Vehicle Assisted Computing Offloading for Unmanned Aerial Vehicles in Smart City

Minghui Dai, Zhou Su, *Senior Member, IEEE*, Qichao Xu, and Ning Zhang, *Senior Member, IEEE*

***Abstract*—Smart city emerges a promising paradigm for improving operational efficiency of city and comfort of people. With embedded multi-sensors, Unmanned Aerial Vehicles (UAVs) hold great potential for collecting sensing data and providing social services in smart city. However, due to the limited battery lifetime and processing capacities of UAVs, the efficient offloading scheme of UAVs is urgently needed in smart city. Therefore, in this article, a vehicle-assisted computing offloading architecture for UAVs is proposed to improve offloading efficiency by harnessing the moving vehicles in smart city. We first develop an offloading model for UAVs to determine the offloading strategy. Next, to select the optimal vehicles for offloading, we formulate a matching scheme based on the preference lists of UAVs and vehicles to derive the optimal matching between UAVs and vehicles. After that, to improve the offloading efficiency and maximize the utilities of UAVs and vehicles, the transaction process of computing data between UAVs and vehicles is modeled as a bargaining game. Moreover, an offloading algorithm for UAVs and vehicles is proposed to obtain the optimal strategy. Finally, simulations are performed to validate the efficiency of the proposed offloading scheme. The results demonstrate that the proposed offloading scheme can significantly save resource and improve the utilities of UAVs and vehicles.**

***Index Terms*—Smart city, unmanned aerial vehicles (UAVs), computing offloading, matching scheme, bargaining game.**

I. Introduction

By incorporating Internet of Things (IoT) and computing facilities to collect and analyze data [1]–[4], smart city is promising for providing diverse services and improving the quality of experience (QoE) and comfort of residents. Particularly, with multi-sensors (e.g., on-board units, cameras, thermometer, etc.) embedded in unmanned aerial vehicles (UAVs), a large amount of sensing data in the city can be collected efficiently to improve the service efficiency of smart city [5]. Due to high mobility and flexibility characteristics, UAVs hold great potential to collect environmental information and make decisions in smart city [6]–[8]. For instance, with path planning and artificial intelligence techniques, UAVs can be scheduled for intelligent logistics to promote the efficiency of delivery. With line-of-sight (LoS) links, a group of UAVs can cooperatively collect traffic information for city service platform to regulate traffic streaming and improve travel efficiency. Besides, UAVs have the advantages for emergency rescuing and communication relief without human involved. Therefore, the wide utilization of UAVs paves the way to facilitate service effectiveness in smart city.

The ever-increasing sensor nodes deployed in smart city can lead to exponential growth in data volume [9], [10]. However, due to the limited computing resources of sensors, it is difficult to satisfy the demands of computation-intensive and delay-sensitive applications in smart city. The deployment of UAVs in smart city can reduce loads of sensor nodes through offloading data. In literature, some works have been proposed to improve the performance of UAV offloading in smart city. An efficient UAV-aided edge offloading scheme is proposed in [11] to meet the requirements for big data processing. A novel network framework that deploys UAVs as aerial mobile base stations is developed in [12] to offload data streaming for mobile devices. A computing-efficient UAV assisted edge computing scheme is presented in [13] to improve the offloading efficiency for energy-limited devices. Moreover, a time division multiple access based on work flow model is developed in [14] to achieve energy saving for UAVs. Therefore, the performance optimization in terms of computing offloading for smart city has been attracted from academia and industry [15]–[17].

The use of UAVs has remarkable advantages in improving the city services. However, the limited computing capacity and energy supply pose several challenges for the application of UAVs in smart city with the following reasons. a) Low endurance: due to the limited onboard battery in UAVs, the flying and hovering durations are short and incapable of providing long-term sensing and communications. This results in the poor quality of service (QoS) of UAVs in smart city [18]. b) Limited computing power: the computing power embedded in UAVs is constrained. The computation-intensive tasks processed in UAVs lead to high delay and low QoE of citizens [19]. c) Inefficient offloading: in conventional centralized architectures, the offloading tasks from UAVs are transferred to cloud servers or base stations. It brings high transmission delay and energy consumption. Besides, the computing efficiency is low when processing a large number of tasks simultaneously [20]. Therefore, an efficient offloading architecture is

Manuscript received March 15, 2020; revised July 28, 2020 and November 23, 2020; accepted December 3, 2020. Date of current version March 1, 2021. This work was supported in part by NSFC under Grant U1808207 and Grant U20A20175 and in part by the project of Shanghai Municipal Science and Technology Commission under Grant 18510761000. The Associate Editor for this article was H. Gao. *(Corresponding author: Zhou Su.)*

Minghui Dai, Zhou Su, and Qichao Xu are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: minghui_dai@shu.edu.cn; zhousu@ieee.org; xqc690926910@shu.edu.cn).

Ning Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: ning.zhang@uwindsor.ca).

Digital Object Identifier 10.1109/TITS.2021.3052979

needed to improve the effectiveness of UAV offloading in smart city.

With the advancement of vehicle technologies, vehicles which are endowed with powerful capacities (e.g., computing, communication, storage) can help alleviate the aforementioned issues in smart city [21]–[23]. Specially, when driving on the roads, vehicles can assist in task computing offloaded from UAVs to reduce the loads of cloud servers or roadside units (RSUs). Moreover, the LoS links between UAVs and vehicles can reduce the transmission delay and energy consumption to support high-quality services. Besides, in comparison with the fixed infrastructures, the mobility feature of vehicles can support high coverage offloading services for UAVs. Therefore, the cooperation between UAVs and vehicles is imperative to provide enough computing resources and improve the QoS in smart city [24]–[26].

In this article, we propose a novel vehicle-assisted computing offloading architecture for UAVs in smart city, which achieves the following three goals. 1) Energy efficiency: based on the energy consumption and computing delay, the optimal offloading strategy of UAV is determined to achieve resource saving. 2) Computing effectiveness: with powerful computing capacity of vehicles, the computing tasks of UAVs are offloaded to vehicles to improve the QoS in smart city. 3) Incentive scheme: to improve the offloading efficiency and maximize the utilities of UAVs and vehicles, the transaction process of computing data between UAVs and vehicles is modeled as a bargaining game. The optimal transaction strategy can be derived by analyzing the bargaining process.

In a nutshell, the main contributions of this work include:

- *Architecture for vehicle-assisted computing offloading:* By leveraging the computing capacity of moving vehicles, we propose a novel vehicle-assisted computing offloading architecture for UAVs in smart city. With the proposed architecture, the computing tasks of UAVs can be offloaded to vehicles to improve offloading efficiency and achieve resource saving.
- *Matching scheme for computing offloading:* To facilitate the computing offloading of UAVs, we present a matching scheme between UAVs and vehicles. Based on the preference lists of UAVs and vehicles, the optimal matching between UAVs and vehicles can improve the utilization of computing resources. Then, the matching algorithm for UAVs and vehicles is provided.
- *Incentive design:* To stimulate vehicles to perform computing tasks, we model the transaction process of computing data between UAVs and vehicles as the bargaining game. By determining the valuations of computing tasks, the optimal strategy is obtained to maximize the utilities of UAVs and vehicles. Then, the computing offloading algorithm between UAVs and vehicles is proposed.
- *Performance evaluation:* Simulations are conducted to verify the proposed offloading scheme. The results show that the proposed scheme can significantly improve offloading efficiency and resource saving compared with other conventional schemes.

The remainder of this article is organized as follows. Section II reviews the related work. Section III presents the system model. Section IV introduces the vehicle-assisted computing offloading scheme. Performance evaluations are provided in Section V, followed by conclusion and future work in Section VI.

II. Related Work

In this section, we review the related work on matching scheme, offloading scheme, and incentive design in smart city.

A. Matching Scheme in Smart City

A number of works have studied the matching scheme in smart city. Zeng *et al*. [27] modeled the assignment of electric vehicles to charging stations as a many-to-one matching game and proposed a stable matching algorithm. Zhang *et al*. [28] studied the pairing problem between data service operators and fog nodes using a many-to-many matching game. Wang *et al*. [29] modeled the resource allocation problem as a many-to-one matching game. Moreover, a coloring-based heuristic algorithm was designed to find a near-optimal solution. Wang *et al*. [30] proposed a distributed matching scheme between content helpers and content requesters to maximize the overall success rate of multi-hop. Duan *et al*. [31] proposed a subchannel assignment algorithm based on matching game to maximize the system capacity of UAV communication. The above works have discussed the matching theory to improve the effectiveness. However, few works have studied the matching scheme between UAVs and vehicles for computing offloading in smart city.

