2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 978-1-6654-0527-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/CSCWD54268.2022.9776115

# Vehicular Computation Offloading in UAV-enabled MEC Systems

Junfeng Wang[1], Dayu Feng[1], Jie Zhu[*1,2], Haiping Huang[*1,3]
[1]*Nanjing University of Posts & Telecommunications, Nanjing 210003, China*
[2]*State Key Laboratory Chinese of Computer Architecture, Institute of Computing Technology,*
*Chinese Academy of Sciences, Beijing 100864, China*
[3]*Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
junfwang@163.com, 2218785142@qq.com, zhujie@njupt.edu.cn, hhp@njupt.edu.cn

***Abstract*—The UAV-enable Mobile Edge Computing (MEC) systems and Vehicular Ad-hoc Network (VANET)-supported applications are very popular topics these days. The paper considers the vehicular task offloading problems for the Software-Defined Vehicular Network (SDVN)-supported services in the UAV-enabled MEC system. In the considered problem, one UAV and one edge server (ES) are provisioned for the workload from the moving vehicles in a certain region. For each vehicle in the region, it would periodically submit requests to the UAV-enable MEC system until it leaves the region. Each request will be taken as a computation task and could be offloaded locally on the vehicle, the UAV, or the ES. Multiple communication and energy consumption models are employed to formulate the problem model. The objectives are to minimize the total time delays and the energy consumption. A greedy heuristic based dynamic scheduling framework is proposed for the problem under study. Simulated experiments are delicately designed with dynamic traffics, various road and building distributions. Experimental results show that the proposal is more effective than the compared algorithm.**

***Keywords*—UAV; UAV-enabled Mobile Edge Computing systems; dynamic scheduling framework; vehicular computation offloading.**

## I. Introduction

The low-latency communication through vehicle-to-everything has been enabled by the advanced communication technologies such 5G, DSRC [1] and LTE-V [2]. Vehicular Ad-hoc Network (VANET) [3] is an interactive wireless network built on a large number of heterogeneous smart vehicles equipped with IoT devices and intelligent control systems. It provides a new networking paradigm for urban computing, data sharing, passenger safety, traffic efficiency and infotainment. Due to the new opportunities for a wide variety of innovative applications and services, VANET has attracted increasing research interest and investigation. VANET-supported services focus on safe driving and provide drivers and passengers with comfort, safety and entertainment. However, VANET has still faced great challenges in terms of security, accuracy and stability, due to its highly dynamic topology, variable network density and complex city conditions. Some of these challenges could be resolved by advanced networking solutions such as Software-Defined Vehicular Network (SDVN). By separating the logical control plane and data plane in SDVN [4], mobile vehicles are abstracted as SDN switches or computing nodes that can be centralized controlled and assigned with computing tasks. With the aid of SDVN, VANET-supported services could be decoupled from the control logic from the hardware and focus on security management, QoS, resource allocating and task scheduling.

VANET-supported services are deployed in vehicles and may require the cooperation of nearby vehicles or remote computing systems. These services are data and computing-intensive, so embedded computing systems in vehicles can not meet the complex computing, low latency and low energy consumption demands. Therefore, the edge computing has been extended to VANET. The edge-computing devices can be deployed in the local regions. They provide high-performance and reliable computing services to the passing vehicles on nearby roads. However, the edge-computing devices are deployed with static locations and moved rarely, which means they are low in agility and flexibility to the varied workload. The workload from roads are regular with predictable peak and trough periods. It is costly and wasteful to deploy the edge devices according to the workload of peak times. It is also inefficient to offload workload from busy devices to idle devices in the near region under complicated radio environments. Besides, in some special region there may be no access point and base station to allow network access, or these infrastructures are temporarily damaged by natural disasters. Dense buildings or special geographical environments may also hinder data transmission. Therefore, we turn to the mobile edge computing devices such as unmanned-aerial-vehicles (UAVs) to generate the flexible layout of computing capabilities.

In recent years, UAV has been widely investigated in military and civil industries due to its high mobility and low environmental disruption [5]. UAVs are usually taken as auxiliary computing devices to the overloaded local systems. They can also be employed as relay nodes to forwarding requests to nearby edge devices. In the UAV-enabled Mobile Edge Computing (MEC) system, UAVs can hover over the certain area to process real-time tasks or offload them to idle edge devices.

