applied sciences

MDPI

Article

Optimizing Task Offloading Energy in Multi-User Multi-UAV-Enabled Mobile Edge-Cloud Computing Systems

Soha Alhelaly [1], Ammar Muthanna [2] and Ibrahim A. Elgendy [3,*]

[1] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; s.alhelaly@seu.edu.sa
[2] Department of Telecommunication Networks and Data Transmission, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 193232 Saint Petersburg, Russia; ammarexpress@gmail.com
[3] Department of Computer Science, Faculty of Computers and Information, Menoufia University, Shibin El Kom 32511, Egypt
* Correspondence: ibrahim.elgendy@ci.menofia.edu.eg

Abstract: With the emergence of various new Internet of Things (IoT) devices and the rapid increase in the number of users, enormous services and complex applications are growing rapidly. However, these services and applications are resource-intensive and data-hungry, requiring satisfactory quality-of-service (QoS) and network coverage density guarantees in sparsely populated areas, whereas the limited battery life and computing resources of IoT devices will inevitably become insufficient. Unmanned aerial vehicle (UAV)-enabled mobile edge computing (MEC) is one of the most promising solutions that ensures the stability and expansion of the network coverage area for these applications and provides them with computational capabilities. In this paper, computation offloading and resource allocation are jointly considered for multi-user multi-UAV-enabled mobile edge-cloud computing systems. First, we propose an efficient resource allocation and computation offloading model for a multi-user multi-UAV-enabled mobile edge-cloud computing system. Our proposed system is scalable and can support increases in network traffic without performance degradation. In addition, the network deploys multi-level mobile edge computing (MEC) technology to provide the computational capabilities at the edge of the radio access network (RAN). The core network is based on software-defined networking (SDN) technology to manage network traffic. Experimental results demonstrate that the proposed model can dramatically boost the system performance of the system in terms of time and energy.

Keywords: computation offloading; resource allocation; energy efficient; unmanned aerial vehicle (UAV); mobile edge computing; optimization

check for updates

Citation: Alhelaly S. ;Muthanna A.; Elgendy, I.A. Optimizing Task Offloading Energy in Multi-User Multi-UAV-Enabled Mobile Edge-Cloud Computing Systems. *Appl. Sci.* **2022**, *12*, 6566. https://doi.org/10.3390/app12136566

Academic Editor: Fabrizio Granelli

Received: 25 May 2022
Accepted: 27 June 2022
Published: 28 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

By 2025, market revenue from unmanned aerial vehicles (UAVs) is expected to reach 52 billion dollars, a dramatic increase compared with today's UAV market [1,2]. By then, UAVs will improve commercial, civil, and government sectors by introducing a wide range of heterogeneous applications [3,4]. These applications will vary from simple environmental monitoring to complex high-security military applications. Based on release 14 of the 3GPP and the technical reports of the ITU-R, UAVs are considered to be among the most important facets of the fifth-generation cellular system (5G/IMT2020), as well as beyond 5G and 6G networks [5–8]. UAVs will be actively used for various purposes, such as military tasks, remote sensing, remote localization, target detection, transportation monitoring, and route assessment [9–11].

Several related system architectures based on UAV-mobile edge computing (MEC) have recently been proposed, in which intensive computation tasks are offloaded from the mobile user to a richer resource located at the radio access network (RAN) edge [12–16]. Most of these architectures address only two layers of computation offloading in UAV-MEC systems [17,18],

although a few studies address multi-layer computation offloading [19]. Most of these architectures share the goal of minimizing energy consumption, allocating radio and computation resources efficiently, and/or meeting mobile user delay requirements.

However, in order to support the network in which UAVs operate, several features such as low power consumption and information processing must be considered. In addition, in a large-scale environment, the connected servers at the edge network are limited by computational capacity, which leads to high delays and makes these devices unsuitable for real-time tasks. Moreover, deriving the best offloading decision in a multi-user multi-UAV wireless environment is considered a non-trivial problem.

Confronting these challenges, we believe that constructing an energy-efficient model for a multi-user multi-UAV-enabled mobile edge-cloud computing system would be useful. Therefore, we proposed an efficient task offloading and allocation of resources model for a multi-user with multi-UAV-enabled mobile edge-cloud computing system. In addition, a combination model of resource allocation and offloading is formulated as a problem which aims to minimize the energy consumption of the whole system with a latency constraint. Moreover, a multi-UAV energy-efficient task offloading algorithm is designed to derive the task offloading decision for all device users. The main contributions of this paper include:

- An energy-efficient system model for multi-user multi-UAV-enabled mobile edge-cloud computing. This system is a scalable system and can support increased network traffic without performance degradation;
- A network of UAVs is deployed to cover the dead and high-density network areas. In addition, UAVs can provide other applications over the smart city network in which the multilevel MEC technology is deployed to provide both the computational capabilities at the edge network as well as a core network based on SDN technology to manage the network traffic and provide an innovative interface to the network operators;
- A combination model of resource allocation and offloading is formulated as a problem which aims to minimize the energy consumption of the whole system with a latency constraint;
- Simulations validate our proposed model and demonstrate that the system's energy consumption can be significantly decreased.

The remainder of this paper is organized as follows. Related work on computation offloading approaches is presented in Section 2. In Section 3, the system model is presented, followed by the optimization problem formulation. Then, experimental simulations are evaluated in Section 4 to demonstrate the performance improvement of our system model. Finally, the paper's conclusion is deduced in Section 5.

2. Related Work

In recent years, several system architectures and optimization models based on UAV-MEC have been developed to address the mobile users' main challenges where the computation offloading concept is applied. Most of these approaches have only considered MEC networks with single [20,21] or multiple edge servers [22,23], although a few approaches have considered MEC networks with multi-edge servers and a centralized cloud [16,19,24]. This section provided a brief overview of popular approaches based on MEC networks with single and multi-edge servers with and without a centralized cloud.

2.1. Single Edge Server

You et al. [20] proposed an energy-efficient framework for wirelessly powered mobile cloud computing that integrates microwave power transfer (MPT) with mobile cloud computing. This framework considers only a single-user, single-edge system in which the mobile user can harvest energy from the edge or offload computation to it. Additionally, an optimization problem is formulated as a non-convex formula that aims to minimize the mobile user's energy consumption for local computation harvesting while maximizing the energy savings from mobile users for computation offloading. Meanwhile, an efficient and online deep reinforcement learning-based offloading framework is proposed in [21].

Specifically, a deep neural network is used to learn and determine the offloading decision for mobile users such that they can either execute the computation tasks locally using mobile device resources or offload them to the edge for remote execution. In addition, an adaptive procedure is designed to reduce the complexity by automatically adjusting the hyper-parameters of the deep model. Finally, the numerical and experimental results show that this framework can provide a near-optimal solution.

Recently, an efficient algorithm based on deep meta reinforcement learning was proposed in [25] for IoT-edge-cloud computing systems with the goal of reducing the burden of computing and improving the processing of tasks. Specifically, a different number of DNNs are combined with Q-learning to derive an efficient offloading decision and increase the learning ability of the system. Furthermore, the results proved that the proposed algorithm outperforms binary and partial offloading schemes. Meanwhile, in [26], quality of experience (QoE) is optimized in an edge-enabled Internet of Things environment in which transmission and computing delay, the success rate of each task, and energy consumption are considered. In addition, deep reinforcement learning is utilized to solve the proposed model, where double Q-learning and dueling networks-based algorithms are designed to accelerate the convergence and enhance the stability. Similarly, the computation capability limitations of IoT devices and energy consumption are addressed in [27], where an integrated framework is proposed for task offloading and allocating resources with the goal of maximizing energy efficiency while reducing the quality of service (QoS) constraints. However, one of the common drawbacks of [27] is that the transferred data is susceptible to different attacks.