B. Offloading Scheme in Smart City

There are a number of studies on offloading scheme in smart city. Feng *et al*. [32] proposed an autonomous vehicular edge framework for edge computing to improve the computational capabilities of vehicles. By utilizing the neighboring vehicle's idle resource, Feng *et al*. [33] developed an online framework of vehicular edge computing to promote the computation capability for emerging vehicle applications. Messous *et al*. [34] developed a computation-offloading problem for UAVs to achieve the optimal performance between energy consumption, time delay and computation cost. Wu *et al*. [35] presented a coordination offloading scheme between base stations and UAVs through ground-to-air (G2A) link to improve the performance of cell-edge mobile users. Zhao *et al*. [36] developed a collaborative computation offloading scheme to optimize the computation offloading decision for automobiles in vehicular networks. However, the offloading schemes by utilizing the computing resources of vehicles in smart city are not effectively discussed in most of existing works.

C. Incentive Design in Smart City

A number of incentive schemes have been proposed for improving the services of smart city. Wang *et al*. [37] proposed a bargaining game-based negotiation scheme to effectively allocate tasks to machines based on their real-time status. Motlagh *et al*. [38] developed a bargaining game for fair

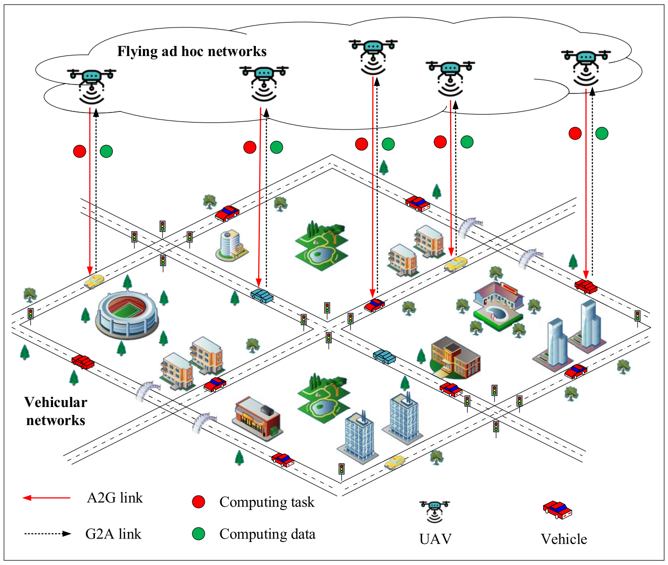


Fig. 1. Vehicle-assisted computing offloading architecture for UAVs in smart city.

tradeoff UAV selection to guarantee the fair tradeoff between energy consumption and operation time. Huang *et al*. [39] designed a Nash bargaining game for energy trading among nodes in software defined energy harvesting IoT to solve the benefit allocation problem. A bargaining game approach was studied by Zheng *et al*. [40] to balance the tradeoff for the energy efficiency of information transmission in mobile relay networks. Yang *et al*. [41] studied the interference among small cells and macrocells and developed a novel bargaining cooperative game to optimize the interference and energy management. However, few works have designed the incentive scheme for vehicles to improve the offloading efficiency in smart city.

III. System Model

In this section, we present the vehicle-assisted computing offloading model, including the UAV model, vehicle model, communication model and offloading model. Table I summarizes the key notations.

A. UAV Model

The vehicle-assisted computing offloading architecture for UAVs is shown in Fig. 1. The computing offloading process of UAVs consists of three phases: a) offloading tasks to vehicles through the air-to-ground (A2G) links, b) executing the offloading tasks by vehicles, and c) uploading results to UAVs via G2A links.

We consider a number of UAVs sensing and performing computing tasks in smart city. The set of UAVs is denoted by $\mathcal{U} = \{1, 2, \ldots u, \ldots, U\}$. Each UAV can execute at most one task at a time, which should be finished within a period of time. Due to the limited onboard battery, UAVs have limited computing capacities [42]. Let $\overline{a_u}^{\max}$ and $\overline{a_u}$ denote the maximum available computing resource and the occupied computing resource of UAV u, respectively. Then, the idle computing resource of UAV u can be calculated by

$$a_u = \overline{a_u}^{\max} - \overline{a_u}. \tag{1}$$

B. Vehicle Model

In the vehicular networks, vehicles with computing capacities can be utilized to execute a number of computing tasks [43]. The set of vehicles in vehicular networks is denoted by $\mathcal{N} = \{1, 2, \ldots n, \ldots, N\}$. Each vehicle operates in two states: the idle state and the busy state [44]. When a vehicle is in the idle state, it can execute the computing tasks offloaded from UAVs. Otherwise, the busy state indicates that the vehicle needs to select the computing tasks to execute according to its remaining computing capacity. The maximum available computing resource and the occupied computing resource of vehicle n are denoted by $\overline{a_n}^{\max}$ and $\overline{a_n}$, respectively. Then, the idle computing resource of vehicle n can be given by

$$a_n = \overline{a_n}^{\max} - \overline{a_n}. \tag{2}$$

C. Communication Model

There are a number of wireless channels in the networks, and the set is denoted by $\mathcal{W} = \{1, 2, \ldots w, \ldots, W\}$. The A2G links can be connected between UAVs and vehicles when executing task offloading. However, due to the mutual interference among A2G links, the data transmission rate between UAV and vehicle is affected by other UAVs [45]. The signal-to-interference-plus-noise ratio (SINR) between UAV u and vehicle n can be calculated by

$$\varsigma_{u,n} = \frac{p_u g_{u,n}}{\delta_0 + \sum_{i \in \mathcal{U}, i \neq u} p_i g_{i,n}}, \tag{3}$$

where p_u denotes the transmission power of UAV u. $g_{u,n}$ is the A2G link gain between UAV u and vehicle n, and δ_0 is the background noise power. $\sum_{i \in \mathcal{U}, i \neq u} p_i g_{i,n}$ denotes the interference from other UAVs except for UAV u. Eq. (3) indicates that if a number of UAVs simultaneously offload computing tasks to vehicle n via A2G link. It results in the high interference and low SINR between UAV u and vehicle n. Then, the available transmission rate between UAV u and vehicle n can be calculated by

$$r_{u,n} = w \log_2 \left(1 + \varsigma_{u,n}\right), \tag{4}$$

where w is the bandwidth of A2G link.

D. Offloading Model

When the computing resource of UAVs is insufficient, the computing tasks will be offloaded to vehicles. The set of computing tasks in the networks is denoted by $\mathcal{M} = \{1, 2, \ldots m, \ldots, M\}$. The characteristics of task m can be described as a triple $\{b_m, c_m, d_m\}$. b_m indicates the size of task m, and c_m denotes the required processor cycles to process task m. The time-to-live of task m is denoted by d_m, which means the emergency degree of task m that needs to be executed [46]. The remaining time for UAVs to offload task m can be denoted by ${d_m}^{rem} = d_m - d^{cur}$, where d^{cur} indicates the current time.

TABLE I
SUMMARY OF NOTATIONS

Symbols	Descriptions
$\mathcal{U}$	The set of UAVs in smart city.
$\mathcal{N}$	The set of vehicles in smart city.
$\mathcal{M}$	The set of computing tasks in smart city.
$\mathcal{W}$	The set of wireless channels.
$\mathcal{P}$	The set of preference lists for UAVs and vehicles.
ρ_u	The offloading proportion of UAV u to vehicles.
$\overline{a_u}^{\max}, \overline{a_u}$	The maximum available computing resource and occupied computing resource of UAV u, respectively.
a_u	The abundant idle computing resource of UAV u.
$\overline{a_n}^{\max}, \overline{a_n}$	The maximum available computing resource and occupied computing resource of vehicle n, respectively.
a_n	The abundant idle computing resource of vehicle n.
$\varsigma_{u,n}$	The signal-to-interference-plus-noise ratio when communicating between UAV u and vehicle n.
p_u	The transmission power of UAV u.
$g_{u,n}$	The A2G link gain between UAV u and vehicle n.
δ_0	The background noise power.
w	The bandwidth of A2G link.
$r_{u,n}$	The available transmission rate between UAV u and vehicle n.
${s_m}^O, {s_m}^L$	The size of offloading computing and local computing, respectively.
${\tau_n}^O$	The computing delay of vehicle n.
${\tau_u}^L$	The data transmission delay of UAV u.
${e_m}^L, {e_m}^O, {e_m}^T$	The energy consumption of local computing, offloading computing and data transmission, respectively.
${\varrho_u}^L$	The unit energy consumption of UAV u.
${\varrho_n}^O$	The unit energy consumption of vehicle n.
μ^O	The unit energy consumption of data transmission.
${t_m}^L, {t_m}^T, {t_m}^O$	The local computing delay, transmission delay and offloading computing delay, respectively.
$\mathcal{V}_u(\rho_u), \mathcal{V}_n(\rho_u)$	The valuations of computing data for UAV u and vehicle n, respectively.
$\mathcal{C}_m$	The difference of the reserve price between UAV u and vehicle n.
$\mathcal{U}_u(\mathcal{C}_m), \mathcal{U}_n(\mathcal{C}_m)$	The utilities of UAV u and vehicle n, respectively.
$\gamma_{u,m}, \gamma_{n,m}$	The offers of UAV u and vehicle n for the cake, respectively.
$\varepsilon_u, \varepsilon_n$	The discount factors of UAV u and vehicle n, respectively.
$\mathcal{Q}_u(\gamma_{u,m}), \mathcal{Q}_n(\gamma_{n,m})$	The profits of UAV u and vehicle n, respectively.