In this paper, we consider the task offloading problem in the UAV-enabled MEC system. In the considered problem,



Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.

one UAV and one edge server (ES) are provisioned for the workload from the moving vehicles in a certain region (as shown in Fig. 1).

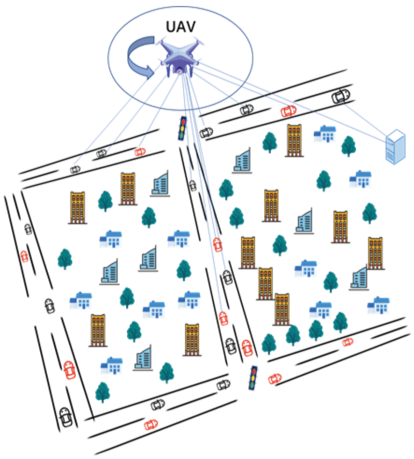



Fig. 1. UAV-enabled MEC system

From the time that a vehicle enters the region, it would generate computing tasks periodically until it leaves the region. Each vehicle has heterogeneous computing capability which is usually not sufficient for its own tasks. There is one UAV hovering over the area with the certain path and speed. It either provides the computing capability directly to vehicles or offloads the tasks to the edge server. A vehicle could also offload its tasks directly to the ES. However, the edge server may not be connectable due to dense buildings in some location of the region. The UAV may be not connectable when the distance between a vehicle and the UAV exceeds communication range. The objectives are to minimize the total time delays and the energy consumption. There are three main challenges for the considered problem. (i) Both the UAV and vehicles are moving. The data transmission rate is varied depending on the distance between the UAV, vehicles and edge server. It is complex to determine the task offloading plan with all these time-varied factors. (ii) Tasks are periodically generated by each vehicle. It would be difficult to schedule the periodical tasks which are generated at different locations and times. (iii) It is challenging to evaluate how the buildings affect the communication quality. Introducing the communication blind areas greatly increases the difficulty of the problem. (iv) Offloading tasks to vehicles, UAV or ES leads to different data transmission times, task computing times and energy consumption. The considered objectives are conflicting to some extent. How to balance the conflicting objectives is a big challenge.

The main contributions of this paper are summarized as follows:

- In terms of the different data transmission models among UAV, heterogeneous vehicles and ES, the considered problem is mathematically defined.
- An dynamic algorithm framework is proposed to offload periodic tasks with the objectives of minimizing total time delay and energy consumption. A greedy heuristic is introduced.

The rest of the paper is organized as follows: Related works are reviewed in Section II. Section III presents the problem model. A heuristic algorithm is proposed in Section IV. Section V evaluates the performance of the proposal followed by conclusions and future research in Section VI.

## II. Related Work

The vehicle task offloading problems (VTOP) have been widely studied in past decades. Huang et al. [6] propose a Lyapunov optimization based dynamic offloading algorithm to improve the mobile cloud computing performance while meeting the application execution time. Pan et al. [7] present a Deep Reinforcement Learning-based URLLC (ultra-reliable and low-latency communications)-Aware task offloading algorithm named DREAM to maximize the throughput of the user vehicles while satisfying the URLLC constraints in a best-effort way. Du et al. [8] study a cognitive vehicular network that uses the TVWS (TV white space) band, and formulate a dual-side optimization problem, to minimize the cost of VTs (vehicular terminals) and that of the ES at the same time. Jang et al. [9] employ vehicular edge computing (VEC) which offloads the computation to the VEC node and jointly optimize the offloading proportion and uplink/computation/downlink bit allocation of multiple vehicles for the purpose of minimizing the total energy consumption of the vehicles under the delay constraint.

