HOTSPOT: A UAV-Assisted Dynamic Mobility-Aware Offloading for Mobile-Edge Computing in 3-D Space. *IEEE Internet Things J.* **2021**, *8*, 10940–10952. [CrossRef]
11. Rabet, I.; Selvaraju, S.P.; Fotouhi, H.; Alves, M.; Vahabi, M.; Balador, A.; Björkman, M. SDMob: SDN-Based Mobility Management for IoT Networks. *J. Sens. Actuator Netw.* **2022**, *11*. [CrossRef]
12. Wei, H.; Luo, H.; Sun, Y. Mobility-Aware Service Caching in Mobile Edge Computing for Internet of Things. *Sensors* **2020**, *20*, 610. [CrossRef] [PubMed]
13. Kim, Y.; Park, J.; Kyung, Y. Mobility-Aware Hybrid Flow Rule Cache Scheme in Software-Defined Access Networks. *Electronics* **2022**, *11*, 160. [CrossRef]
14. Avgeris, M.; Spatharakis, D.; Dechouniotis, D.; Leivadeas, A.; Karyotis, V.; Papavassiliou, S. ENERDGE: Distributed Energy-Aware Resource Allocation at the Edge. *Sensors* **2022**, *22*, 660. [CrossRef] [PubMed]
15. Huang, A.; Nikaein, N.; Stenbock, T.; Ksentini, A.; Bonnet, C. Low latency MEC framework for SDN-based LTE/LTE-A networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]
16. Emara, M.; Filippou, M.C.; Sabella, D. MEC-Assisted End-to-End Latency Evaluations for C-V2X Communications. In Proceedings of the 2018 European Conference on Networks and Communications (EuCNC), Porto, Portugal, 8–11 June 2018; pp. 1–9. [CrossRef]
17. Xenakis, D.; Tsiota, A.; Koulis, C.T.; Xenakis, C.; Passas, N. Contract-Less Mobile Data Access Beyond 5G: Fully-Decentralized, High-Throughput and Anonymous Asset Trading Over the Blockchain. *IEEE Access* **2021**, *9*, 73963–74016. [CrossRef]
18. Ren, D.; Gui, X.; Zhang, K.; Wu, J. Mobility-Aware Traffic Offloading via Cooperative Coded Edge Caching. *IEEE Access* **2020**, *8*, 43427–43442. [CrossRef]
19. Taleb, T.; Samdanis, K.; Mada, B.; Flinck, H.; Dutta, S.; Sabella, D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1657–1681. [CrossRef]
20. Khan, M.A.; Ghosh, S.; Busari, S.A.; Huq, K.M.S.; Dagiuklas, T.; Mumtaz, S.; Iqbal, M.; Rodriguez, J. Robust, Resilient and Reliable Architecture for V2X Communications. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4414–4430. [CrossRef]
21. Huang, C.M.; Lai, C.F. The Delay-Constrained and Network-Situation-Aware V2V2I VANET Data Offloading Based on the Multi-Access Edge Computing (MEC) Architecture. *IEEE Open J. Veh. Technol.* **2020**, *1*, 331–347. [CrossRef]
22. Nkenyereye, L.; Nkenyereye, L.; Islam, S.M.R.; Kerrache, C.A.; Abdullah-Al-Wadud, M.; Alamri, A. Software Defined Network-Based Multi-Access Edge Framework for Vehicular Networks. *IEEE Access* **2020**, *8*, 4220–4234. [CrossRef]
23. Wan, S.; Lu, J.; Fan, P.; Letaief, K.B. Toward Big Data Processing in IoT: Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System. *IEEE Internet Things J.* **2020**, *7*, 5995–6009. [CrossRef]