Let $x_u\ (x_u \in 0, 1)$ denote the offloading strategy of UAV u. $x_u = 1$ indicates that UAV u offloads its tasks to vehicles. Otherwise, the tasks will be executed by itself. Moreover, each UAV can dynamically adjust the offloading proportion for task computing [47]. Let ρ_u indicate the offloading proportion of UAV u and $\rho_u \in [0, 1]$. Then, the size of offloading computing and local computing can be calculated by

$$ {s_m}^O = b_m \rho_u, \tag{5} $$

$$ {s_m}^L = b_m (1 - \rho_u). \tag{6} $$

Each UAV can coordinate the size of offloading computing (i.e., ${s_m}^O$) and local computing (i.e., ${s_m}^L$) to finish its tasks.

The design goals are as follows: on one hand, by taking the computing resource of vehicles into account, the proposed matching scheme between UAVs and vehicles should be efficiently designed to improve computing efficiency. On the other hand, based on the valuations of computing data for UAVs and vehicles, the proposed incentive scheme should be effectively studied to maximize the utilities of UAVs and vehicles.

IV. VEHICLE-ASSISTED COMPUTING OFFLOADING SCHEME

This section first introduces the matching scheme between UAVs and vehicles. Next, the computing offloading scheme based on bargaining game is designed. Finally, the optimal computing offloading strategy is derived.

A. Matching Scheme Between UAVs and Vehicles

When UAV executes task offloading, it will select the optimal vehicle to offload its computing tasks. The selection process between UAVs and vehicles can be formulated as a matching game [28]. Specifically, the UAV selects the optimal vehicle based on its preference for the computing delay, and the vehicle chooses the optimal UAV according to its preference for the data transmission rate.

The matching game can be described as a triple $\Phi = \{\mathcal{U}, \mathcal{N}, \mathcal{P}\}$. The players (i.e., UAVs and vehicles) in the game are the sets $\mathcal{U}$ and $\mathcal{N}$. Here, $\mathcal{U} \cap \mathcal{N} = \emptyset$. $\mathcal{P}$ is the set of preference lists for UAVs and vehicles. Each player in set $\mathcal{U}(\mathcal{N})$ has a preference for each player in set $\mathcal{N}(\mathcal{U})$. Therefore, a one-to-one matching construction Φ can be expressed as: $\Phi(u) = n, \forall u \in \mathcal{U}, n \in \mathcal{N}$ indicates that UAV u is matched with vehicle n. However, $\Phi(u) = u, \forall u \in \mathcal{U}$ means that UAV u does not match with any vehicles.

To implement the two-sided matching, each UAV constructs its preference list by ranking vehicles according to the preference value [48]. For UAV u, the preference is inversely proportional to the computing delay of vehicle, which is denoted by

$$ P(u)|_{\Phi(u)=n} = \frac{1}{{\tau_n}^O}, \tag{7} $$

where ${\tau_n}^O$ denotes the computing delay of vehicle n, which can be calculated by

$$ {\tau_n}^O = b_m \rho_u a_n. \tag{8} $$

Eq. (8) indicates that the low computing delay of vehicle will be matched by a UAV with high priority.

For vehicle n, the preference is inversely proportional to the data transmission delay of UAV, which is denoted by

$$P(n)|_{\Phi(n)=u} = \frac{1}{{\tau_u}^L}, \tag{9}$$

where ${\tau_u}^L$ represents the data transmission delay of UAV u, which can be calculated by

$${\tau_u}^L = \frac{b_m \rho_u}{r_{u,n}}. \tag{10}$$

From Eq. (10), we can obtain that the low transmission delay of UAV will be matched by a vehicle with high priority.

Let $\succ$ denote the preference relations of UAVs and vehicles. Here, $P(u)|_{\Phi(u)=n} > P(u)|_{\Phi(u)=n'}$, $n \succ n'$ means that UAV u prefers vehicle n to vehicle n'. $P(n)|_{\Phi(n)=u} > P(n)|_{\Phi(n)=u'}$, $u \succ u'$ indicates that vehicle n prefers UAV u to UAV u'. We set the preferences of UAVs and vehicles by a descending order based on $P(u)|_{\Phi(u)=n}$ and $P(n)|_{\Phi(n)=u}$. Therefore, the set of preference lists can be obtained by $\mathcal{P} = \left\{ P(u)|_{\Phi(u)=n};\ P(n)|_{\Phi(n)=u} \right\}, \forall u \in \mathcal{U}, n \in \mathcal{N}$.

Matching rule: a) For UAV $u \in \mathcal{U}$, it proposes to match its most preferred vehicle in the preference list $P(u)|_{\Phi(u)=n}$. b) For vehicle $n \in \mathcal{N}$, it proposes to match its most preferred UAV in the preference list $P(n)|_{\Phi(n)=u}$.

If UAVs cannot match with any vehicles to offload its tasks, the computing tasks will be offloaded to base stations through A2G links. The base stations can assist task computing for UAVs, and then the computing data will be transmitted to UAVs. Algorithm 1 shows the matching algorithm for UAVs and vehicles. Specifically, the matching process consists of two stages: 1) Matching preference initialization. 2) Iterative matching. In stage 1, the preference lists of UAVs and vehicles are first calculated, respectively. In stage 2, the optimal matching between UAVs and vehicles is obtained through a number of iterations.

Theorem 1: The proposed matching scheme is stable. For any UAV $u \in \mathcal{U}$, there does not exist a vehicle such that $n \succ n'$.

Proof: The contradiction method is adopted to validate the stability of the proposed matching scheme. Assuming that UAV u and vehicle n have no matching in the matching result $\Phi(u) = n$. However, they prefer to match with each other. It means that the matching result $\Phi(u) = n$ is restricted by the participants u and n. Since the rank of vehicle n in the preference list of UAV u is higher than that of the matching Φ assigned to UAV n, i.e., $n \succ \Phi(u)$. Therefore, UAV u must have sent a matching request to vehicle n before the matching is completed. However, they do not match with each other at the end of matching. It means that vehicle n must have rejected the request of UAV u during the matching process. The matching result of vehicle n in the matching Φ is at least as well as that of UAV u. Therefore, the matching result $\Phi(u) = n$ is not restricted by the participants u and n. This contradicts the assumption, and the matching is stable. This completes our proof.

Algorithm 1: Matching Algorithm for UAVs and Vehicles

Input: $u \in \mathcal{U}$, $n \in \mathcal{N}$, $\overline{{a_u}^{\max}}$, $\overline{a_u}$, $\overline{{a_n}^{\max}}$, $\overline{a_n}$, b_m, ${\tau_n}^O$, ${\tau_u}^L$, $r_{u,n}$, ρ_u, Θ [2];
Output: $\Phi(u) = n$, $\Phi(n) = u$;
Stage 1: Matching Preference Initialization
for $u \in \mathcal{U}$ **do**
  Calculate $P(u)|_{\Phi(u)=n}$ by using (7);
end for
for $n \in \mathcal{N}$ **do**
  Calculate $P(n)|_{\Phi(n)=u}$ by using (9);
end for
Sort the preferences of UAVs and vehicles by a descending order;
Set $\Phi = \emptyset$, $\Theta = \emptyset$;
Stage 2: Iterative Matching
while $\Phi(u) \neq \emptyset || P(u)|_{\Phi(u)=n} \neq \emptyset$ **do**
  for $u \in \mathcal{U}$ **do**
    UAV u proposes to its most preferred vehicle according to $P(u)|_{\Phi(u)=n}$;
  end for
  for $n \in \mathcal{N}$ **do**
    if $\Phi(u) = n$ **then**
      if $|\Theta| > 1 || P(n)|_{\Phi(n)=u'}$ **then**
        $\Phi(n) = u'$;
      else
        $\Phi(u) = n$;
      end if
    else
      if $|\Theta| > 1 || P(n)|_{\Phi(n)=u}$ **then**
        $\Phi(n) = u$;
      else
        $\Phi(u) = u$;
      end if
    end if
    Update: remove vehicle n from preference list $P(u)|_{\Phi(u)=n}$;
  end for
  Until no UAVs propose to match with vehicles;
end while

Theorem 2: The obtained stable matching is optimal.