UAVs have the advantages in flexible mobility, low deployment costs, low environmental requirements, and a variety of sizes [10]. In recent years, UAVs have been introduced to assist VTOP. Some studies investigate on how the obstruction of obstacles affects the communication quality and introduce UAVs to the task offloading problems in order to reduce the packet loss. Zhao et al. [11] propose an SDN (Software-Defined Networking)-enabled UAV-assisted vehicular computation offloading optimization framework to minimize the system cost of vehicle computing tasks. Liu et al. [12] research a game based secure data transmission scheme in UAV assisted vehicular internet of things and exploited the offensive and defensive game to model the interactions between the normal UAVs and jammers. Samir et al. [13] leverage deep reinforcement learning to propose an approach for learning the optimal trajectories of the deployed UAVs to efficiently maximize the vehicular coverage, and adopt a Actor-Critic algorithm to learn the vehicular environment and its dynamics to handle the complex continuous action space. He et al. [14] investigate the relay selection problem for the air-to-ground VANETs, and use UAV to enhance VANETs communications and network performance.

Although the above studies offer effective solutions to the VTOP by introducing the UAVs, there are still many real-time factors they do not consider. In most of these studies, each vehicle only has one task to process and the location of the task has no effect to the transmission rate. They usually employ one data transmission model and the heterogeneity



Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.

of computing resources is not considered. The most related study is the one in [11]. The major difference between the considered problem in this paper and the one in [11] is that the ES in [11] has unlimited computing capability. While in the paper, the tasks are processed on the edge server in parallel with a upper-bounded limit. Based on these related studies, the considered problem is more realistic and worth studying.

## III. Problem Description

In this section, we formulate the task offloading problem for the Software-Defined Vehicular Network (SDVN)-supported services in the UAV-enabled Mobile Edge Computing (MEC) system.

We consider a region covering a set $\mathbf{L}$ of lanes and a set $\mathbf{B}$ of buildings. A lane $L_j \in \mathbf{L}$ is described by a path with a start point $l_j^s$ and an end point $l_j^e$ in this region. The driving direction of vehicles is determined by the lane, i.e., from the start point to the end point of a lane. A building $B_j \in \mathbf{B}$ is described by its geographic information mainly involving its location and 3D profile.

In the region, there are a set $\mathbf{V}$ of $n$ vehicles driving on the roads. A vehicle $v_i \in \mathbf{V}$ can be represented by a vector $(l_i^0, \mathcal{L}_i, \nu_i, p_i, s_i, F_i^v, P_i^v, e_i^v, \mathbf{T}_i)$. $l_i^0$ and $s_i$ are the location and time that $v_i$ submits its first request. $\mathcal{L}_i \in \mathbf{L}$ is the lane that $v_i$ is on. $l_i^0$ must be on the path of $\mathcal{L}_i$. $\nu_i$ is the average driving speed of $v_i$. The direction of $v_i$ is determined by the path of $\mathcal{L}_i$. $p_i$ is the period length for $v_i$ to generate requests. $v_i$ periodically submits requests until it leaves the region. A request can be taken as a task to be processed. $\mathbf{T}_i$ is the set of tasks that $v_i$ submits during the time it drives in the region. $F_i^v$ is the computation capability of $v_i$ and $P_i^v$ is the transmission power of $v_i$. $e_i^v$ is the energy consumed in each CPU cycle of $v_i$.

There is one UAV and one edge server (ES) provisioned in the region. The ES is located at the location $l_e$ and its computation capability is $F^{es}$. ES can process requests in parallel. The maximum number of tasks that can process in parallel on ES is $I_{max}$ The UAV flies on a fixed path $L^{uav}$ at the height $H$. $L^{UAV}$ is a circle with the center $\theta$ and the radius $\gamma$. Suppose the UAV is located at $L^0$ at the very beginning and flies on the path at the average speed of $\nu^{uav}$. The computation capability of the UAV is $F^{uav}$.

A task $T_{i,k}$ is the $k^{th}$ request submitted by $v_i$, $k = 1, \ldots, |\mathbf{T}_i|$. $s_{i,k}$ is the submitted time of $T_{i,k}$, i.e., $s_{i,k} = s_i + (k-1) \times p_i$. $C_{i,k}$, $O_{i,k}$ and $D_{i,k}$ are the total CPU cycles required for processing $T_{i,k}$, the size of data to be processed and the size of result to return, respectively. $l_{i,k}$ is the location of $T_{i,k}$. The distance between $l_{i,k}$ and $l_i^0$ is $(k-1) \times p_i \times \nu_i$.