24. Zhao, P.; Tian, H.; Qin, C.; Nie, G. Energy-Saving Offloading by Jointly Allocating Radio and Computational Resources for Mobile Edge Computing. *IEEE Access* **2017**, *5*, 11255–11268. [CrossRef]
25. Zhu, J.; Wang, J.; Huang, Y.; Fang, F.; Navaie, K.; Ding, Z. Resource Allocation for Hybrid NOMA MEC Offloading. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 4964–4977. [CrossRef]
26. Xue, J.; An, Y. Joint Task Offloading and Resource Allocation for Multi-Task Multi-Server NOMA-MEC Networks. *IEEE Access* **2021**, *9*, 16152–16163. [CrossRef]
27. Nguyen, T.D.T.; Nguyen, V.; Pham, V.N.; Huynh, L.N.T.; Hossain, M.D.; Huh, E.N. Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications. *IEEE Internet Things J.* **2021**, *8*, 1687–1701. [CrossRef]
28. Hui, H.; Zhou, C.; An, X.; Lin, F. A New Resource Allocation Mechanism for Security of Mobile Edge Computing System. *IEEE Access* **2019**, *7*, 116886–116899. [CrossRef]
29. Wang, P.; Yao, C.; Zheng, Z.; Sun, G.; Song, L. Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems. *IEEE Internet Things J.* **2019**, *6*, 2872–2884. [CrossRef]
30. Chien, H.T.; Lin, Y.D.; Lai, C.L.; Wang, C.T. End-to-End Slicing With Optimized Communication and Computing Resource Allocation in Multi-Tenant 5G Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2079–2091. [CrossRef]
31. Liu, C.F.; Bennis, M.; Debbah, M.; Poor, H.V. Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing. *IEEE Trans. Commun.* **2019**, *67*, 4132–4150. [CrossRef]
32. Hu, H.; Wang, Q.; Hu, R.Q.; Zhu, H. Mobility-Aware Offloading and Resource Allocation in a MEC-Enabled IoT Network With Energy Harvesting. *IEEE Internet Things J.* **2021**, *8*, 17541–17556. [CrossRef]
33. Shayea, I.; Ergen, M.; Kouhalvandi, L.; Alhammadi, A. Dynamic Mobility Robustness Optimization Based on Individual Weight Function for 5G Networks and Beyond In Proceedings of the 2021 24th International Symposium on Wireless Personal Multimedia Communications (WPMC), Okayama, Japan, 12–16 December 2021; pp. 1–6. [CrossRef]
34. Ouyang, T.; Zhou, Z.; Chen, X. Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2333–2345. [CrossRef]
35. Gures, E.; Shayea, I.; Alhammadi, A.; Ergen, M.; Mohamad, H. A comprehensive survey on mobility management in 5g heterogeneous networks: Architectures, challenges and solutions. *IEEE Access* **2020**, *8*, 195883–195913. [CrossRef]
36. Angjo, J.; Shayea, I.; Ergen, M.; Mohamad, H.; Alhammadi, A.; Daradkeh, Y.I. Handover Management of Drones in Future Mobile Networks: 6G Technologies. *IEEE Access* **2021**, *9*, 12803–12823. [CrossRef]
37. Wang, S.; Urgaonkar, R.; Zafer, M.; He, T.; Chan, K.; Leung, K.K. Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1272–1288. [CrossRef]
38. Wang, S.; Urgaonkar, R.; Zafer, M.; He, T.; Chan, K.; Leung, K.K. Dynamic service migration in mobile edge-clouds. In Proceedings of the 2015 IFIP Networking Conference (IFIP Networking), Toulouse, France, 20–22 May 2015; pp. 1–9. [CrossRef]
39. Ceselli, A.; Premoli, M.; Secci, S. Mobile Edge Cloud Network Design Optimization. *IEEE/ACM Trans. Netw.* **2017**, *25*, 1818–1831. [CrossRef]
40. *Mobile Edge Computing (MEC) End to End Mobility Aspects ETSI GR MEC 018 V1.1.1*; European Telecommunications Standards Institute: Valbonne, France, 2017.
41. Zhang, X.; Zhang, J.; Liu, Z.; Cui, Q.; Tao, X.; Wang, S. MDP-Based Task Offloading for Vehicular Edge Computing Under Certain and Uncertain Transition Probabilities. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3296–3309. [CrossRef]
42. Park, C.; Lee, J. Mobile Edge Computing-Enabled Heterogeneous Networks. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 1038–1051. [CrossRef]
43. Sun, Y.; Zhou, S.; Xu, J. EMM: Energy-Aware Mobility Management for Mobile Edge Computing in Ultra Dense Networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2637–2646. [CrossRef]
44. Shayea, I.; Ergen, M.; Hadri Azmi, M.; Aldirmaz Çolak, S.; Nordin, R.; Daradkeh, Y.I. Key Challenges, Drivers and Solutions for Mobility Management in 5G Networks: A Survey. *IEEE Access* **2020**, *8*, 172534–172552. [CrossRef]
45. Alhammadi, A.; Roslee, M.; Alias, M.Y.; Shayea, I.; Alraih, S.; Mohamed, K.S. Auto Tuning Self-Optimization Algorithm for Mobility Management in LTE-A and 5G HetNets. *IEEE Access* **2020**, *8*, 294–304. [CrossRef]
46. Liu, L.; Qin, X.; Zhang, Z.; Zhang, P. Joint Task Offloading and Resource Allocation for Obtaining Fresh Status Updates in Multi-Device MEC Systems. *IEEE Access* **2020**, *8*, 38248–38261. [CrossRef]
47. Zhan, W.; Luo, C.; Wang, J.; Wang, C.; Min, G.; Duan, H.; Zhu, Q. Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 5449–5465. [CrossRef]
48. Zhang, H.; Wang, R.; Liu, J. Mobility Management for Ultra-Dense Edge Computing: A Reinforcement Learning Approach. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
49. Balasubramanian, V.; Zaman, F.; Aloqaily, M.; Alrabaee, S.; Gorlatova, M.; Reisslein, M. Reinforcing the Edge: Autonomous Energy Management for Mobile Device Clouds. In Proceedings of the IEEE INFOCOM 2019–IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 44–49. [CrossRef]
50. Wu, Y.; Ni, K.; Zhang, C.; Qian, L.P.; Tsang, D.H.K. NOMA-Assisted Multi-Access Mobile Edge Computing: A Joint Optimization of Computation Offloading and Time Allocation. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12244–12258. [CrossRef]
51. El Haber, E.; Nguyen, T.M.; Assi, C. Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-Tier Edge-Clouds. *IEEE Trans. Commun.* **2019**, *67*, 3407–3421. [CrossRef]