Proof: We use the contradiction method to prove the optimality of the stable matching. Assuming that there exists another optimal matching result Φ', where Φ' is superior to Φ. For UAVs u' and u'', UAV u' is matched with vehicle n' in the matching Φ'. While UAV u'' is matched with vehicle n' in the matching Φ. We can obtain that vehicle n' prefers UAV u' to u''. However, if UAV u' is matched with n'' in the matching result Φ', apparently, vehicle n' prefers UAV u', and UAV u' prefers vehicle n'. We can obtain $u' \succ u''$ and $n' \succ n''$. Based on Theorem 1, the matching result Φ' is restricted by participants u' and n'. The matching result Φ' is not stable, which contradicts the assumption. Therefore, there

[2]The set of vehicles which receive matching proposal.

does not exist the matching result Φ'. The matching result Φ is optimal. This completes our proof.

B. Computing Offloading Scheme Based on Bargaining Game

After UAVs are matched with the optimal vehicles to offload its computing tasks, the optimal offloading strategy is analyzed. Here, the energy efficiency and delay efficiency are used to evaluate the effectiveness of offloading, which are introduced as follows.

Energy efficiency: The energy efficiency is divided into three aspects: the energy consumption of local computing, offloading computing and data transmission. The energy efficiency can be calculated by

$$\theta_1{}^{over} = e_m{}^L + e_m{}^O + e_m{}^T, \tag{11}$$

where $e_m{}^L$, $e_m{}^O$ and $e_m{}^T$ denote the energy consumption of local computing, offloading computing and data transmission, respectively.

The energy consumption of local computing is related to the size of task m and the energy consumption of processor for UAV u, which can be calculated by

$$e_m{}^L = b_m\left(1-\rho_u\right) a_u \varrho_u{}^L, \tag{12}$$

where $\varrho_u{}^L$ indicates the unit energy consumption of UAV u.

The energy consumption of offloading computing is associated with the task size of offloading and the energy consumption of processor for vehicle n, which can be calculated by

$$e_m{}^O = b_m \rho_u a_n \varrho_n{}^O, \tag{13}$$

where $\varrho_n{}^O$ represents the unit energy consumption of vehicle n.

Let μ^O denote the unit energy consumption of data transmission. The energy consumption of data transmission can be calculated by

$$e_m{}^T = b_m \rho_u \mu^O. \tag{14}$$

Delay efficiency: The delay efficiency consists of three parts: the local computing delay, the transmission delay and the offloading computing delay. The delay efficiency can be calculated by

$$\theta_2{}^{over} = t_m{}^L + t_m{}^T + t_m{}^O, \tag{15}$$

where $t_m{}^L$, $t_m{}^T$ and $t_m{}^O$ indicate the local computing delay, transmission delay and offloading computing delay, respectively.

The local computing delay is related to the task size and the idle computing resource of UAV u, which can be calculated by

$$t_m{}^L = b_m\left(1-\rho_u\right) a_u. \tag{16}$$

The transmission delay can be calculated by the task size of offloading and the data transmission rate, i.e.,

$$t_m{}^T = \frac{b_m \rho_u}{r_{u,n}}. \tag{17}$$

The offloading computing delay is associated with the task size of offloading and the idle computing resource of vehicle n, which can be calculated by

$$t_m{}^O = b_m \rho_u a_n. \tag{18}$$

When UAVs offload their computing tasks to vehicles, the transaction process of computing data between UAVs and vehicles is modeled as a bargaining game. UAVs (i.e., buyers) buy computing data from vehicles and pay payments. Vehicles (i.e., sellers) sell their computing data and charge from UAVs. Each participant pursues the maximum utility during the transaction. The description of the bargaining game can be defined as follows:

- Players: The set of UAVs $\mathcal{U}$, and the set of vehicles $\mathcal{N}$.
- Strategies: The offloading proportion ρ_u $(0 \leq \rho_u \leq 1)$.
- Utilities: The utility of each UAV is calculated based on its satisfaction degree. The utility of each vehicle is determined by its valuation of computing data.
- Equilibrium: None of players can increase their utility by changing its strategies.

For UAV u (i.e., buyer), the reserve price of computing data indicates the highest price that can be accepted by the UAV. The reserve price of UAV u can be determined based on the valuations of transmission delay and offloading computing delay, which can be calculated by

$$\mathcal{V}_u\left(\rho_u\right) = \begin{cases} \lambda_u \log_2\left(1 + \frac{\overline{t_m{}^T}}{t_m{}^T} + \frac{\overline{t_m{}^O}}{t_m{}^O}\right), & x_u = 1, \\ 0, & x_u = 0, \end{cases} \tag{19}$$

where λ_u is the adjustment parameter. $\overline{t_m{}^T}$ and $\overline{t_m{}^O}$ are the maximum transmission delay and maximum offloading computing delay, respectively. From (19), we can obtain that if the transmission delay and offloading computing delay are high, the valuation of computing data will be low, resulting in the low reserve price of UAV u.

For vehicle n (i.e., seller), the reserve price of computing data denotes the lowest price that can be accepted by the vehicle. The reserve price of vehicle n can be determined based on the valuations of energy consumption and offloading computing delay. We can calculate by

$$\mathcal{V}_n\left(\rho_u\right) = \begin{cases} \xi_n e_m{}^O + \zeta_n \frac{1}{t_m{}^O}, & x_u = 1, \\ 0, & x_u = 0, \end{cases} \tag{20}$$

where ξ_n and ζ_n are the adjustment parameters. From (20), we can obtain that if the energy consumption of vehicle n is high, the valuation of computing data is large. Moreover, if the offloading computing delay is low, the valuation of offloading data is large, resulting in the high reserve price of vehicle n.

Based on the valuation of UAV u and vehicle n (i.e., $\mathcal{V}_u\left(\rho_u\right)$, $\mathcal{V}_n\left(\rho_u\right)$), we have the following three cases:

Case 1: $\mathcal{V}_u\left(\rho_u\right) < \mathcal{V}_n\left(\rho_u\right)$. The reserve price of UAV u is lower than that of vehicle n. UAV u and vehicle n fail to reach an agreement for offloading computing. The transaction will be cancelled. To facilitate the transaction between UAV and vehicle, the UAV u should increase the valuation of computing data. The vehicle n should reduce the valuation of computing data.

Case 2: $\mathcal{V}_u(\rho_u) = \mathcal{V}_n(\rho_u)$. The reserve price of UAV u is equal to that of vehicle n. The UAV u and vehicle n reach an agreement for computing data. In this case, the reserve price $\mathcal{V}_u(\rho_u)$ is paid to vehicle n.

Case 3: $\mathcal{V}_u(\rho_u) > \mathcal{V}_n(\rho_u)$. The reserve price of UAV u is higher than that of vehicle n. Since each participant wants to obtain more utilities from the transaction, the optimal transaction price can be derived by analyzing the bargaining game between UAV u and vehicle n.

In case 3, the difference of the reserve price between UAV u and vehicle n can be calculated by

$$\mathcal{C}_m = \mathcal{V}_u(\rho_u) - \mathcal{V}_n(\rho_u). \tag{21}$$

Here, $\mathcal{C}_m$ can be regarded as a cake of the valuation for computing data. Each participant pursues more offers from the cake to maximize its utility, and the utilities of UAV u and vehicle n can be calculated by

$$\mathcal{U}_u(\mathcal{C}_m) = \gamma_{u,m}\mathcal{C}_m = \gamma_{u,m}\left(\mathcal{V}_u(\rho_u) - \mathcal{V}_n(\rho_u)\right), \tag{22}$$

$$\mathcal{U}_n(\mathcal{C}_m) = \gamma_{n,m}\mathcal{C}_m = \gamma_{n,m}\left(\mathcal{V}_u(\rho_u) - \mathcal{V}_n(\rho_u)\right), \tag{23}$$

where $\gamma_{u,m}, \gamma_{n,m} \in [0, 1]$ are the offers of UAV u and vehicle n for the cake, respectively. We have $\gamma_{u,m} + \gamma_{n,m} = 1$. The partition of the cake for UAV u and vehicle n can be described as

$$\gamma_m = \left\{\left(\gamma_{u,m}, \gamma_{n,m}\right) \middle| \gamma_{u,m} + \gamma_{n,m} = 1\right\}. \tag{24}$$

Therefore, the profits of UAV u and vehicle n can be calculated by

$$\mathcal{Q}_u\left(\gamma_{u,m}\right) = \mathcal{V}_u(\rho_u) - \mathcal{U}_u(\mathcal{C}_m), \tag{25}$$

$$\mathcal{Q}_n\left(\gamma_{n,m}\right) = \mathcal{V}_n(\rho_u) + \mathcal{U}_n(\mathcal{C}_m). \tag{26}$$