The computation time $t_{i,k}$, the transmission time $d_{i,k}$ and the energy consumption $E_{i,k}$ of $T_{i,k}$ are determined by the offloading destination of the task. A task can be either processed on the vehicle, the UAV, or the ES. When a task is processed on the ES, it may be offloaded directly from the vehicle, or offloaded through UAV which works as a relay.

- If $T_{i,k}$ is locally processed on $v_i$, $t_{i,k} = \frac{C_{i,k}}{F_i^v}$. There is no transmission time, i.e., $d_{i,k} = 0$. The energy consumption $E_{i,k} = C_{i,k} \times e_i^v$.
- If $T_{i,k}$ is processed on the UAV, $t_{i,k} = \frac{C_{i,k}}{F^{uav}}$. The transmission time involves the transmission time $d_{i,k}^{data}$ of $O_{i,k}$ and the transmission time $d_{i,k}^{result}$ of $D_{i,k}$. $d_{i,k}^{data}$ and $d_{i,k}^{result}$ can be computed by Equ.(1)-(2). The energy consumption $E_{i,k} = E_{i,k}^{v-uav} + E_{i,k}^{uav} + E_{i,k}^{uav-v}$, where $E_{i,k}^{v-uav}$ is the energy consumption for transmitting $O_{i,k}$ from the vehicle to the UAV, $E_{i,k}^{uav}$ is the energy consumption for processing the task on the UAV and $E_{i,k}^{uav-v}$ is the energy consumption for transmitting $D_{i,k}$ from the UAV to vehicle. These energy consumptions can be computed by Equ.(4)-(6).
- If $T_{i,k}$ is offloaded to the ES directly, $t_{i,k} = \frac{C_{i,k}}{F^{ES}}$. The transmission time involves the transmission time $d_{i,k}^{data}$ of $O_{i,k}$ and the transmission time $d_{i,k}^{result}$ of $D_{i,k}$. Where $d_{i,k}^{data}$ and $d_{i,k}^{result}$ can be computed by Equ.(1)-(2). The energy consumption $E_{i,k} = O_{i,k} \times e_{SEND}^i$.
- If $T_{i,k}$ is offloaded to the ES through the UAV, $t_{i,k} = \frac{C_{i,k}}{F^{ES}}$. The transmission time involves the transmission time $d_{i,k}^{data}$ of $O_{i,k}$ from $v_i$ to the ES through the UAV, the transmission time $d_{i,k}^{result}$ of $D_{i,k}$ from the ES to to $v_i$ through the UAV. Where $d_{i,k}^{data}$ and $d_{i,k}^{result}$ can be computed by Equ.(1)-(2). The energy consumption $E_{i,k} = E_{i,k}^{v-uav} + E_{i,k}^{uav-es} + E_{i,k}^{uav-v}$, where $E_{i,k}^{v-uav}$ is the energy consumption for transmitting $O_{i,k}$ from $v_i$ to the UAV (Equ.(4)), $E_{i,k}^{uav-es}$ is the energy consumption for transmitting $O_{i,k}$ from the UAV to the ES (7) and $E_{i,k}^{uav-v}$ is the energy consumption for transmitting $D_{i,k}$ from the UAV to $v_i$ (6).

Some important parameters defined in the communication models are given in Table I. All the communication models and parameters employed in Equs. (3),(8) and (9) are defined in [11] and [15].

$$d_{i,k}^{data} = \frac{O_{i,k}}{R(l,l')} \tag{1}$$

$$d_{i,k}^{result} = \frac{D_{i,k}}{R(l,l')} \tag{2}$$

$$R(l,l') = \begin{cases} R_{i,uav}, & l = v_i, l' = UAV \\ R_{WiFi}, & l = UAV, l' = v_i \\ R_{i,es}, & l = v_i, l' = ES \\ R_{LTE}, & l = ES, l' = v_i \\ R_{WiFi}, & l = UAV, l' = ES \\ R_{LTE}, & l = ES, l' = UAV \end{cases} \tag{3}$$