52. Al-Maashri, A.; Ould-Khaoua, M. Performance Analysis of MANET Routing Protocols in the Presence of Self-Similar Traffic. In Proceedings of the 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 14–16 November 2006; pp. 801–807. [CrossRef]

53. Hung, S.; Zhang, X.; Festag, A.; Chen, K.; Fettweis, G. Vehicle-Centric Network Association in Heterogeneous Vehicle-to-Vehicle Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5981–5996. [CrossRef]

54. Ale, L.; Zhang, N.; Wu, H.; Chen, D.; Han, T. Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network. *IEEE Internet Things J.* **2019**, *6*, 5520–5530. [CrossRef]

55. Guan, X.; Wan, X.; Ye, F.; Choi, B. Handover Minimized Service Region Partition for Mobile Edge Computing in Wireless Metropolitan Area Networks. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 16–19 September 2018; pp. 1–6.

56. Wang, S.; Urgaonkar, R.; Chan, K.; He, T.; Zafer, M.; Leung, K.K. Dynamic service placement for mobile micro-clouds with predicted future costs. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 5504–5510. [CrossRef]

57. Park, J.; Solanki, S.; Baek, S.; Lee, I. Latency Minimization for Wireless Powered Mobile Edge Computing Networks with Nonlinear Rectifiers. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8320–8324. [CrossRef]

58. Wang, F.; Xu, J.; Wang, X.; Cui, S. Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 1784–1797. [CrossRef]

59. Goldsmith, A. *Wireless communications*; Cambridge University Press: Cambridge, UK, 2005.

60. Niu, C.; Li, Y.; Hu, R.Q.; Ye, F. Fast and Efficient Radio Resource Allocation in Dynamic Ultra-Dense Heterogeneous Networks. *IEEE Access* **2017**, *5*, 1911–1924. [CrossRef]

61. Guo, H.; Liu, J.; Zhang, J.; Sun, W.; Kato, N. Mobile-Edge Computation Offloading for Ultradense IoT Networks. *IEEE Internet Things J.* **2018**, *5*, 4977–4988. [CrossRef]

62. Lopez-Perez, D.; Guvenc, I.; Chu, X. Mobility management challenges in 3GPP heterogeneous networks. *IEEE Commun. Mag.* **2012**, *50*, 70–78. [CrossRef]

63. Mehrabi, M.; Shen, S.; Hai, Y.; Latzko, V.; Koudouridis, G.P.; Gelabert, X.; Reisslein, M.; Fitzek, F.H.P. Mobility- and Energy-Aware Cooperative Edge Offloading for Dependent Computation Tasks. *Network* **2021**, *1*, 191–214. [CrossRef]

64. Tian, X.; Zhu, J.; Xu, T.; Li, Y. Mobility-Included DNN Partition Offloading from Mobile Devices to Edge Clouds. *Sensors* **2021**, *21*, 229. [CrossRef] [PubMed]