C. Optimal Computing Offloading Strategy

In the bargaining game, UAV u and vehicle n provide their offers in turn to determine the optimal partition (i.e., $\gamma_{u,m}{}^*, \gamma_{n,m}{}^*$) of the cake [49]. Without loss of generality, we consider that UAV u first provides its offer. The bargaining process between UAV u and vehicle n can be described as follows: In the first round, UAV u provides the offer $\gamma_m{}^1 = \left(\gamma_{u,m}{}^1, \gamma_{n,m}{}^1\right)$. Here, $\gamma_{u,m}{}^1 + \gamma_{n,m}{}^1 = 1$. If vehicle n accepts the offer, the bargaining game is ended with the offer $\gamma_m{}^1$. The utilities of UAV u and vehicle n can be calculated by $\mathcal{U}_u(\mathcal{C}_m)^1 = \gamma_{u,m}{}^1\mathcal{C}_m$ and $\mathcal{U}_n(\mathcal{C}_m)^1 = \gamma_{n,m}{}^1\mathcal{C}_m$, respectively. Otherwise, the transaction fails to reach an agreement. The bargaining process enters the next round. In the second round, the vehicle n provides its offer $\gamma_m{}^2 = \left(\gamma_{u,m}{}^2, \gamma_{n,m}{}^2\right)$. Here, $\gamma_{u,m}{}^2 + \gamma_{n,m}{}^2 = 1$. If UAV u accepts the offer, the bargaining game is ended with the offer $\gamma_m{}^2$. The utilities of UAV u and vehicle n can be calculated by $\mathcal{U}_u(\mathcal{C}_m)^2 = \gamma_{u,m}{}^2\mathcal{C}_m$ and $\mathcal{U}_n(\mathcal{C}_m)^2 = \gamma_{n,m}{}^2\mathcal{C}_m$, respectively. Otherwise, the transaction fails to reach an agreement. The bargaining process enters the next round. The bargaining game is ended until one of the players accepts the offer provided by another player. In the odd round, the offer is determined by the UAV u. In the even round, the offer is provided by the vehicle n.

The discount factor ε $(0 \le \varepsilon \le 1)$ is introduced to analyze the bargaining game, which is related to the patience degree of the players (i.e., UAVs and vehicles). The patience value represents the economic endurance of the players, and different players have different economic endurance in the negotiation. Due to the effect of discount factor in bargaining game, the utility achieved by the players with the offer $\left(\gamma_{u,m}{}^k, \gamma_{n,m}{}^k\right)$ in round k is not equal to the utility achieved by the players with the same offer in the next round $k+1$. The utility achieved by the players in the next round is lower than that of the current round. Therefore, the players (i.e., UAV u and vehicle n) should accept the offer provided by another player as soon as possible to obtain more utilities.

The discount factor can be formulated as the function of the player's patience. For UAV u, the discount factor is related to the offloading computing delay, which can be calculated by

$$\varepsilon_u = \frac{1}{(1+\psi_n)^{\upsilon_u}}, \tag{27}$$

where υ_u denotes the coefficient of patience degree for UAV u, which is related to the economic endurance (i.e., valuation of computing data) of UAV u. $\psi_n = t_m{}^O$ indicates the offloading computing delay of vehicle n. If ψ_n is large, the patience degree of UAV u is low, resulting in the low discount factor for UAV u.

For vehicle n, the discount factor is associated with the data transmission delay, which can be calculated by

$$\varepsilon_n = \frac{1}{(1+\psi_u)^{\upsilon_n}}, \tag{28}$$

where υ_n denotes the coefficient of patience degree for vehicle n, which is associated with the economic endurance (i.e., valuation of computing data) of vehicle n. $\psi_u = t_m{}^T$ is the data transmission delay of UAV u. If ψ_u is large, the patience degree of vehicle n is low, resulting in the low discount factor for vehicle n.

Based on the discount factors, the utilities of UAV u and vehicle n in the round k can be calculated by

$$\mathcal{U}_u(\mathcal{C}_m)^k = \varepsilon_u{}^{k-1}\gamma_{u,m}{}^k\mathcal{C}_m, \tag{29}$$

$$\mathcal{U}_n(\mathcal{C}_m)^k = \varepsilon_n{}^{k-1}\gamma_{n,m}{}^k\mathcal{C}_m. \tag{30}$$

From (29) and (30), we can find that both UAV u and vehicle n should reach an agreement as soon as possible in the bargaining game to gain more utilities.

Theorem 3: There exists the Nash equilibrium in the proposed bargaining game. When vehicle n first provides its offer, the optimal partition is given by

$$\begin{cases} \gamma_{n,m}{}^* = \dfrac{1-\varepsilon_u}{1-\varepsilon_u\varepsilon_n}, \\ \gamma_{u,m}{}^* = \dfrac{\varepsilon_u-\varepsilon_u\varepsilon_n}{1-\varepsilon_u\varepsilon_n}. \end{cases} \tag{31}$$

When UAV u first provides its offer, the optimal partition is given by

$$\begin{cases} \gamma_{u,m}{}^* = \dfrac{1-\varepsilon_n}{1-\varepsilon_u\varepsilon_n}, \\ \gamma_{n,m}{}^* = \dfrac{\varepsilon_n-\varepsilon_u\varepsilon_n}{1-\varepsilon_u\varepsilon_n}. \end{cases} \tag{32}$$

Proof: Without loss of generality, we consider that vehicle n first provides its offer to validate the optimal partition. Since the offer provided by vehicle n is in the odd rounds, the infinite repeated bargaining game between vehicle n and UAV u can be modeled as three rounds bargaining game [49]. The backward induction method is adopted to determine the optimal partition.

In the third round, let ${\gamma_{n,m}}^*$ denote the optimal offer provided by vehicle n. We have

$$\begin{aligned} {\gamma_{n,m}}^3 &= {\gamma_{n,m}}^*, &(33)\\ {\gamma_{u,m}}^3 &= 1 - {\gamma_{n,m}}^*. &(34)\end{aligned}$$

The utilities of vehicle n and UAV u in the third round can be calculated by

$$\begin{aligned} \mathcal{U}_n(\mathcal{C}_m)^3 &= {\varepsilon_n}^2 {\gamma_{n,m}}^* \mathcal{C}_m. &(35)\\ \mathcal{U}_u(\mathcal{C}_m)^3 &= {\varepsilon_u}^2 \left(1 - {\gamma_{n,m}}^*\right) \mathcal{C}_m. &(36)\end{aligned}$$

In the second round, the offer is provided by UAV u. Let ${\gamma_{n,m}}'$ denote the offer provided by UAV u, the offer can be denoted by

$$\begin{aligned} {\gamma_{u,m}}^2 &= {\gamma_{u,m}}'. &(37)\\ {\gamma_{n,m}}^2 &= 1 - {\gamma_{u,m}}'. &(38)\end{aligned}$$

The utilities of UAV u and vehicle n in the second round can be calculated by

$$\begin{aligned} \mathcal{U}_u(\mathcal{C}_m)^2 &= \varepsilon_u {\gamma_{u,m}}' \mathcal{C}_m. &(39)\\ \mathcal{U}_n(\mathcal{C}_m)^2 &= \varepsilon_n \left(1 - {\gamma_{u,m}}'\right) \mathcal{C}_m. &(40)\end{aligned}$$

With the individual rational constraint, if $\mathcal{U}_n(\mathcal{C}_m)^2 \geq \mathcal{U}_n(\mathcal{C}_m)^3$, vehicle n will accept the offer provided by UAV u in the second round. We can obtain ${\gamma_{u,m}}' \leq 1 - \varepsilon_n {\gamma_{u,m}}^*$. Otherwise, the bargaining game will enter the third round. The utility of UAV u can be calculated by

$$\mathcal{U}_u(\mathcal{C}_m) = \begin{cases} \varepsilon_u {\gamma_{u,m}}' \mathcal{C}_m, & \text{if } {\gamma_{u,m}}' \leq 1 - \varepsilon_n {\gamma_{u,m}}^* \\ {\varepsilon_u}^2 \left(1 - {\gamma_{n,m}}^*\right) \mathcal{C}_m, & \text{otherwise.} \end{cases} \quad (41)$$

In the first round, the offer is provided by vehicle n. Considering the individual rational constraint, the offer is equal to the third round to maximize its utility. We have

$$\begin{aligned} {\gamma_{n,m}}^1 &= {\gamma_{n,m}}^*. &(42)\\ {\gamma_{u,m}}^1 &= 1 - {\gamma_{n,m}}^*. &(43)\end{aligned}$$

The utilities of vehicle n and UAV u in the first round can be calculated by

$$\begin{aligned} \mathcal{U}_n(\mathcal{C}_m)^1 &= {\gamma_{n,m}}^* \mathcal{C}_m. &(44)\\ \mathcal{U}_u(\mathcal{C}_m)^1 &= \left(1 - {\gamma_{n,m}}^*\right) \mathcal{C}_m. &(45)\end{aligned}$$

With the individual rational constraint, if $\mathcal{U}_u(\mathcal{C}_m)^1 \geq \mathcal{U}_u(\mathcal{C}_m)^2$, UAV u will accept the offer provided by vehicle n in the first round. We can obtain ${\gamma_{n,m}}^* \leq 1 - \varepsilon_u {\gamma_{u,m}}'$. Otherwise, the bargaining game will enter the second round.