2.2. Multiple Edge Servers

Minimizing the mobile devices' energy consumption and reducing the mobile applications' execution latency in a multi-user system are the main goals of [22], in which the computational speed and the power of transmission for mobile devices are jointly optimized with the offloading ratio. Specifically, the authors formulated energy consumption and latency execution as non-convex problems. Then, they used variable substitution and univariate search techniques to obtain the optimal solution. Their results demonstrate that the proposed model can significantly decrease energy consumption and shorten latency with respect to the existing offloading schemes. A multi-user with a multi-edge offloading model for static and dynamic channels is proposed in [23]. First, on the static channel, the computation offloading decision is formulated as a non-cooperative potential game in which each mobile device can selfishly maximize CPU cycles and minimize consumed energy. On the dynamic channel (i.e., under time-varying and unknown channel state information), a Q-learning-based distributed computation offloading algorithm is formulated to determine the optimal solution, where the Nash equilibrium can be achieved. Finally, numerical results indicated that not only can this model outperform the energy consumption performance over local processing and random assignment, but it can also improve average long-term payoffs by up to 87.87% in comparison with the perfect channel state information case.

Yang et al. [28] proposed an energy-efficient offloading technique for multitask unloading of data based on UAV clusters for virtual and augmented reality games on vehicle networks. The VR/AR hybrid gaming scene should allow drivers and passengers to enjoy various visual effects in real time. However, due to dynamic changes in the real-time distribution of users, fixed edge nodes cannot cope with the changing load. Therefore, the use of UAVs together with fixed nodes as edge nodes helps to cope with the computing requirements of automotive networks. The deployment of a large number of UAVs is not practical due to mutual interference and the high cost of UAVs. To solve these problems, the paper presents a system with the following characteristics: multitask unloading (UAVs can work on several tasks and the results of one task can be used to perform other tasks in the future), UAV clusters working on one task, computing optimization, caching and communication resources, and AI-based decision making. In addition, an architecture for UAV clusters with various task implementations is proposed in which all the large-

scale applications for VR/AR games of vehicle networks are considered. Furthermore, an AI-based decision-making system is proposed to facilitate collaboration between UAVs and joint computing optimization, caching, and communication resources. The proposed architecture, compared with traditional MEC structures (both with and without UAVs), has advantages in the amount of information collected, real time performance, decision-making ability, and efficiency. Two algorithms were considered: preliminary deployment based on the analysis of historical data, and real-time deployment based on real-time perception. In addition, in this work, an experiment was conducted to predict the load of several UAVs based on long short-term memory (LSTM).

Elsewhere, ref. [29] presented a multi-server task offloading model in a decentralized game manner with the goal of maximizing the system utility; a decentralized and efficient algorithm is proposed to derive the optimal offloading where the differential neural computing approaches and deep reinforcement learning models are combined without the need of prior knowledge of the user preference and network bandwidth. Meanwhile, in [30], unmanned ground vehicles (UGVs) are introduced as an effective solution to address the computing resources and battery limitations of UAVs, in which the intensive tasks can be offloaded from UAVs to be executed with UGVs. Moreover, a new stable matching algorithm is designed to transform the offloading problem into an equivalent two-sided matching form. Further, an iterative algorithm is introduced to match UAVs with the appropriate UGVs. Nevertheless, despite improvements recorded in [29,30], data transferred is not adequately protected from different types of attacks during transmission.

2.3. Multiple Edge Servers and Remote Cloud

Beiqing et al. [19] proposed an energy-efficient edge-cloud framework for computation offloading of a multi-UAV system. They formulated an optimization problem as a three-layer game in which the offloading decision could be determined in a decentralized way. In addition, efficient decentralized algorithms are developed in which the performance of the game is analyzed through its efficiency ratio and the Nash equilibrium can be achieved. Finally, experimental results demonstrate that these algorithms can dramatically decrease the consumption of energy by 30% with less than 10% performance loss compared with other benchmark solutions. Liu et al. [24] presented an energy-efficient cloud-assisted edge computing framework for collaborative task computation offloading in IoT sensors. This framework was composed of a three-tier network with IoT sensors in the first tier and the edge server and the cloud server in the other two tiers. In addition, an offloading strategy was formulated as a mixed-integer problem in which minimizing the energy consumption of IoT sensors was the main goal. Furthermore, a semi-definite relaxation-based energy-efficient collaborative task offloading algorithm was developed to solve this problem and to obtain the optimum decision of offloading. Finally, the simulation results showed that this algorithm can outperform the existing algorithm in terms of energy consumption. In addition, it can efficiently adapt to different system parameters.

The standard MEC structure with fixed nodes is not able to satisfy the needs of users for computing resources and connections with rapidly changing characteristics of time, space, and spectrum. To solve this problem, ref. [31] discusses the use of an air-to-ground MEC-integrated environment, consisting of the deployment of UAVs and ground vehicles as additional edge nodes. The UAVs make it possible to ensure coverage of the network in complex geographical areas, and provide a wider selection of services. The proposed network consists of three levels: the level of cloud computing, the level of edge computing, and the level of devices. The integrated structure basically assumes the use of the main advantages provided by software-defined networking: centralization of control, separation of user data from control data, and a large abstraction of basic structures. The hypervisor in this network is responsible for the virtualization of the resources of user devices, as well as for the computing and storage resources of the upper level (relative to the device level). Variants of the proposed network structure correspond to the main scenarios in networks: support for ultra-low latency—URLLC, distributed processing and analysis of big data along with the caching of content and delivery to mobile devices—eMBB, and scalable

arrays and mobile communications management—mMTC. The benefits provided by the structure include adaptive resource allocation, differentiated QoS guarantees, enhanced connectivity between network nodes, and efficient mobility management.

The delay in transferring data from mobile devices to a remote computing cloud has a large load value in existing traffic offloading systems. To solve this problem, ref. [32] developed code for a traffic offloading system oriented to the network edge called Echo. The network structure for the Echo code consists of three levels: mobile devices, border, and cloud. Echo uses a centralized mechanism in which the decision to offload traffic is carried out at the edge nodes. The code for unloading traffic is implemented on Android devices. Echo solves the problem of which method and which platform are used—the cloud platform or the border. For more efficient use of the resources of the edge nodes, the Preemption-Constrained Shortest-Remaining Time-First (PC-SRTF) scheduling algorithm is used to minimize the average time for solving the problem. The proposed code provides a guarantee of the quality of service when unloading traffic at border nodes. To optimize the data transfer, lazy object transfer algorithms and differential object update algorithms are used. In the first algorithm, when traffic is unloaded, a proxy is created to transfer related objects. In the second method, only the parts of objects that differ when loading several methods on a mobile device are updated. The proposed Echo code was found to have lower average task execution time and power consumption for mobile devices compared to existing traffic offloading systems.

Recently, a multi-user with multi-UAV-enabled and cloud servers' model is presented in [33,34]. Specifically, in [33], a new weighted cost model is proposed with the goal of minimizing the energy consumption and execution time. Additionally, an efficient placement algorithm is developed to assign the tasks to the most suitable server in an efficient manner. Moreover, a lightweight failure recovery algorithm is designed to optimize the parallel execution for tasks. Xia et al. [34], however, proposed an intelligent task offloading model for a multi-tier MEC network, in which minimizing the execution delay of tasks and meeting the UAVs' computing capacity are the main goals. In addition, an equivalent and convex form of the formulated problem is given using two-layer optimization technique. In the first layer, differential evolution learning approach is utilized, while a distributed deep neural network is used in the other layer. However, measuring the execution time is not considered in [34], which is one of the major downsides.

Due to the high dynamics in urban networks, which lead to an overload of the connection between the UAV and the boundary server (which also leads to an increase in the time spent on tasks), and due to the large queues that occur on the boundary servers when processing several tasks, the total time of n-th task transfer to the boundary server and the task processing time may exceed the time spent analyzing the task locally on the UAV. Here, we present an optimization process that allows the user to choose where the task will be processed on a UAV. We consider the communication, computational, and energy aspects of the proposed process.