$$E_{i,k}^{v-uav} = O_{i,k} \times e_i^{SEND} \tag{4}$$

$$E_{i,k}^{uav} = C_{i,k} \times e^{uav} \tag{5}$$

$$E_{i,k}^{uav-v} = D_{i,k} \times e_{SEND}^{uav} \tag{6}$$

$$E_{i,k}^{uav-es} = O_{i,k} \times e_{SEND}^{uav} \tag{7}$$



Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DEFNITION OF NOTATION

| Symbol | Discription |
|---|---|
| $P_i^v$ | The transmission power of $v_i$ |
| $e^{uav}/e_i^v$ | Energy consumed in each CPU cycle of the UAV/vehicles executing locally |
| $R(l,l')$ | The data transmission rate from $l$ to $l'$ |
| $R_{i,uav}/R_{i,es}$ | The transmission rate of data sent by vehicle n to UAV/MEC server |
| $R_{LTE}/R_{WiFi}$ | The data transmission rate of LTE/Wi-Fi interface |
| $e_{SEND}^{uav}/e_i^{SEND}$ | Energy consumed by a UAV/vehicle to send one data unit to the other devices |
| $W$ | Channel bandwidth |
| $\alpha_V$ | Path loss exponent for V2I channels |
| $N_0$ | Noise power |
| $\varpi$ | Weight of execution time in the payoff function |
| $d_{i,uav}/d_{i,es}$ | Distance between vehicle i and the UAV/MEC |
| $N$ | Number of vehicles |

$$R_{i,uav} = W\log_2(1+\frac{P_i^v d_{i,uav}^{-\alpha_V}}{N_0+\sum_{s\in N,s\neq i} P_i^v d_{s,uav}^{-\alpha_V}}) \tag{8}$$

$$R_{i,es} = W\log_2(1+\frac{P_i^v d_{i,es}^{-\alpha_V}}{N_0+\sum_{s\in N,s\neq i} P_i^v d_{s,es}^{-\alpha_V}}) \tag{9}$$

$$TEC = \sum_{v_i\in V}\sum_{k=1}^{|\mathbf{T}_i|} E_{i,k} \tag{10}$$

$$TET = \sum_{v_i\in V}\sum_{k=1}^{|\mathbf{T}_i|} (t_{i,k}^{end} - t_{i,k}^{start}) \tag{11}$$

$$LWS = \varpi \times TEC + (1-\varpi)\times TET \tag{12}$$

The assignment $a_{i,k}$ of a task can be represented as a vector $(t_{i,k}^{start}, t_{i,k}^{end}, w_{i,k})$, where $t_{i,k}^{start}$ is the time that the task starts to transmit data, $t_{i,k}^{end}$ is the finish time of the task, $w_{i,k}$ is the offloading path of the task. $w_{i,k} = uav$ means the task is offloaded to the UAV. $w_{i,k} = es$ means the task is offloaded to the ES. $w_{i,k} = v_i$ means the task is processed locally. $w_{i,k} = uav - es$ means the task is first offloaded to the UAV, then transferred from the UAV to ES to process.

A schedule $S$ is the set of feasible assignments for all tasks. A feasible schedule must satisfy the following constraints.

- Each task cannot start before its generation time, i.e., $t_{i,k}^{start} \geq s_{i,k}$.
- Each task cannot finish until the vehicle receives the result of the task, i.e., $t_{i,k}^{end} \geq t_{i,k}^{start} + d_{i,k} + t_{i,k}$.
- Since UAV can process only one task at a time, the processing times of the tasks assigned to the UAV cannot overlap.
- The number of tasks processed in parallel on ES cannot exceed $I_{max}$.

The objective is to minimize the total energy consumption $TEC$ and the total execution time $TET$ of all tasks. $TEC$ and $TET$ can be computed by Equs. (10)-(11). Since the considered problem is bi-objective, we apply a LWS (Linear Weighted Sum) method to evaluate solutions. Equ. 12 is the LWS of the total energy consumption $TEC$ and the total execution time $TET$. $\varpi$ ($\varpi \in (0,1)$) is the weight coefficient to control the tradeoff between energy and time.

## IV. Proposed Algorithm

In this paper, we employ the Greedy Heuristic (GH) based dynamic scheduling framework (GHDSF) for the bi-objective optimization problem under study. The major framework of the proposed algorithm is formally described in Algorithm 1.