Therefore, based on ${\gamma_{u,m}}' \leq 1 - \varepsilon_n {\gamma_{u,m}}^*$ and ${\gamma_{n,m}}^* \leq 1 - \varepsilon_u {\gamma_{u,m}}'$, we can obtain the optimal offer provided by vehicle n as follows

$${\gamma_{n,m}}^* = \frac{1 - \varepsilon_u}{1 - \varepsilon_u \varepsilon_n}. \quad (46)$$

The optimal offer provided by UAV u can be calculated by

$${\gamma_{u,m}}^* = \frac{\varepsilon_u - \varepsilon_u \varepsilon_n}{1 - \varepsilon_u \varepsilon_n}. \quad (47)$$

Similarly, when UAV u first provides its offer, the optimal partition can be derived based on the similar analysis method. This completes our proof.

Based on the optimal partition, the profits that UAV u and vehicle n can acquire from the bargaining game are calculated by

$$\begin{aligned} \mathcal{Q}_u\left({\gamma_{u,m}}^*\right) &= \mathcal{V}_u\left(\rho_u\right) - {\gamma_{u,m}}^* \mathcal{C}_m, &(48)\\ \mathcal{Q}_n\left({\gamma_{n,m}}^*\right) &= \mathcal{V}_n\left(\rho_u\right) - {\gamma_{n,m}}^* \mathcal{C}_m. &(49)\end{aligned}$$

The computing offloading algorithm between UAVs and vehicles based on bargaining game is shown in Algorithm 2. UAVs first match with vehicles according to Algorithm 1. Next, the energy efficiency and delay efficiency are calculated to evaluate the valuation of computing data. After that, the reserve price of UAV u and vehicle n are obtained, respectively. Finally, the optimal offers for computing data between UAV u and vehicle n are derived.

V. Performance Evaluations

This section reports on simulations to evaluate the performance of the proposed offloading scheme. The simulation setup is first introduced, followed by numerical results and analysis.

A. Simulation Setup

In the simulations, the flying range of UAVs in smart city is set to be 500m × 500m × 500m. The number of UAVs is set to be 20, and the number of vehicles is set as 20. The size of computing task is selected from the interval [1, 5] Mb [50]. The number of computing tasks is set as [50, 100]. The transmission power of UAVs is selected from the interval [152, 190] mW [51]. When executing task offloading, the A2G links are established between UAVs and vehicles. The offloading proportion of UAVs follows the uniform distribution within [0, 1]. The abundant idle computing resource of UAVs is determined by the interval [0.5, 1] GHz. The abundant idle computing resource of vehicles is chosen from [0.5, 10] GHz. Other parameters in the simulations are given in Table II.

The proposed offloading scheme is evaluated in terms of the following criteria: the valuation of computing data, the average overhead, and the utilities of UAVs and vehicles. We compare the proposed offloading scheme with other conventional schemes, which are listed as follows:

- *The distributed computation offloading strategy (DCOS) [34]:* The computation offloading strategy is modeled as a non-cooperative game to derive the strategy profile of player through analyzing the utility function.
- *The offloading computing scheme (OCS):* The computing tasks are totally offloaded to vehicles to execute. The energy consumption, transmission delay and computing delay will generate the extra overheads for vehicles.
- *The local computing scheme (LCS):* The computing tasks are executed locally by UAVs. The energy consumption

Algorithm 2: Computing Offloading Algorithm Between UAVs and Vehicles

Input: $u \in \mathcal{U}$, $n \in \mathcal{N}$, b_m, $r_{u,n}$, λ_u, ξ_n, ζ_n, ${\varrho_u}^L$, ${\varrho_n}^O$, μ^O, ${t_m}^O$, ${t_m}^T$;
Output: ρ_u, ${\gamma_{u,m}}^*$, ${\gamma_{n,m}}^*$;
Initialization: UAVs match with vehicles based on Algorithm 1;
for $u \in \mathcal{U}, n \in \mathcal{N}$ **do**
  Calculate the energy efficiency and delay efficiency based on (11) and (15), respectively;
end for
for $u \in \mathcal{U}$ **do**
  Calculate the reserve price $\mathcal{V}_u(\rho_u)$ based on (19);
end for
for $n \in \mathcal{N}$ **do**
  Calculate the reserve price $\mathcal{V}_n(\rho_u)$ based on (20);
end for
for $u \in \mathcal{U}, n \in \mathcal{N}$ **do**
  if $\mathcal{V}_u(\rho_u) > \mathcal{V}_n(\rho_u)$ **then**
    Calculate the difference of the reserve price $\mathcal{C}_m$ based on (21);
    if vehicle n first provides its offer **then**
      ${\gamma_{n,m}}^* = \frac{1-\varepsilon_u}{1-\varepsilon_u\varepsilon_n}$;
      ${\gamma_{u,m}}^* = \frac{\varepsilon_u-\varepsilon_u\varepsilon_n}{1-\varepsilon_u\varepsilon_n}$;
    else
      if UAV u first provides its offer **then**
        ${\gamma_{u,m}}^* = \frac{1-\varepsilon_n}{1-\varepsilon_u\varepsilon_n}$;
        ${\gamma_{n,m}}^* = \frac{\varepsilon_n-\varepsilon_u\varepsilon_n}{1-\varepsilon_u\varepsilon_n}$;
      end if
    end if
  else
    if $\mathcal{V}_u(\rho_u) = \mathcal{V}_n(\rho_u)$ **then**
      The reserve price $V_u(\rho_u)$ is paid to vehicle n by UAV u;
    end if
  else
    if $\mathcal{V}_u(\rho_u) < \mathcal{V}_n(\rho_u)$ **then**
      The transaction between UAV u and vehicle n is cancelled;
    end if
  end if
end for
Return: ${\gamma_{u,m}}^*$, ${\gamma_{n,m}}^*$;

and computing delay will generate the extra overheads for UAVs.

B. Numerical Results and Analysis

Fig. 2 shows the valuation of computing data with different number of UAVs. The x-axis plots the number of UAVs increasing from 1 to 20, while the y-axis plots the valuation of computing data. From Fig. 2, we can see that as the number of UAVs increases, the valuation of computing data for UAVs and vehicles increases. Moreover, the valuation of computing data for UAVs is higher than that of vehicles with the increase of the number of UAVs. This is because when the number of UAVs increases, more computing tasks are offloaded to vehicles. Therefore, when the number of UAVs increases, the valuation of computing data for UAVs and vehicles increases. On the other hand, by offloading computing tasks to vehicles, the valuation of computing data for UAVs is high due to the low computing delay.

Fig. 3 shows the valuation of computing data with different offloading proportions (i.e., ρ_u). The x-axis plots the

TABLE II
SIMULATION PARAMETERS

Parameters	Values
δ_0	-100dBm
$g_{u,n}$	1dBi
w	10kHz
${\varrho_u}^L$	4mW
${\varrho_n}^O$	1mW
μ^O	1mW
λ_u	2
ξ_n	0.8
ζ_n	0.2

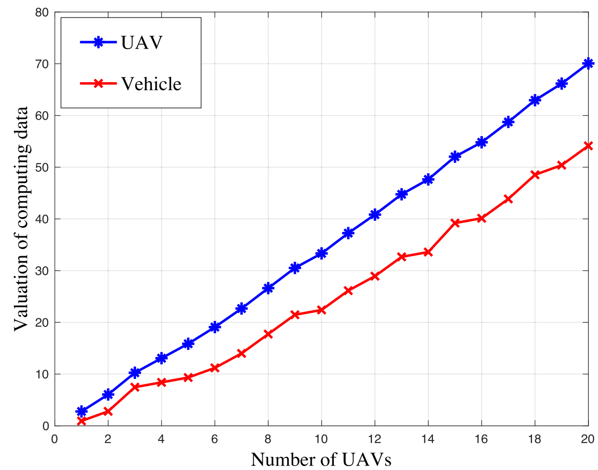


Fig. 2. The valuation of computing data with different number of UAVs.

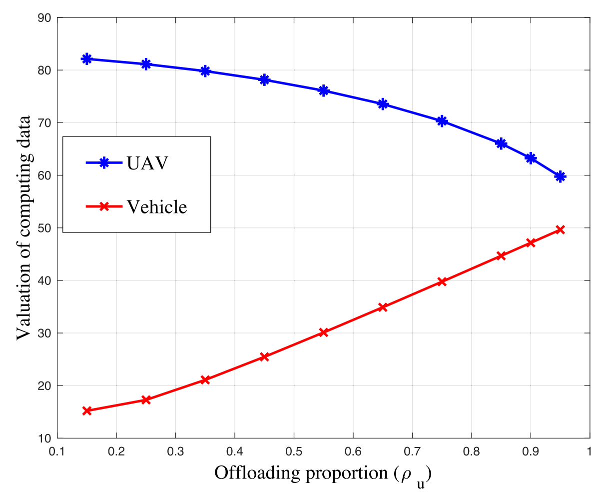


Fig. 3. The valuation of computing data with different offloading proportions.