Table 1 introduces a summary of the related works reviewed in terms of number of users, edge, and remote cloud servers, and highlights the main weaknesses of the studies.

Table 1. Overview of studies on MEC.

Reference	Objective	Proposed Solution	User		UAV Server		Remote Cloud	Evaluation Methods	Weakness
			Single	Multiple	Single	Multiple			
[20]	Minimize energy consumption	An energy-efficient framework for wireless-powered mobile cloud computing	✓		✓		✓	Simulation	Considering only one IoT user in the model is unrealistic.
[21]	Maximize weighted sum of computation rate	An online deep reinforcement learning framework for mobile edge computing		✓	✓			Numerical results, simulation	Execution time is not considered.
[25]	Minimize time and energy	An efficient algorithm based on deep meta reinforcement learning		✓	✓		✓	Simulation	Deadline for computation tasks is not considered
[26]	Optimize quality of experience	A new quality of experience model		✓	✓			Simulation	Resource cooperation is not considered.
[27]	Maximize energy efficiency under minimum quality of service constraint	An integrated model is proposed for task offloading and allocating resources		✓	✓		✓	Simulation	Private data protection and the resource cooperation are not addressed.
[22]	Minimize energy and execution latency	A partial task offloading model	✓			✓		Simulation	Deadline for computation tasks is not considered and considering only one IoT user in the model is unrealistic.
[23]	Reduce energy consumption	A multi-user with a multi-edge task offloading model	✓			✓		Simulation	Considering only one IoT user in the model is unrealistic.
[28]		An energy-efficient task offloading technique		✓		✓		Numerical results, simulation	Deadline for computation tasks and data privacy are not considered.
[29]	Maximize system utility	A multi-server task offloading model		✓		✓		Simulation	Data privacy is not addressed.
[30]	Maximize system utility	A task offloading model for post-disaster rescue		✓		✓		Simulation	Data privacy is not addressed.
[19]	Minimize the overall energy consumption	An energy-efficient edge-cloud architecture for intelligent multi-UAV		✓		✓	✓	Simulation	Resource cooperation is not considered.
[24]	Minimize energy consumption	An energy-efficient collaborative task offloading model for cloud-enabled edge computing network		✓		✓	✓	Simulation	Data privacy is not addressed.
[31]	Minimize energy consumption	An air–ground integration technique for MEC		✓		✓	✓	Numerical results, simulation	Deadline for computation tasks and data privacy are not considered.
[32]		An edge-centric code offloading model		✓		✓	✓	Simulation	Data privacy is not addressed.
[33]	Minimize the energy consumption and execution time	A new weighted cost model for multi-UAV environment		✓		✓	✓	Simulation	Task execution's deadline is not considered.
[34]	Minimize the processing delay of tasks	An intelligent task offloading model		✓		✓	✓	Numerical results, simulation	Execution time is not measured.
Our work	Minimize energy consumption with delay satisfaction	An efficient resource allocation and task offloading model		✓		✓	✓	Simulation	Mobility and security are not addressed.

3. System Model

As shown in Figure 1, we present a multi-user with multi-UAV-enabled MEC system model. In this system, we consider $\mathcal{M}$ as a set of small UAVs that are equipped with an edge computing server and connected to centralized cloud computing through a backbone router. In addition, we consider $\mathcal{N}$ as a set of mobile devices that are associated with UAVs via a wireless channel. Furthermore, each mobile device is involved in highly intensive computation tasks that need to be processed locally or be transmitted and processed by the nearby UAV or cloud server via the wireless channel. In our simulation settings, we assumed a quasi-static scenario where the UAV does not move during the computation offloading period, though it can be moved to another location across different periods [35–37].

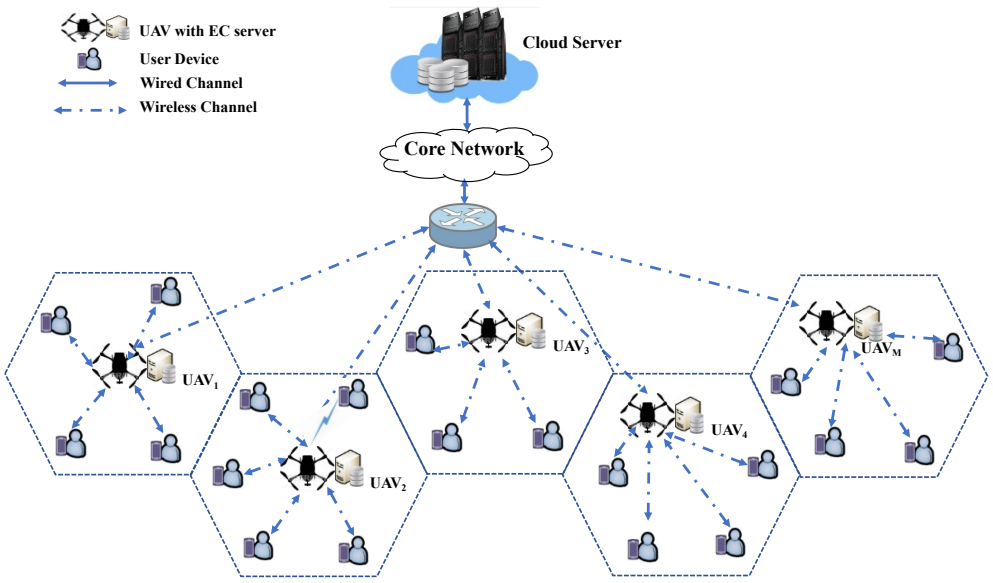


Figure 1. System model.

In the next subsections, the communication and computation models are firstly discussed in more detail. Afterward, our problem formulation is presented.

Table 2 summarizes the main notations of this study.

Table 2. Notations.

Notation	Description
$\mathcal{M}$	The set of small UAVs.
$\mathcal{N}$	The set of device users.
i	The *ith* UAV.
j	The *jth* device user.
$b_{i,j}$	Size of data for task of mobile device j which is connected with UAV i.
$c_{i,j}$	CPU cycles to accomplish the task of mobile device j which is connected with UAV i.
$\tau_{i,j}$	Deadline requirement for task of mobile device j which is connected with UAV i.
$\varphi_{i,j,a}$	Offloading decision of mobile device.
$r_{i,j}$	Uplink data rate for mobile device j which is connected with UAV i.
B_i	System bandwidth of UAV i
$p_{i,j}$	Transmission power of mobile device j which is connected with UAV i.
G_i	Channel gain of UAV i

Table 2. *Cont.*

Notation	Description
ω_i	Density of noise power of channel which is connected with UAV i
$\zeta_{i,j}$	Energy consumed per CPU cycle of mobile device j which is connected with UAV i.
δ	Propagation delay between UAV and cloud
$f^l_{i,j}$	Computational capability of mobile device j which is connected with UAV i.
$f^e_{i,j}$	CPU resource of UAV i assigned to mobile device j
$f^c_{i,j}$	CPU resource of cloud assigned to mobile device j which is connected with UAV i.
$T^l_{i,j}$	Time for local execution for the task of mobile device j which is connected with UAV i.
$E^l_{i,j}$	Energy for local executing task of mobile device j which is connected with UAV i
$T^{off}_{i,j}$	Time for transmitting task of mobile device j to UAV i.
$T^{e_exec}_{i,j}$	Time for executing task of mobile device j at UAV i.
$T^{c_exec}_{i,j}$	Time for executing task of mobile device j at Cloud server.

3.1. Communication Model

We first present the model of communication for edge-cloud systems, where the environment has a number $\mathcal{M}$ of small UAVs that are connected to the centralized cloud computing via a backbone router. In addition, each UAV is connected with a number $\mathcal{N}$ of mobile devices that have highly intensive computation tasks that should be completed. We refer to the small UAVs' set and the mobile devices' set as $\mathcal{M} = \{1, 2, \ldots, M\}$, and $\mathcal{N} = \{1, 2, \ldots, N\}$, respectively.