A feasible schedule § is initially empty at time $q = 0$ and updated iteratively until the end time $q_{max}$ (Lines 3-15):

1) The execution progress of tasks is monitored and the resource availability of vehicles, UAV and ES is updated (Line 3). After some tasks are completed, the corresponding resource will be set to available.
2) Tasks arriving at the time interval $[q, q+1]$ are collected to be scheduled (Line 4).
3) A GH-based heuristic is performed on the collect tasks to generate the priorities of tasks in the waiting queue (Line 5).
4) Arrange tasks according to their priorities in the queue(Lines 6-15).

**Algorithm 1:** GH-based Dynamic Scheduling Framework

```
1  S ← ∅;
2  for q = 0 to q_max do
3      Monitoring the execution of tasks and updating the
       availability of vehicles, UAV and ES;
4      Collecting tasks to be scheduled in the current time
       window [q, q+1] and packaging them into the
       priority queue Q;
5      Perform GH on Q to generate the improved priority
       queue;
6      for ∀t ∈ Q do
7          Set t* to be the task in Q with the highest
           priority;
8          Offload t* to appropriate offloading path w*;
9          if w* is not available then
10             Break;
11         else
12             Compute the start and end time of t*;
13             Generate the assignment a* of t*;
14             S ← S + {a*};
15             Remove t* from Q;
16 return S.
```

The greedy heuristic is the most important component of GHDSF. Details of the heuristic are introduced in Algorithm 2. In GH, a task is arranged to the destination with the minimum contribution to LWS. The following steps are employed for the task arrangement process.

- Check the connectivity of the task with the UAV and ES and generate the candidate destinations. There are four available options for a task: the vehicle, the UAV, the ES or transferring from the UAV to the ES.
- Estimate the start and end time of the task for each available destination and generate candidate assignments.



Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.

- Compute the LWS of all candidate assignments and the one with the minimum LWS is taken as the arrangement of the task.

The termination condition of $GH$ is the maximum running time of the heuristic, i.e., the heuristic stops at the end of the current time window.

**Algorithm 2:** Greedy Heuristic

```
Q* ← Q;
LWS* ← +∞;
while (termination condition not met) do
    Swapping multiple random unassigned tasks of Q to
    generate Q^new;
    S ← ∅;
    for Each task t ∈ Q^new do
        Arrange t to appropriate destination;
        Generate assignment a of t;
        S ← S + {a};
    if LWS(S) < LWS* then
        LWS* ← LWS(S);
        Q* ← Q^new;
return Q*.
```

## V. Experimental Evaluation

In order to evaluate the performance of the proposed algorithm, the proposed algorithm is compared to UAV-assisted Vehicular computation Cost Optimization (UVCO) algorithm based on the game theory [11] for similar problems. The main idea of the compared game-theory-based algorithm is that each task is assigned to the resource by a game theory based strategy. The algorithms are encoded in Java, compiled by Eclipse JDK1.7 and run on an Intel Core $i7-8565U\ CPU$ @1.80 GHz with 8 GBytes of RAM.

In order to evaluate the performance of the compared algorithms, the average relative percentage deviation (ARPD) is employed as the metric for the solutions. ARPD is computed by Equ. (13).

$$ARPD = \frac{LWS^{ALG} - LWS^{*}}{LWS^{*}} \times 100\% \tag{13}$$

Where $LWS^{*}$ is the best LWS obtained by the compared algorithms and $ALG \in \{UVCO, GHDSF\}$.

Since there are no existing benchmark instances for the considered problem, the testing instances are generated based on the existing and related studies [11]. Generating the testing instances involves the map size ($\{800 \times 800, 1000 \times 1000, 1200 \times 1200\} m^2$), the average driving speed of vehicles $\nu_i \in \{54, 72\} km/h$, the maximum parallel number $I_{max} \in \{10, 15\}$ of ES and four types of road distribution as shown in Fig. 2. There are four lanes on each road, where two lanes have one direction and the other two take the other direction. The other parameter settings are given in Table II.