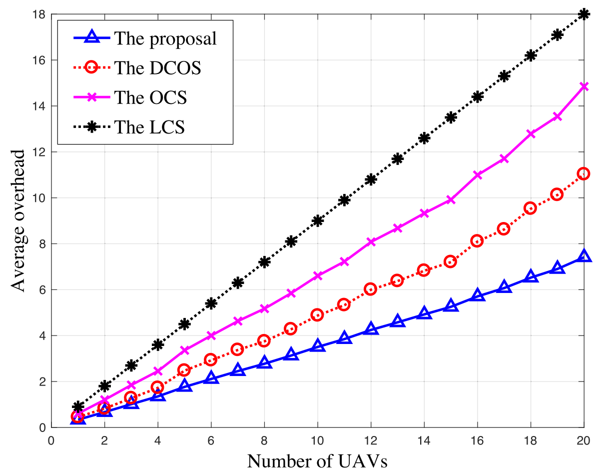


Fig. 4. The average overhead with different number of UAVs.

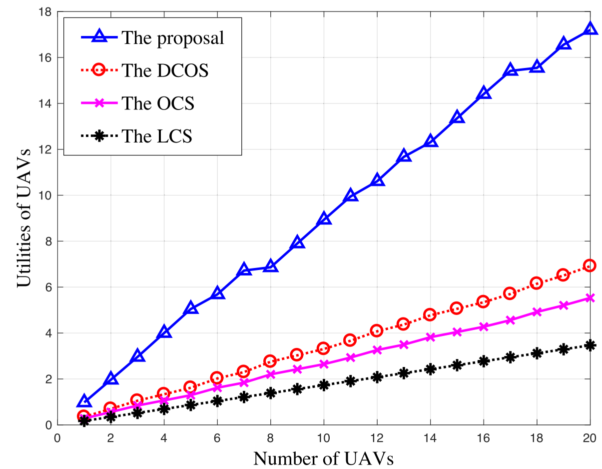


Fig. 5. The utilities of UAVs with different number of UAVs.

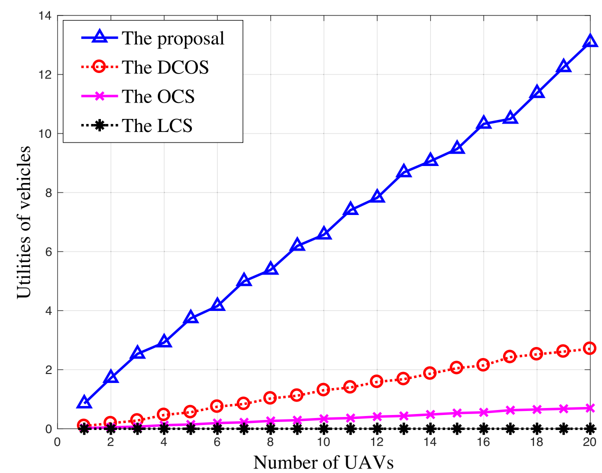


Fig. 6. The utilities of vehicles with different number of UAVs.

offloading proportion increasing from 0.15 to 0.95, while the y-axis plots the valuation of computing data. As we can see from Fig. 3, the valuation of computing data for UAV decreases with the increase of the offloading proportion. While the valuation of computing data for vehicle increases with the increase of offloading proportion. The reasons are two-fold: first, with the increase of the offloading proportion, the offloading computing delay of vehicle increases. It leads to a low discount factor for UAV. Therefore, the valuation of computing data for UAV is reduced. Second, the valuation of computing data for vehicle is proportional to the energy consumption. Therefore, as the offloading proportion increases, the high energy consumption of vehicle results in the high valuation of computing data.

Fig. 4 shows the average overhead with different number of UAVs. It can be seen that the average overhead in the four schemes increases with the increase of the number of UAVs. Moreover, the proposed scheme can achieve the lowest overhead compared with other three schemes. The reasons are as follows: when the number of UAVs increases, more computing tasks are executed by UAVs and vehicles. Therefore, the average overhead in the four schemes increases. In the proposed scheme, the matching scheme can help UAVs select the optimal vehicles to execute tasks. Moreover, the optimal offloading strategy is determined based on the bargaining game to improve the offloading efficiency and reduce the overhead. Therefore, the average overhead in the proposed scheme is low. In the DCOS, when a number of tasks are offloaded to vehicles, the overhead will be high without considering the energy consumption and computing delay of vehicles. In the OCS and LCS, the computing tasks are totally offloaded to vehicles and executed by UAVs, resulting in the high overhead.

Fig. 5 shows the utilities of UAVs with different number of UAVs. As can be seen that the utilities of UAVs increase with the increase of the number of UAVs, and the proposed scheme can obtain the highest utility for UAVs in the four schemes. This is because the matching scheme can select the optimal vehicles for UAVs to execute computing tasks. The computing delay can be reduced, and the valuation of computing data for UAVs is high. Moreover, the optimal transaction of computing data between UAVs and vehicles is determined by the bargaining game, resulting in the high utilities of UAVs. In the DCOS, when a large number of tasks are offloaded to vehicles, the efficiency of task computing is low. This results in the low utilities for UAVs. In the OCS, since the computing tasks are totally offloaded to vehicles to execute, the computing delay is high, resulting in the low utilities for UAVs. In the LCS, due to the limited computing resource of UAVs, the computing tasks executed by UAVs lead to high energy consumption and computing delay. Therefore, the utilities of UAVs are the lowest in the four schemes.

Fig. 6 shows the utilities of vehicles with different number of UAVs. From Fig. 6, we can see that the utilities of vehicles increase as the number of UAVs increases in the proposed scheme, the DCOS and the OCS. Moreover, the utilities of vehicles in the proposed scheme are higher than that in other schemes. The reasons are as follows: on one hand, more vehicles can be matched with UAVs based on the matching scheme to execute tasks. On the other hand, the transaction of computing data is based on bargaining game, which can maximize the utilities of vehicles in the proposed scheme. In the DCOS, the offloading strategy of UAVs is determined

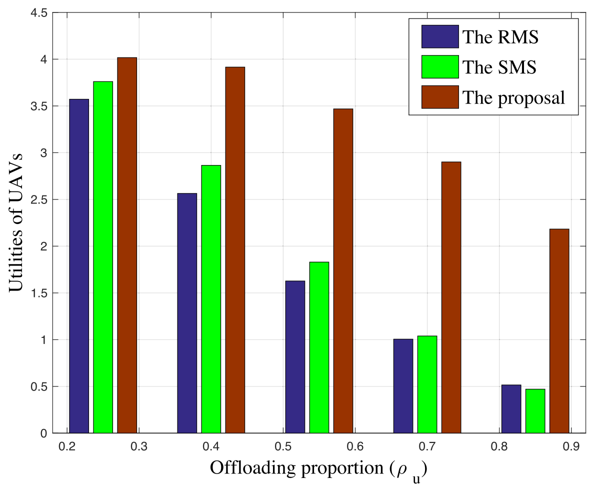


Fig. 7. The utilities of UAVs with different offloading proportions.

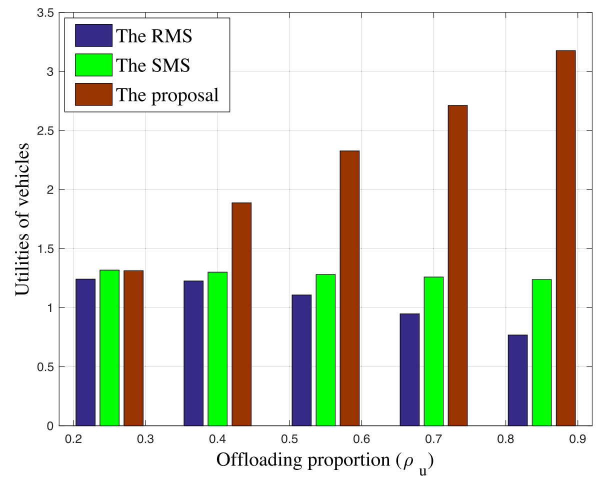


Fig. 8. The utilities of vehicles with different offloading proportions.

based on the utility function. When the utility of UAV for offloading is lower than that of other strategies, vehicles cannot execute tasks and gain more utilities, resulting in the low utilities of vehicles. In the OCS, the computing task is totally offloaded to vehicles. The energy efficiency of vehicles is low, which results in low utilities for vehicles. In the LCS, the computing tasks are totally executed by UAVs. Therefore, the utilities of vehicles are zero.

We compare the proposed matching scheme with other two schemes. In the stable matching scheme (SMS): the UAVs and vehicles are matched based on the stable computing resource of vehicle and transmission rate of UAV. In the random matching scheme (RMS): UAVs are randomly matched with vehicles without considering the computing resource and transmission rate.