We denote $\varphi_{i,j,a}$ as the decision offloading where the task of mobile device j, which is connected with UAV i, is assigned to be executed at the server a, which can be a local execution (i.e., a = 0), cloud execution (i.e., $a = M + 1$), or UAVs execution (i.e., $a \in [1..M]$). In particular, ($\varphi_{i,j,a} = 1$, where $a \in [1..M]$) denotes that the computation task of device j, which is connected to UAV i, will be transmitted and executed remotely at one of the available UAVs, while ($\varphi_{i,j,0} = 1$) and ($\varphi_{i,j,M+1} = 1$) denote that the task of mobile device j will be processed locally by itself or on the cloud server, respectively. Overall, each mobile device j must execute its computation task only one time, whether locally ($a = 0$) or remotely ($a \in \{1, 2, \ldots, M + 1\}$), i.e., $\sum_{a=0}^{M+1} \varphi_{i,j,a} = 1$.

Following Shannon's law, the maximum uplink data rate between each mobile device j and the connected UAV to transmit its task's data through the communications channel can be expressed as [38]:

$$r_{i,j} = B_i log_2(1 + \frac{p_{i,j} g_i^2}{\omega_i B_i}), \quad (1)$$

where B_i denotes the channel bandwidth between UAV i and its connected mobile device, $p_{i,j}$, and ω_i and g_i denote the transmission power of mobile devcie j, the power density, and the gain of the channel between the mobile device and UAV. Guided by the intuition in [39], TDMA technology is utilized to mitigate the co-channel interference between device users.

In this paper, the total overhead consumption in terms of energy and time for returning the output data size is neglected due to its size being smaller than the input data size. The link data rate for downloading result from server is larger than the uploading [40–42].

3.2. Computation Model

This subsection will introduce the offloading model where the simulation has a number $\mathcal{M}$ of small UAVs that are connected to the centralized cloud computing via a backbone router. In addition, each UAV is connected to a number $\mathcal{N}$ of mobile devices that have highly intensive computation tasks that should be finished. For each mobile device j, the tuple $\{b_{i,j}, c_{i,j}, \tau_{i,j}\}$ represents the task requirement, where $b_{i,j}$ indicates the size of the dataset that needs to be transmitted from mobile device j to UAV i, whereas $c_{i,j}$ and $\tau_{i,j}$ denote the CPU cycles' number and the required deadline for the task of mobile device j. The values of $b_{i,j}$ and $c_{i,j}$ can be acquired via task execution profiling [40,43].

3.2.1. Local Execution

For the local execution method, the computation task of mobile device j will be executed locally and the total time execution and consumption of energy can be expressed as, respectively:

$$T_{i,j}^{l} = \frac{c_{i,j}}{f_{i,j}^{l}} \tag{2}$$

$$E_{i,j}^{l} = \zeta_{i,j} c_{i,j}, \tag{3}$$

where $f_{i,j}^{l}$ indicates the capability of computation (CPU/cycles per second) of the mobile device, and $\zeta_{i,j}$ is a coefficient that indicates the CPU cycle's energy consumed for mobile device j, which is connected with UAV i. We set $\zeta_{i,j} = 10^{-11}(f_{i,j}^{l})^2$, in which the consumption of energy is a superlinear function of the mobile device's frequency [40,44].

3.2.2. Remote Execution

For the remote execution method, the computation task of mobile device j, which is connected with UAV i, will be transmitted and executed on one of the UAVs or the cloud server, and the total execution time can be expressed as:

$$T_{i,j}^{e} = T_{i,j}^{off} + T_{i,j}^{e_exec} \tag{4}$$

$$T_{i,j}^{c} = T_{i,j}^{off} + \delta + T_{i,j}^{c_exec}, \tag{5}$$

where δ is a constant referring to the delay of propagation for transmitting the task between the UAV and the cloud. $T_{i,j}^{off}$, $T_{i,j}^{e_exec}$, and $T_{i,j}^{c_exec}$ indicate the transmitting and processing time for executing the task of mobile device j at the UAV and cloud server, respectively, which can be calculated as follows:

$$T_{i,j}^{off} = \frac{b_{i,j}}{r_{i,j}} \tag{6}$$

$$T_{i,j}^{e_exec} = \frac{c_{i,j}}{f_{i,j}^{e}} \tag{7}$$

$$T_{i,j}^{c_exec} = \frac{c_{i,j}}{f_{i,j}^{c}}, \tag{8}$$

where $f_{i,j}^{e}$ and $f_{i,j}^{c}$ denote the capability of computation for the UAV and the cloud server that is allocated to mobile device j. As a consequence, the consumption of energy for transmitting and processing the task of mobile device j remotely at the UAV and cloud server can be calculated as follows:

$$E_{i,j}^{r} = p_{i,j} T_{i,j}^{off}, \tag{9}$$

In view of the above models, the energy and time overhead for processing the computation task of mobile device j can be expressed as, respectively:

$$E_{i,j} = \left[\varphi_{i,j,0} E_{i,j}^{l} + \sum_{a=1}^{M+1} \varphi_{i,j,a} E_{i,j}^{r} \right], \tag{10}$$

$$T_{i,j} = \left[\varphi_{i,j,0} T_{i,j}^{l} + \varphi_{i,j,M+1} T_{i}^{c} + \sum_{a=1}^{M} \varphi_{i,j,a} T_{i,j}^{e} \right] \tag{11}$$

3.3. Problem Formulation

This subsection will present the formulation of our optimization problem for which computation offloading and allocation of resources for a multi-user and multi-UAV-enabled mobile edge-cloud computing system are considered. The main issue of this problem is to decrease the consumption of energy for the system with a latency constraint. This formulation is as follows:

$$\begin{aligned}
\min_{\varphi} \quad & \Big[\sum_{i=1}^{M}\sum_{j=1}^{N} E_{i,j}\Big] & & \\
\text{s.t} \quad & \Big[E_{i,j} - E^{l}_{i,j}\Big] \leq 0, & \forall_{i,j} \quad & C1 \\
& T_{i,j} \leq \tau_{i,j}, & \forall_{i,j} \quad & C2 \\
& \sum_{j=1}^{N}\sum_{a=1}^{M+1} \varphi_{i,j,a} r_{i,j} \leq B_i, & \forall_i \in [1..M] \quad & C3 \\
& \sum_{j=1}^{N}\sum_{a=1}^{M} \varphi_{i,j,a} f^{e}_{i,j} \leq F_i, & \forall_i \in [1..M] \quad & C4 \\
& \sum_{a=0}^{M+1} \varphi_{i,j,a} = 1, & \forall_i \quad & C5 \\
& \varphi_{i,j,a} \in \{0,1\}, & \forall_{i,j}, \quad & C6
\end{aligned} \tag{12}$$

Constraint $C1$ is the upper bound of energy consumption. Constraint $C2$ controls the requirement deadline for each task. Constraint $C3$ manages the capacity of bandwidth. Constraint $C4$ describes the edge server's capacity upper limit. Constraint $C5$ ensures that each task must be processed only one time. Lastly, constraint $C6$ ensures that the variable of decision offloading is binary.

