Joint communication and computation resource scheduling of a UAV-assisted mobile edge computing system for platooning vehicles

Yang Liu, Jianshan Zhou, Daxin Tian, Zhengguo Sheng, Xuting Duan, Guixian Gu, Victor C M Leung

Publication date
10-06-2023

Document Version
Accepted version

Citation for this work (American Psychological Association 7th edition)
Liu, Y., Zhou, J., Tian, D., Sheng, Z., Duan, X., Gu, G., & Leung, V. C. M. (2021). *Joint communication and computation resource scheduling of a UAV-assisted mobile edge computing system for platooning vehicles* (Version 1). University of Sussex. https://hdl.handle.net/10779/uos.23482205.v1

Published in
IEEE Transactions on Intelligent Transportation Systems

Link to external publisher version
https://doi.org/10.1109/TITS.2021.3082539

Joint Communication and Computation Resource Scheduling of a UAV-Assisted Mobile Edge Computing System for Platooning Vehicles

Yang Liu, Jianshan Zhou, Daxin Tian, *Senior Member, IEEE,* Zhengguo Sheng, *Senior Member, IEEE,*
Xuting Duan, Guixian Qu, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—Connected and autonomous vehicles (CAVs) are recently envisioned to provide a tremendous social impact, while they put forward a much higher requirement for both vehicular communication and computation capacities to process resource-intensive applications. In this paper, we study unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) for a platoon of wireless power transmission (WPT)-enabled vehicles. Our objective is to maximize the system-wide computation capacity under both communication and computation resource constraints. We incorporate the coupled effects of the platooning vehicles and the flying UAV, air-to-ground (A2G) and ground-to-air (G2A) communications, onboard computing and energy harvesting into a joint scheduling optimization model of communication and computation resources. To tackle the resulting optimization problem, we propose a successive convex programming method based on a second-order convex approximation, in which feasible search directions are obtained by solving a sequence of quadratic programming subproblems and used to generate feasible points that can approach a local optimum. We also theoretically prove the feasibility and convergence of the proposed method. Moreover, simulation results are provided to validate the effectiveness of our proposed method and demonstrate its superior performance over other conventional schemes.**

***Index Terms*—Aerial-ground cooperative vehicular network, mobile edge computing, wireless power transmission, unmanned aerial vehicle, convex optimization.**

I. Introduction

A. Background

With the integration of information communication technologies (ICT) with vehicles, connected and autonomous vehicles (CAVs) have emerged as an important revolution for our mobility society. They are envisioned to bring huge social, industrial and economic benefits to our life, such as promoting traffic safety and efficiency, reducing environmental pollution, and saving traffic energy consumption, etc [1], [2]. Hence, many research efforts have been dedicated to developing effective solutions to address the cooperative control-related challenge of CAV platoons, and various control paradigms have been proposed in current literature [3]–[10]. However, the cooperative control or other control-related upper-layer applications of platooning vehicles usually rely on the high-performance communication and computation of vehicular terminals. In particular, the envisioned autonomous driving systems have to rely on powerful computing capacity for realizing data-massive and latency-critical environmental perception and artificial intelligence (AI)-based driving decision-making [11], [12]. That is, advanced communication and computation technologies will be critical for enabling the connectivity and autonomy of platooning vehicles [13].

In addition, with the fast-growing demands on pervasive communication and computing, terrestrial vehicular networks alone cannot meet the needs effectively and efficiently. In the vision of Beyond-5G or 6G era, terrestrial vehicular networks, aerial or even spacial ICT infrastructures are expected to be integrated to provision more ubiquitous wireless connectivity and computing services, anywhere and anytime. To facilitate B5G/6G application scenarios, a cloudlet or other mobile edge computing (MEC) facilities can be mounted on unmanned aerial vehicles (UAVs) to offer edge computing services from the sky [14]. The UAVs equipped with cloudlets are further envisioned to cooperate with ground vehicles or other Internet-of-Things (IoT) devices to spawn aerial-ground cooperative networks [15], [16]. To support the future evolution of Intelligent Transportation Systems (ITS), UAVs and CAVs can cooperate on communication and computation management, and then form aerial-ground cooperative vehicular networks to flexibly adapt to highly dynamic topologies and diverse Quality-of-Service (QoS) requirements for vehicular computation-intensive and task-critical applications.

Although aerial-ground cooperative vehicular networks are viewed as a promising and appealing paradigm for a variety of connected intelligence and automation applications, they inevitably encounter fundamental challenges arising from the dy-

This research was supported in part by the China Postdoctoral Science Foundation under Grant No. 2020M680299, the National Postdoctoral Program for Innovative Talents, the "Zhuoyue" Program of Beihang University (Postdoctoral Fellowship) under Grant No. 262716, the National Natural Science Foundation of China under Grant No. 61822101 and U20A20155, the Beijing Municipal Natural Science Foundation No. L191001, the Newton Advanced Fellowship under Grant No. 62061130221. (Yang Liu and Jianshan Zhou contributed equally to this work.) (Corresponding author: Jianshan Zhou.)