Fig. 7 shows the utilities of UAVs with different offloading proportions (i.e., ρ_u). It can be seen that the utilities of UAVs decrease with the increase of the offloading proportion. Moreover, the proposed scheme can achieve the best performance in three schemes. This is because when the offloading proportion increases, more computing tasks are executed by vehicles. The valuation of computing data for vehicles becomes high, and more payments are paid to vehicles. Therefore, the utilities of UAVs are reduced. In the SMS, UAVs cannot match with the optimal vehicles, the computing efficiency of vehicles is low. Therefore, the low valuation of computing data leads to the low utilities of UAVs. In the RMS, since UAVs randomly match with vehicles, the high computing delay causes the low valuation of computing data and low utilities of UAVs.

Fig. 8 shows the utilities of vehicles with different offloading proportions (i.e., ρ_u). From Fig. 8, we can see that the utilities of vehicles increase with the increase of offloading proportion in the proposed scheme. The reason is that with the increase of offloading proportion, more computing tasks are executed by vehicles. The high valuation of computing data results in the high utilities for vehicles. In the SMS, without considering the optimal matching between UAVs and vehicles, the computing efficiency of vehicles is low. Moreover, the transaction of computing data is randomly determined. Therefore, the utilities of vehicles are low. In the RMS, vehicles randomly match with UAVs, and the transaction of computing data is random. This results in the low utilities of vehicles.

VI. Conclusion and Future Work

In this article, we have proposed a novel vehicle-assisted computing offloading scheme for UAVs in smart city to improve the offloading efficiency. First, the offloading model has been developed to determine the offloading strategy of UAVs. Second, based on the computing delay of vehicles and data transmission delay of UAVs, the preference lists have been derived to obtain the optimal matching results between UAVs and vehicles. Third, to improve the offloading efficiency, the transaction process of computing data between UAVs and vehicles has been modeled as the bargaining game. Finally, simulation results have demonstrated that the proposed scheme can jointly improve the offloading efficiency and increase the utilities of UAVs and vehicles. For the future work, we will take the security into account to improve the reliability of data offloading between UAVs and vehicles in smart city.

References

[1] C. Esposito, A. Castiglione, F. Frattini, M. Cinque, Y. Yang, and K.-K.-R. Choo, "On data sovereignty in cloud-based computation offloading for smart cities applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4521–4535, Jun. 2019.

[2] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: A TD-LTE-based V2X solution for future vehicular network," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 997–1005, Dec. 2016.

[3] L. Zhou, D. Wu, J. Chen, and Z. Dong, "Greening the smart cities: Energy-efficient massive content delivery via D2D communications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1626–1634, Apr. 2018.

[4] G. Han, H. Wang, X. Miao, L. Liu, J. Jiang, and Y. Peng, "A dynamic multipath scheme for protecting source-location privacy using multiple sinks in WSNs intended for IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5527–5538, Aug. 2020.

[5] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.

[6] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5G: Challenges, methodologies, and directions," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 78–85, Apr. 2016.

[7] M. Zhang, S. He, C. Yang, J. Chen, and J. Zhang, "VANET-assisted interference mitigation for millimeter-wave automotive radar sensors," *IEEE Netw.*, vol. 34, no. 2, pp. 238–245, Mar. 2020.

[8] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.

[9] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, Jan. 2020.

[10] L. Liu, G. Han, Y. He, and J. Jiang, "Fault-tolerant event region detection on trajectory pattern extraction for industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2072–2080, Mar. 2020.

[11] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[12] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, Jun. 2018.

[13] J. Zhang *et al.*, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[14] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.

[15] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[16] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[17] C. Zhou, Y. Gu, S. He, and Z. Shi, "A robust and efficient algorithm for coprime array adaptive beamforming," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1099–1112, Feb. 2018.

[18] G. Gui, H. Huang, Y. Song, and H. Sari, "An effective nonorthogonal multiple access scheme based on deep learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.

[19] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[20] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-efficient computation offloading for secure UAV-edge-computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6074–6087, Jun. 2019.

[21] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019.

[22] N. Chen, M. Wang, N. Zhang, and X. Shen, "Energy and information management of electric vehicular network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 967–997, 2nd Quart., 2020.

[23] Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, "A novel load balancing and low response delay framework for edge-cloud network based on SDN," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5922–5933, Jul. 2020.

[24] Y. Xu, G. Yu, X. Wu, Y. Wang, and Y. Ma, "An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1845–1856, Jul. 2017.

[25] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020.

[26] W. Zhang, G. Han, X. Wang, M. Guizani, K. Fan, and L. Shu, "A node location algorithm based on node movement prediction in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3166–3178, Mar. 2020.

[27] M. Zeng, S. Leng, Y. Zhang, and J. He, "QoE-aware power management in vehicle-to-grid networks: A matching-theoretic approach," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2468–2477, Jul. 2018.

[28] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[29] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1193–1209, Feb. 2019.

[30] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2127–2145, Apr. 2018.

[31] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7025–7037, Aug. 2019.

[32] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.

[33] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Mobile edge computing for the Internet of Vehicles: Offloading framework and job scheduling," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 28–36, Mar. 2019.

[34] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an UAV network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4964–4974, May 2019.

[35] H. Wu, Z. Wei, Y. Hou, N. Zhang, and X. Tao, "Cell-edge user offloading via flying UAV in non-uniform heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2411–2426, Apr. 2020.

[36] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[37] J. Wang, Y. Zhang, Y. Liu, and N. Wu, "Multiagent and bargaining-game-based real-time scheduling for Internet of Things-enabled flexible job shop," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2518–2531, Apr. 2019.

[38] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.

[39] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1389–1399, Jun. 2018.

[40] Z. Zheng, L. Song, D. Niyato, and Z. Han, "Resource allocation in wireless powered relay networks: A bargaining game approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6310–6323, Jul. 2017.

[41] C. Yang, J. Li, A. Anpalagan, and M. Guizani, "Joint power coordination for spectral-and-energy efficiency in heterogeneous small cell networks: A bargaining game-theoretic perspective," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1364–1376, Feb. 2016.

[42] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[43] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[44] Y. Hui, Z. Su, T. H. Luan, and J. Cai, "Content in motion: An edge computing based relay scheme for content dissemination in urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3115–3128, Aug. 2019.

[45] Q. Wu, J. Xu, and R. Zhang, "Capacity characterization of UAV-enabled two-user broadcast channel," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1955–1971, Sep. 2018.

[46] Y. Wang, X. Tao, X. Zhang, P. Zhang, and Y. T. Hou, "Cooperative task offloading in three-tier mobile computing networks: An ADMM framework," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2763–2776, Mar. 2019.

[47] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sep. 2019.

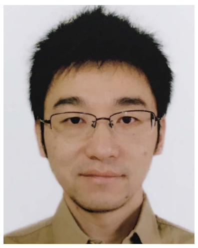
[48] D. Liu *et al.*, "Task-driven relay assignment in distributed UAV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11003–11017, Nov. 2019.

[49] X. Shao, C. Yang, D. Chen, N. Zhao, and F. R. Yu, "Dynamic IoT device clustering and energy management with hybrid NOMA systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4622–4630, Oct. 2018.

[50] H. Liao *et al.*, "Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.

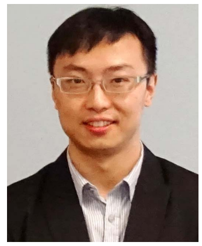
[51] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

Minghui Dai is currently pursuing the Ph.D. degree with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. His research interests include wireless network architecture and vehicular networks.



Zhou Su (Senior Member, IEEE) received the Ph.D. degree from Waseda University, Tokyo, Japan, in 2003. His research interests include multimedia communication, wireless communication, and network traffic. He is a TPC member of some flagship conferences including IEEE INFOCOM, IEEE ICC, and IEEE GLOBECOM. He received the Best Paper Award of the International Conference CHINACOM2008 and the Funai Information Technology Award for Young Researchers in 2009. He is the Chair of the Multimedia Services and Applications over Emerging Networks Interest Group (MENIG), IEEE Comsoc Society, and the Multimedia Communications Technical Committee. He also served as the Co-Chair for several international conferences including IEEE VTC Spring 2016 and IEEE CCNC2011. He is currently an Associate Editor of *IET Communications* and of *IEICE Transactions on Communications*.

Qichao Xu received the Ph.D. degree from the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China, in 2019. He is currently an Assistant Professor with Shanghai University. His research interests include wireless network architecture and vehicular networks.



Ning Zhang (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, Canada, in 2015. He was a Post-Doctoral Research Fellow with the University of Waterloo and also with the University of Toronto, Canada. He is currently an Associate Professor with the University of Windsor, Canada. He received the Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, and the *Journal of Communications and Information Networks* in 2018, IEEE ICC in 2019, the IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and IEEE ICCC in 2019, respectively. He also serves/served as a track chair for several international conferences and a co-chair for several international workshops. He serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE ACCESS, *IET Communications*, and *Vehicular Communications*. He was a Guest Editor of several international journals, such as the IEEE WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.