The offloading decision can be derived by solving this problem. However, this problem is classified as NP-hard, in which the offloading decision variables are binary [45]. Therefore, to solve this problem in an efficient manner, a binary relaxation approach is utilized. Specifically, the binary variables are relaxed to new real variables, such as $0 \geq \varphi_{i,j,a} \leq 1$ [46]. Therefore, the formulation of the problem after relaxing the binary variables is shown in Equation (13).

$$\begin{aligned}
\min_{\varphi} \quad & \Big[\sum_{i=1}^{M}\sum_{j=1}^{N} E_{i,j}\Big] & & \\
\text{s.t} \quad & \Big[E_{i,j} - E^{l}_{i,j}\Big] \leq 0, & \forall_{i,j} \quad & C1 \\
& T_{i,j} \leq \tau_{i,j}, & \forall_{i,j} \quad & C2 \\
& \sum_{j=1}^{N}\sum_{a=1}^{M+1} \varphi_{i,j,a} r_{i,j} \leq B_i, & \forall_i \in [1..M] \quad & C3 \\
& \sum_{j=1}^{N}\sum_{a=1}^{M} \varphi_{i,j,a} f^{e}_{i,j} \leq F_i, & \forall_i \in [1..M] \quad & C4 \\
& \sum_{a=0}^{M+1} \varphi_{i,j,a} = 1, & \forall_i \quad & C5 \\
& \varphi_{i,j,a} \in [0,1], & \forall_{i,j}, \quad & C6
\end{aligned} \tag{13}$$

Since the problem is considered as a linear problem, where the objective function and the constraints are linear, the near-optimal solution can be derived using well-studied linear approaches.

3.4. A Multi-Tier Energy-Efficient Task Offloading Algorithm

In this section, a multi-tier energy-efficient task offloading algorithm is presented in which the derivation of the near-optimum task offloading decision for multi-tier environments is outlined as follows (Algorithm 1).

Algorithm 1 Multi-tier energy-efficient computation offloading algorithm.

1: **Initialization**: Each device user j initializes the offloading decision of its' task with $\varphi_{i,j,0} = 1, \forall i, j$
2: **for each** UAV i and at given time slot t **do**
3: **Upload** all the available computational capabilities and bandwidth to the backbone router.
4: **for each** device user j **do**
5: **Upload** the required information for their task, $\{b_{i,j}, ci, j, \tau_{i,j}, \zeta_{i,j}, p_{i,j}\}$, as well as their local capabilities $f^l_{i,j}$ to the backbone router.
6: **end for**
7: **end for**
8: **Compute** the available data rate $r_{i,j}$ for each device user regrading Equation (1).
9: **Solve** the formulated problem in Equation (12) and derive the offloading decision values $\varphi_{i,j,a}$ for each task.
10: **Send** the offloading decision values $\alpha_{i,j,k}$ to each device user.

First, the offloading decision for all computation tasks of mobile devices are initialized with $\varphi_{i,j,0} = 1$, which indicates the local execution. Then, the backbone router iterates over UAVs and gathers their available capabilities including storage and computational as well as the available bandwidth. Next, it receives the required information for the user's task, including $\{b_{i,j}, ci, j, \tau_{i,j}, \zeta_{i,j}, p_{i,j}\}$, as well as the local capabilities for each device $f^l_{i,j}$, through the connected UAV. Further, the maximum allowable data rate for each mobile device is calculated according to Equation (1). Afterwards, the task offloading decision values for each computation task are calculated and derived by solving the formulated problem in Equation (12), and then said values for each device user are calculated, in which decreasing the consumption of energy for the system with a latency constraint is the main goal.

Algorithm 1 describes the comprehensive steps of task offloading decision making, in which the computational complexity is $\mathcal{O}(MN)$, where M denotes the number of UAVs and N denotes the total number mobile devices.

4. Simulation Results and Discussion

Firstly, the experimental setup will be introduced. Then, the simulation results will be discussed to assess the accomplishment of our model. Finally, the traffic management algorithm is presented.

4.1. Experimental Setup

Our simulation was conducted using a personal computer equipped with an Intel® Core(TM) i7-10750H CPU with a 2.6 GHz frequency and 16 GB RAM capacity running on the Windows 10 Home 64-bit platform and preinstalled with Python for development. A multi-user with a multi-UAV environment is considered in which an area of 100 × 100 m^2 with five UAVs cooperating and associated with a single cloud server through a core network that is also equipped with an edge computing server is assumed. The height for the UAVs is set at 20 m. In addition, we have 30 mobile device users distributed across UAVs, and each user device has an intensive task that needs to be completed. The size of data is distributed uniformly within a range of [0, 10] MB, whereas the CPU cycles required to process each data bit are set to 1000 cycles/bit. The cloud server's CPU computational capabilities are set to 20 GHz and for UAV servers is distributed uniformly within a range of [3.5, 5] GHz, whereas the CPU capability of computation for each device is distributed

uniformly within a range of $\{0.5, 0.6, \ldots, 1.0\}$. The energy consumption for each mobile device is uniformly distributed within a range of (0.20×10^{-11}) J/cycle [47]. The remaining settings are summarized in Table 3. The optimization is solved using the GEKKO Python-based package [48].

Table 3. Simulation parameters.

Parameter	Value
Number of UAVs	5
Number of mobile devices	30
Height of UAVs	20 m
System bandwidth	20 MHz
Transmission power	0.1 Watt
Input data size	[0, 10] MB Unif-Dist
CPU cycles to accomplish task	1000 cycles/bit
Computation capability of mobile device	$\{0.5, 0.6, \ldots, 1.0\}$
Energy consumption	$(0, 20 \times 10^{-11})$J/cycle
UAVs server capability	[3.5, 5] Unif-Dist GHz
Cloud server capability	20 GHz
Cloud propagation delay	15 ms

4.2. Experimental Results & Discussion

This subsection evaluates the performance result of our model in comparison with four scenarios:

- **Local Execution:** No offloading exists. The computation tasks of all mobile devices will be processed locally $\varphi_{i,j,0} = 1$;
- **UAVs Server Execution:** The computation tasks of all mobile devices will be processed remotely at UAVs servers $\sum_{a=1}^{M} \varphi_{i,j,a} = 1$;
- **Cloud Execution:** The computation tasks of all mobile devices will be processed remotely at the cloud server $\varphi_{i,j,M+1} = 1$;
- **Computation Offloading Execution in [39]:** The computation tasks of all mobile devices will be processed at one of the available servers, including the local execution based on the model proposed in [39].

First, the consumption of energy for executing the tasks for the mentioned scenarios in relation to different users is shown in Figure 2. As shown, our proposed model achieves the best result compared to the other scenarios. Additionally, the edge and cloud execution scenarios exceed the scenario of local execution as the number of users increases. Moreover, the edge execution policy also exceeds the cloud execution with more than 140 device users. This result is due to increases in communication time when the communication channels are shared between the users as well as the higher energy consumption during the offloading process. In addition, the UAV's server resources are not enough to handle all the device users where the cloud server is more resourceful, thereby becoming better than UAV's execution for a large number of users. Similarly, Figure 3 shows the response time for executing the computation tasks for different numbers of users. This figure shows that the response time for our proposed model is close to or equal to the response time for UAVs, cloud, and model executions in [39], while being less than the local execution policy. However, as the number of mobile users grows, the performance of UAVs and cloud execution policies improves in comparison to local execution, while our model still outperforms the local policy and outperforms the model in [39]. Further, the cloud execution policy outperforms the UAV's server execution. This is traced to the communication channels shared among users being overloaded, thereby increasing the time, whereas the proposed model can be intelligently adapted to environmental change.