TABLE II
Parameter Settings

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $W$ | 20MHz | $\alpha_V$ | 4 |
| $N_0$ | $-100$dB | $e^{uav}$ | 1u |
| $e_i^v$ | 1u | $e_{SEND}^{uav}$ | 50u |
| $e_i^{SEND}$ | 25u | $P_i^v$ | $\{100, 120, 130, 150\}$kw |
| $\varpi$ | 0.8 | $R_{LTE}$ | 1000mWatts |
| $R_{WiFi}$ | 100mWatts | $N$ | 30 |
| $F_i^v$ | $\{0.5, 0.7, 0.8, 1.0\}$GHz | $F_{ES}$ | 50GHz |
| $F^{uav}$ | 10GHz | $H$ | 300m |
| $\gamma$ | 100m | $\nu^{uav}$ | 20m/s |

TABLE III
Experiment Results

| Param. | Value | ARPD(%) | |
|---|---|---|---|
| | | GHDSF | UVCO |
| Map | 800 × 800 | 0.00 | 8.23 |
| | 1000 × 1000 | 0.00 | 10.19 |
| | 1200 × 1200 | 0.00 | 9.89 |
| $\nu_i$ | 54 | 0.00 | 10.03 |
| | 72 | 0.00 | 8.84 |
| Road Distribution | Type one | 0.00 | 9.71 |
| | Type two | 0.00 | 10.12 |
| | Type three | 0.00 | 9.31 |
| | Type four | 0.00 | 8.60 |
| $I_{max}$ | 10 | 0.00 | 9.77 |
| | 15 | 0.00 | 9.10 |
| Avg. | | 0.00 | 9.44 |

There are $3 \times 2 \times 4 \times 2 = 48$ combinations of instance parameters. For each combination of instance parameters, we generate 5 instances with random locations of buildings and initial locations of vehicles. Therefore, there are 240 instances in total. The ARPD metric of all instance combinations of the compared algorithms is shown in Table III. Table III illustrates that the proposed GHDSF has the best ARPDs for all instances, whereas the average ARPD for UVCO is 9.44%. GHDSF outperforms UVCO on all instance combinations. The major reason may be that UVCO does not consider the workload already assigned to resources when arranging tasks.

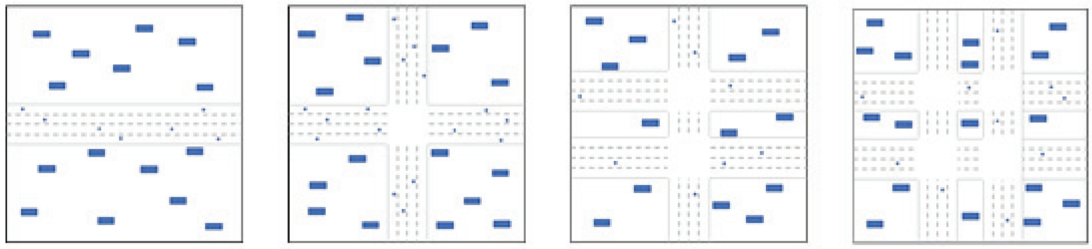
Based on the experimental results, we can conclude that the proposed algorithm is more effective and suitable for the considered problem.

## VI. Conclusion

The paper considers a task offloading problem in the UAV-enable Mobile Edge Computing (MEC) system for the Vehicular Ad-hoc Network (VANET)-supported applications. The greedy heuristic (GH) based dynamic scheduling framework (GHDSF) is proposed for the bi-objective optimization problem under study. In GHDSF, the GH is employed to improve the schedule solution in each period. During the scheduling process, each task is arranged to the destination with the minimum contribution to the objective. In order to evaluate the performance of the proposed algorithm, the proposed algorithm is compared to UAV-assisted Vehicular computation Cost



Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.



(a) Type one (b) Type two (c) Type three (d) Type four

Fig. 2. Road Distribution

Optimization (UVCO) algorithm based on the game theory. The experimental results show that the proposal outperforms UVCO on all instance combinations. The major reason may be that UVCO does not consider the workload already assigned to resources when arranging tasks.

In the considered scenarios, the vehicles are given and certain. In our future work, it is necessary to introduce more dynamic factors into the problem. For example, the scenarios with dynamic arrival vehicles would be more realistic.