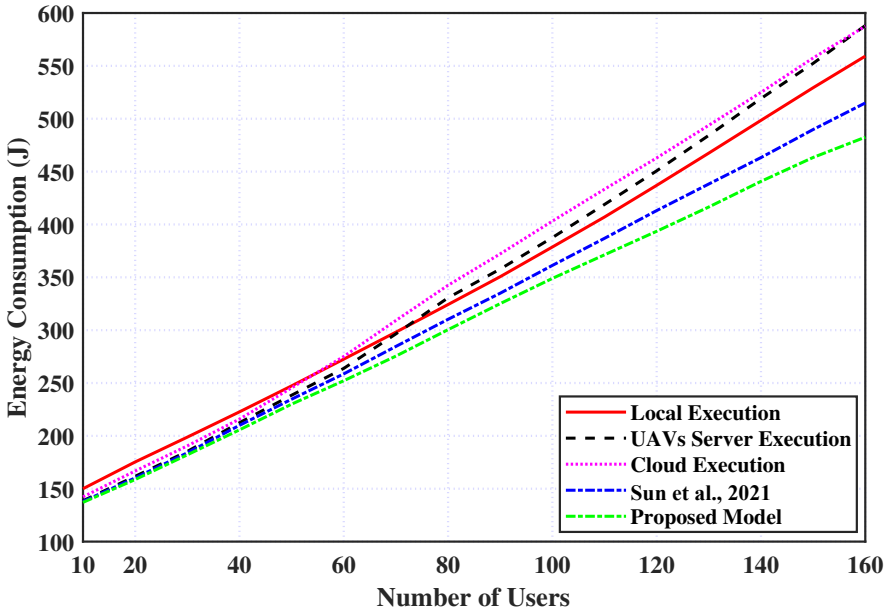


Figure 2. Energy consumption in relation to the different number of users.

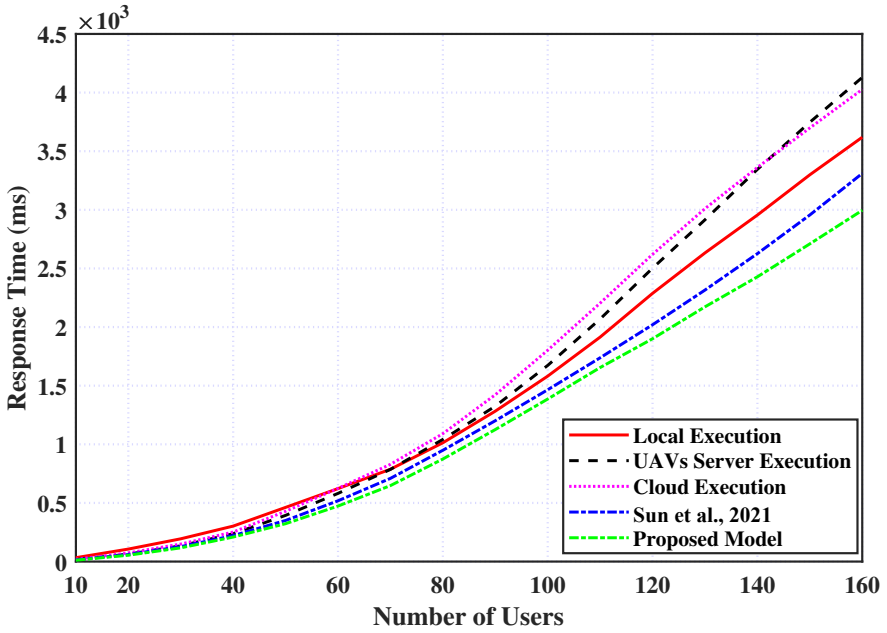


Figure 3. Response time in relation to the different number of users.

Second, the consumption of energy for executing the tasks for the mentioned scenarios in relation to different data sizes is shown in Figure 4. From this plot, we can deduce that the energy consumption of our proposed model, for data sizes of less than 20 MB, is close to the energy consumption of the UAVs, cloud, and model in [39] scenarios, whereas it is less than the energy consumption of the local execution scenario. Nevertheless, with the increase in data size, the UAVs server and cloud scenarios' performance declines compared with the scenario of local execution, whereas the performance of the proposed model remains lower than that of local policies and the policy of the model in [39]. Moreover, the cloud execution policy outperforms UAV's execution policy with large data sizes (i.e., more than 45 MB). Furthermore, the response time for processing the computation tasks with respect to different data sizes is shown in Figure 5. This figure shows that the time gradually increases as the data size increases. In addition, there is a significant gap between

our model and the other policies, which increases as the data size increases. In addition, the UAV's server executions policy nearly reaches the cloud execution at a data size of 45 MB, and will then exceed this for larger data sizes. This variation is explained as follows. Increasing data sizes cause an increase in communication time, which then results in increased energy consumption. Nevertheless, our model can be adapted to process the tasks at best locations, UAVs, cloud, or locally, thereby optimizing the energy and time.

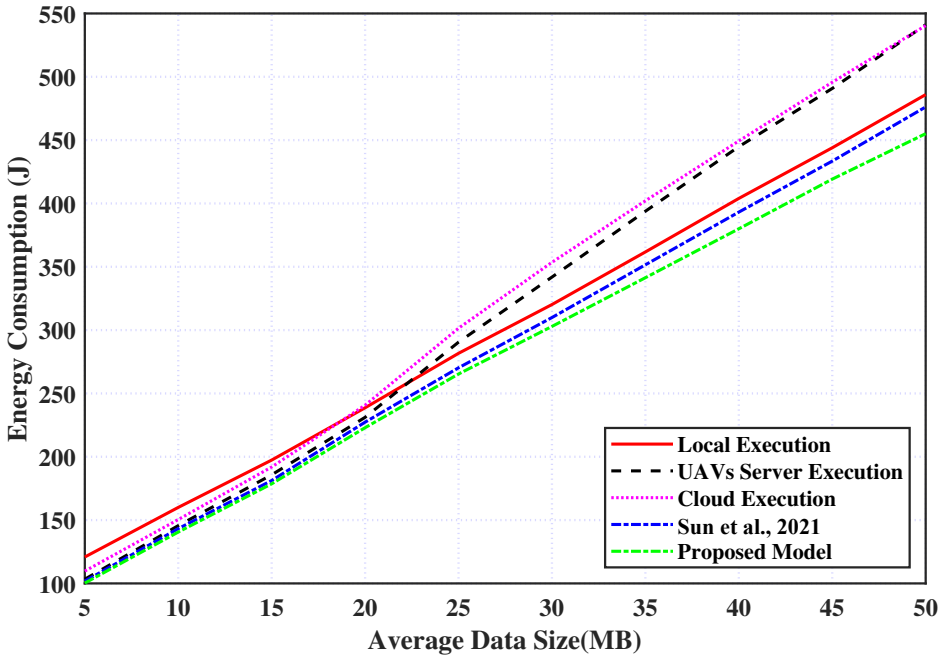


Figure 4. Energy consumption in relation to different values of input data size.

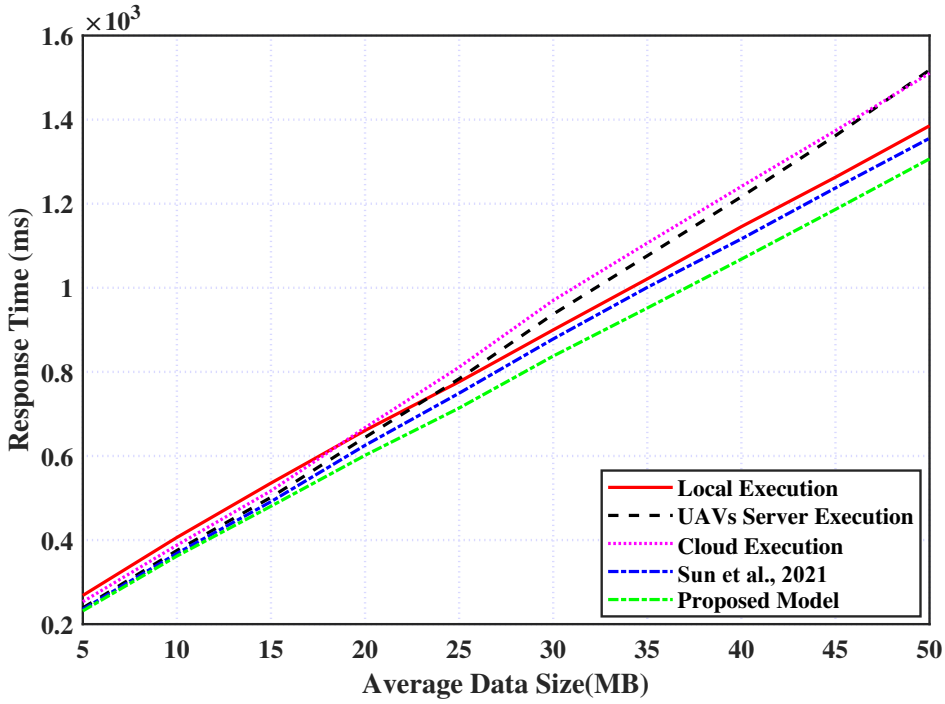


Figure 5. Response time in relation to different values of input data size.

Finally, the consumption of energy and response time for processing the tasks in relation to different numbers of UAVs is presented in Figures 6 and 7, respectively. It is shown from these two figures that the local and cloud execution scenarios are not affected by the number of UAVs, whereas the energy consumption and response time of the other three scenarios (i.e., UAVs server, model in [39], and proposed model) gradually decrease as the number of UAVs increases. In addition, our model can achieve a lower energy consumption and response relative to the other two scenarios. This is because the costs in terms of energy consumption and response time are decreasing as the mobile users are allocated more resources (i.e., by increasing the number of UAVs), whereas the local execution and cloud execution scenarios do not depend on UAV resources.

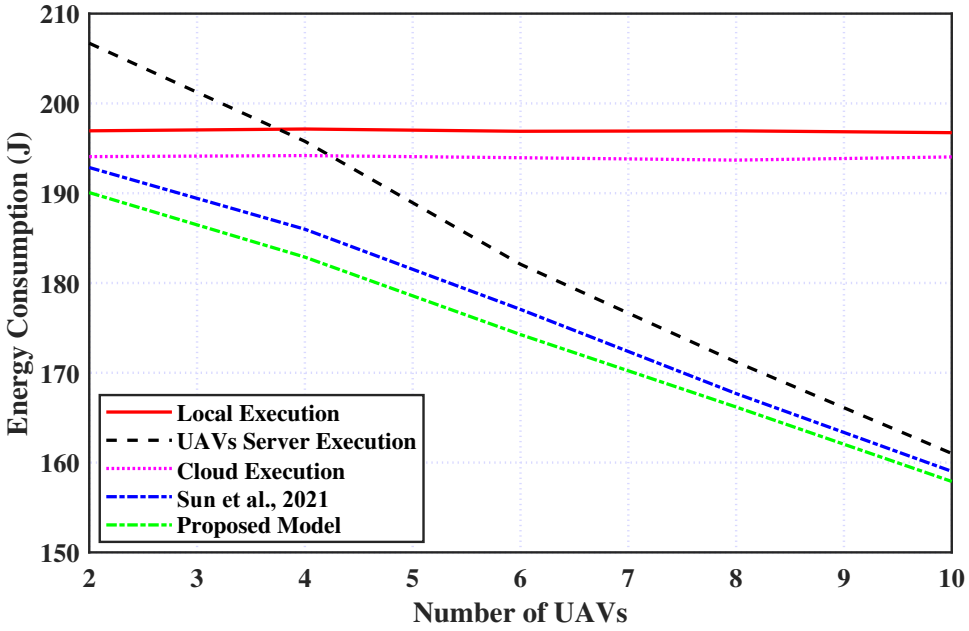


Figure 6. Energy consumption in relation to different number of UAVs.

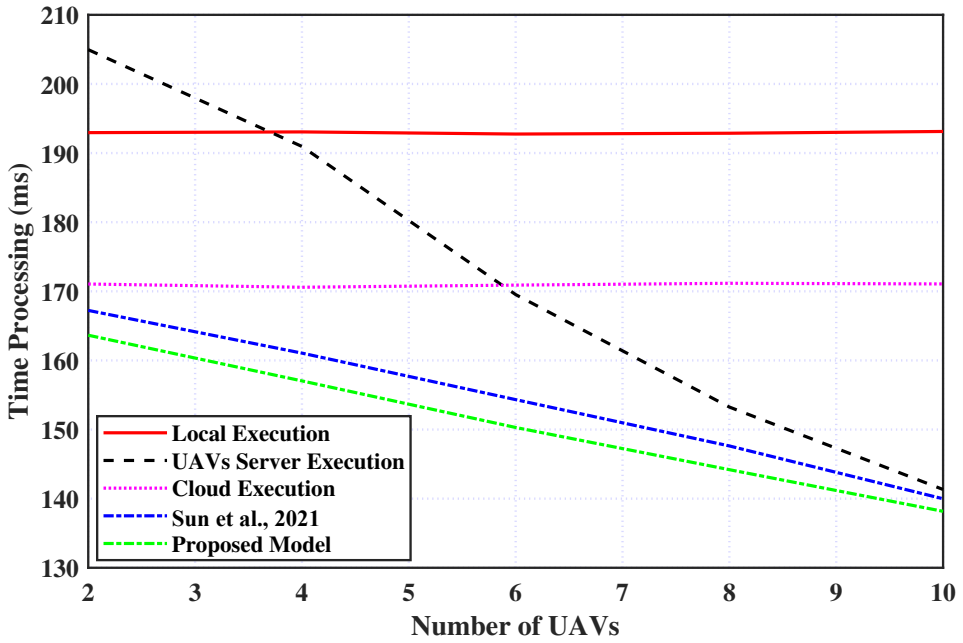


Figure 7. Response time in relation to different number of UAVs.

5. Conclusions

Our study proposed an energy-efficient system model for a multi-user with a multi-UAV-enabled ME-cloud computing system. This system is scalable and can support increases in network traffic without performance decreases. In addition, a network of UAVs is deployed to cover the dead and high-density network areas. Furthermore, UAVs can provide other applications in a smart city network in which the MEC technology is deployed in multiple levels to provide the capabilities of computation at the edge of the network. The core network is based on SDN technology to manage network traffic and to provide an innovative interface for network operators. Finally, simulation experiments demonstrate that our proposed model can significantly decrease the energy consumption of the entire system.

One promising future direction is to utilize blockchain technology for securing and protecting the UAV network from different types of attacks. Moreover, a more general case where UAVs and device users can dynamically move within the period of offloading will be considered. In such cases, deep learning techniques can be utilized and exploited to address the modeling and formulation of the problem as well as the solution in an efficient manner.

Author Contributions: Conceptualization, I.A.E. and A.M.; methodology, I.A.E.; software, A.M. and S.A.; validation, I.A.E and S.A.; formal analysis, A.M. and I.A.E; investigation, A.M.; resources, S.A.; writing—original draft preparation, I.A.E.; writing—review and editing, A.M. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Saudi Electronic University for funding this research work through the project number (8117).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at Saudi Electronic University for funding this research work through the project number (8117).

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

References

1. Wang, J.; Jiang, C.; Han, Z.; Ren, Y.; Maunder, R.G.; Hanzo, L. Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones. *IEEE Veh. Technol. Mag.* **2017**, *12*, 73–82. [CrossRef]
2. CAAC. Low-Altitude Connected Drone Flight Safety Test Report. Available online: http://www.caac.gov.cn/en/ (accessed on 27 April 2022).
3. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 2624–2661. [CrossRef]
4. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
5. Yaacoub, E.; Alouini, M.S. A key 6G challenge and opportunity—connecting the base of the pyramid: A survey on rural connectivity. *Proc. IEEE* **2020**, *108*, 533–582. [CrossRef]
6. Dang, S.; Amin, O.; Shihada, B.; Alouini, M.S. What should 6G be? *Nat. Electron.* **2020**, *3*, 20–29. [CrossRef]
7. Tikhomirov, A.; Omelyanchuk, E.; Semenova, A. Recommended 5G frequency bands evaluation. In Proceedings of the 2018 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, 14–15 March 2018; pp. 1–5.
8. Bargis, M.; Romano, G. From Standards to Service-the European Way to 5G. *IEEE 5G Tech Focus* **2017**, *1*, 15–21.
9. Obanawa, H.; Shibata, H. Applications of UAV Remote Sensing to Topographic and Vegetation Surveys. In *Unmanned Aerial Vehicle: Applications in Agriculture and Environment*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–142.
10. Bithas, P.S.; Michailidis, E.T.; Nomikos, N.; Vouyioukas, D.; Kanatas, A.G. A Survey on Machine-Learning Techniques for UAV-Based Communications. *Sensors* **2019**, *19*, 5170. [CrossRef]
11. Khayyat, M.; Alshahrani, A.; Alharbi, S.; Elgendy, I.; Paramonov, A.; Koucheryavy, A. Multilevel Service-Provisioning-Based Autonomous Vehicle Applications. *Sustainability* **2020**, *12*, 2497. [CrossRef]
12. Mach, P.; Becvar, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 1628–1656. [CrossRef]
13. Cheng, N.; Xu, W.; Shi, W.; Zhou, Y.; Lu, N.; Zhou, H.; Shen, X. Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities. *IEEE Commun. Mag.* **2018**, *56*, 26–32. [CrossRef]