## ACKNOWLEDGMENT

This work is sponsored by the State Key Laboratory of Computer Architecture (ICT,CAS) (Grant No. CARCHA202107), the National Science Foundation for Postdoctoral Scientists of China (Grant No. 2018M640510), the Natural Science Foundation of Jiangsu Province (Grant No. BK20201375), the National Natural Science Foundations of China (Grant Nos. 62072252).


## REFERENCES


[1] K. Ansari, Y. Feng and M. Tang, "A Runtime Integrity Monitoring Framework for Real-Time Relative Positioning Systems Based on GPS and DSRC," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 980-992, April 2015, doi: 10.1109/TITS.2014.2349011.

[2] Q. Luo, C. Li, T. H. Luan and Y. Wen, "Optimal Utility of Vehicles in LTE-V Scenario: An Immune Clone-Based Spectrum Allocation Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 5, pp. 1942-1953, May 2019, doi: 10.1109/TITS.2018.2850311.

[3] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao and C. Liu, "Routing in Internet of Vehicles: A Review," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 5, pp. 2339-2352, Oct. 2015, doi: 10.1109/TITS.2015.2423667.

[4] Y. Fan and N. Zhang, "A survey on software-defined vehicular networks," Journal of Computers, vol. 28, no. 4, pp. 236 - 244, 2017, doi:10.3966/199115592017062803025.

[5] H. Shakhatreh et al., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," in IEEE Access, vol. 7, pp. 48572-48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[6] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," IEEE Transactions on Wireless Communications, vol. 11, no. 6, pp. 1991 - 1995, 2012, doi: 10.1109/TWC.2012.041912.110912.

[7] C. Pan, Z. Wang, Z. Zhou and X. Ren, "Deep reinforcement learning-based URLLC-aware task offloading in collaborative vehicular networks," in China Communications, vol. 18, no. 7, pp. 134-146, July 2021, doi: 10.23919/JCC.2021.07.011.

[8] Du, F. R. Yu, X. Chu, J. Feng and G. Lu, "Computation Offloading and Resource Allocation in Vehicular Networks Based on Dual-Side Cost Minimization," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1079-1092, Feb. 2019, doi: 10.1109/TVT.2018.2883156.

[9] Y. Jang, J. Na, S. Jeong and J. Kang, "Energy-Efficient Task Offloading for Vehicular Edge Computing: Joint Optimization of Offloading and Bit Allocation," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9128785.

[10] P. Pourbaba, K. B. S. Manosha, S. Ali and N. Rajatheva, "Full-Duplex UAV Relay Positioning for Vehicular Communications with Underlay V2V Links," 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019, pp. 1-6, doi: 10.1109/VTC-Spring.2019.8746630.

[11] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma and Z. Liu, "A Novel Cost Optimization Strategy for SDN-Enabled UAV-Assisted Vehicular Computation Offloading," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3664-3674, June 2021, doi: 10.1109/TITS.2020.3024186.

[12] B. Liu, Z. Su and Q. Xu, "Game theoretical secure wireless communication for UAV-assisted vehicular Internet of Things," in China Communications, vol. 18, no. 7, pp. 147-157, July 2021, doi: 10.23919/JCC.2021.07.012.

[13] M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine and A. Ghrayeb, "Leveraging UAVs for Coverage in Cell-Free Vehicular Networks: A Deep Reinforcement Learning Approach," in IEEE Transactions on Mobile Computing, vol. 20, no. 9, pp. 2835-2847, 1 Sept. 2021, doi: 10.1109/TMC.2020.2991326.

[14] Y. He, D. Zhai, Y. Jiang and R. Zhang, "Relay Selection for UAV-Assisted Urban Vehicular Ad Hoc Networks," in IEEE Wireless Communications Letters, vol. 9, no. 9, pp. 1379-1383, Sept. 2020, doi: 10.1109/LWC.2020.2991037.

[15] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974-983, 1 April 2015, doi: 10.1109/TPDS.2014.2316834.




Authorized licensed use limited to: Yanshan University. Downloaded on December 15,2022 at 07:43:27 UTC from IEEE Xplore. Restrictions apply.