14. Zhou, F.; Wu, Y.; Hu, R.Q.; Qian, Y. Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1927–1941. [CrossRef]
15. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 2322–2358. [CrossRef]
16. Khayyat, M.; Elgendy, I.A.; Muthanna, A.; Alshahrani, A.; Alharbi, S.; Koucheryavy, A. Advanced Deep Learning-based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. *IEEE Access* **2020**, *8*, 137052–137062. [CrossRef]
17. Zhang, X.; Zhong, Y.; Liu, P.; Zhou, F.; Wang, Y. Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization. *IEEE Access* **2019**, *7*, 113345–113354. [CrossRef]
18. Zhang, J.; Zhou, L.; Tang, Q.; Ngai, E.C.H.; Hu, X.; Zhao, H.; Wei, J. Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing. *IEEE Internet Things J.* **2018**, *6*, 3688–3699. [CrossRef]
19. Beiqing, C.; Haihang, Z.; Jianguo, Y.; Haibing, G. RESERVE: An Energy-Efficient Edge Cloud Architecture for Intelligent Multi-UAV. *IEEE Trans. Serv. Comput.* **2019**, *15*, 819–832.
20. You, C.; Huang, K.; Chae, H. Energy efficient mobile cloud computing powered by wireless energy transfer. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 1757–1771. [CrossRef]
21. Huang, L.; Bi, S.; Zhang, Y.J. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. *IEEE Trans. Mob. Comput.* **2019**, *19*, 2581–2593. [CrossRef]
22. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [CrossRef]
23. Dinh, T.Q.; La, Q.D.; Quek, T.Q.; Shin, H. Learning for computation offloading in mobile edge computing. *IEEE Trans. Commun.* **2018**, *66*, 6353–6367. [CrossRef]
24. Liu, F.; Huang, Z.; Wang, L. Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors. *Sensors* **2019**, *19*, 1105. [CrossRef] [PubMed]
25. Qu, G.; Wu, H.; Li, R.; Jiao, P. DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 3448–3459. [CrossRef]
26. Lu, H.; He, X.; Du, M.; Ruan, X.; Sun, Y.; Wang, K. Edge QoE: Computation offloading with deep reinforcement learning for Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 9255–9265. [CrossRef]
27. Ji, L.; Guo, S. Energy-efficient cooperative resource allocation in wireless powered mobile edge computing. *IEEE Internet Things J.* **2018**, *6*, 4744–4754. [CrossRef]
28. Hu, L.; Tian, Y.; Yang, J.; Taleb, T.; Xiang, L.; Hao, Y. Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming. *IEEE Netw.* **2019**, *33*, 42–48. [CrossRef]
29. Zhan, Y.; Guo, S.; Li, P.; Zhang, J. A deep reinforcement learning based offloading game in edge computing. *IEEE Trans. Comput.* **2020**, *69*, 883–893. [CrossRef]
30. Wang, Y.; Chen, W.; Luan, T.H.; Su, Z.; Xu, Q.; Li, R.; Chen, N. Task Offloading for Post-Disaster Rescue in Unmanned Aerial Vehicles Networks. *IEEE/ACM Trans. Netw.* 2022, *early access*. [CrossRef]
31. Zhou, Z.; Feng, J.; Tan, L.; He, Y.; Gong, J. An air-ground integration approach for mobile edge computing in IoT. *IEEE Commun. Mag.* **2018**, *56*, 40–47. [CrossRef]
32. Lin, L.; Li, P.; Liao, X.; Jin, H.; Zhang, Y. Echo: An edge-centric code offloading system with quality of service guarantee. *IEEE Access* **2018**, *7*, 5905–5917. [CrossRef]
33. Goudarzi, M.; Wu, H.; Palaniswami, M.; Buyya, R. An application placement technique for concurrent IoT applications in edge and fog computing environments. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1298–1311. [CrossRef]
34. Xia, J.; Wang, P.; Li, B.; Fei, Z. Intelligent task offloading and collaborative computation in multi-UAV-enabled mobile edge computing. *China Commun.* **2022**, *19*, 244–256. [CrossRef]
35. Bai, T.; Wang, J.; Ren, Y.; Hanzo, L. Energy-Efficient Computation Offloading for Secure UAV-Edge-Computing Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6074–6087. [CrossRef]
36. Yang, Z.; Pan, C.; Wang, K.; Shikh-Bahaei, M. Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks. *arXiv* **2019**, arXiv:1902.03158.
37. Elgendy, I.; Zhang, W.; Liu, C.; Hsu, C.H. An efficient and secured framework for mobile cloud computing. *IEEE Trans. Cloud Comput.* **2018**, *9*, 79–87. [CrossRef]
38. Deb, S.; Monogioudis, P. Learning-based uplink interference management in 4G LTE cellular systems. *IEEE/ACM Trans. Netw.* **2015**, *23*, 398–411. [CrossRef]
39. Sun, S.; Zhang, G.; Mei, H.; Wang, K.; Yang, K. Optimizing multi-UAV deployment in 3-D space to minimize task completion time in UAV-enabled mobile edge computing systems. *IEEE Commun. Lett.* **2021**, *25*, 579–583. [CrossRef]
40. Elgendy, I.A.; Zhang, W.; Tian, Y.C.; Li, K. Resource allocation and computation offloading with data security for mobile edge computing. *Future Gener. Comput. Syst.* **2019**, *100*, 531–541. [CrossRef]
41. Lyu, X.; Tian, H.; Sengul, C.; Zhang, P. Multiuser joint task offloading and resource optimization in proximate clouds. *IEEE Trans. Veh. Technol.* **2016**, *66*, 3435–3447. [CrossRef]
42. Elgendy, I.A.; Zhang, W.Z.; Zeng, Y.; He, H.; Tian, Y.C.; Yang, Y. Efficient and Secure Multi-User Multi-Task Computation Offloading for Mobile-Edge Computing in Mobile IoT Networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2410–2422. [CrossRef]

43. Lyu, X.; Tian, H. Adaptive receding horizon offloading strategy under dynamic environment. *IEEE Commun. Lett.* **2016**, *20*, 878–881. [CrossRef]
44. Chen, X. Decentralized Computation Offloading Game for Mobile Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 974–983. [CrossRef]
45. Fooladivanda, D.; Rosenberg, C. Joint resource allocation and user association for heterogeneous wireless cellular networks. *IEEE Trans. Wirel. Commun.* **2012**, *12*, 248–257. [CrossRef]
46. Guignard, M. Lagrangean relaxation. *Top* **2003**, *11*, 151–200. [CrossRef]
47. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* **2015**, *24*, 2795–2808. [CrossRef]
48. Beal, L.D.; Hill, D.C.; Martin, R.A.; Hedengren, J.D. Gekko optimization suite. *Processes* **2018**, *6*, 106. [CrossRef]