Yang Liu, Jianshan Zhou, Daxin Tian, Xuting Duan, and Guixian Qu are with Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems & Safety Control, School of Transportation Science and Engineering, Beihang University, Beijing 100191, China (e-mail: xdfliuyang1@126.com, jianshanzhou@foxmail.com, dtian@buaa.edu.cn, duanxuting@buaa.edu.cn, guixianqu@foxmail.com).

Zhengguo Sheng is with Department of Engineering and Design, the University of Sussex, Richmond 3A09, UK (e-mail: z.sheng@sussex.ac.uk).

Victor C. M. Leung is with Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, B.C., V6T 1Z4 Canada (e-mail: vleung@ieee.org).

namic nature, constrained resources and underlying heterogeneous computing devices of the system [17], [18]. In particular, a cloudlet-mounted UAV, working as an aerial MEC server to collect and process computation-intensive and latency-critical tasks offloaded from ground platooning vehicles, is limited due to its onboard energy constraint. To tackle the challenge, wireless power transmission (WPT) solutions, with which a transmitter converts electricity into electromagnetic field such that a receiver can harvest energy from the field, can be joined with UAVs to enhance their service duration [19], [20]. Nevertheless, another great challenge has to be properly addressed for the practical realization of the UAV-assisted WPT-enabled MEC system that the WPT process is highly coupled with the energy consumption in both communication and computation. Hence, it is of great significance to jointly schedule communication and computation resources to offer a key enabling foundation for such a cyber-physical system.

B. Prior Works

Currently, many researchers are attracted by the potentials of cooperative control of connected platooning vehicles. They are engaged in developing innovative design paradigms to address the platoon control-related challenge from the perspective of control theory, such as H-infinity synthesis-based robust control [3], [4], car-following model and consensus-based control [5]–[7], vehicular sensor-based and Doppler-compensation-based control [10], and model predictive control (MPC)-based methods [8], [9]. Nevertheless, among these platoon control-related works, few have targeted the joint optimization of communication and computation resources. They do not aim to address the optimization problem of air-ground cooperative resource scheduling in the context of UAV-assisted MEC-enabled vehicular application scenarios. Additionally, some researchers focus on joining UAVs and ground mobile users into a MEC system. They are engaged in developing innovative system solutions for addressing either the networking or the computing challenges of envisioned aerial-ground cooperative networks [21]–[23]. Different design models and optimization goals have been achieved in current literature. For example, Li *et al.* have investigated the application scenarios, where blockchain and MEC technologies are integrated into the UAV-assisted communication system [24]. In [25], MEC-enabled UAV networks are also studied, in which blockchain is introduced to ensure communication security and reliability meanwhile a neural learning-based method is proposed to optimize the caching performance of the network. Although these works above can provide effective system optimization solutions for UAV-assisted MEC networks in terms of computation capacity enhancement, the communication and computation resource management of ground end mobile users have not been jointly considered, while their coupled constraints and dynamics inherently affect the system performance.

In addition to the works [21]–[25], many other studies have also developed their resource optimization frameworks based on the deep reinforcement learning methodology such as the works [26]–[31]. Although deep reinforcement learning (deep RL) has shown success in solving specified resource scheduling problems in the context of UAV-assisted networks to some extent, such deep RL-based solutions may encounter difficulty and impracticality in real system deployment since they rely on the basic assumption that the system dynamics follows a Markovian chain, and they usually require a lot of training data and computation for learning the state space of a real physical system. That is, deep RL-based approaches may be sample-inefficient [32], which would preclude their implementation in real aerial-ground cooperative networks. Differently, in [33], Li *et al.* focus on the energy-efficiency maximization for a UAV-assisted MEC system and propose an iterative algorithm based on successive convex approximation (SCA) technique to jointly optimize the UAV trajectory, the user transmit power and computation load allocation. In [34], Jeong *et al.* have also applied SCA technique to jointly optimize the computation task allocation and the flight path of a cloudlet-mounted UAV. In [35], Zhang *et al.* also aim at the optimization of the system energy consumption and they have used a Lyapunov optimization-based method to maximize the average weighted energy consumption under stochastic task queueing situations. Besides, Hu *et al.* have proposed a dual decomposition-based iterative algorithm for optimizing the sum of the maximum delay among all the ground users in [36]. The key design factors include the UAV trajectory, the ratio of offloading tasks and the users' scheduling decisions [36]. In these representative works [33]–[36], the UAV motion has a great effect on the system optimization. Thus, the researchers combine the UAV's trajectory optimization with their resource scheduling solutions. However, in their models, the ground end users are implicitly assumed to be static or with low mobility. The application scenarios do not accommodate fast moving connected vehicles or highly-dynamic vehicle platoons.

Besides, wireless power transmission, also called wireless power transfer, has been used to enhance the service performance of a MEC system in energy-hungry IoT scenarios [37]–[39]. In [37], wearable computing devices are enabled to harvest energy from a base station in order to process their application data locally or offload the data to a cloud. In [38], downlink WPT from an access point (AP) of a cloud to IoT sensors is considered in a multi-user MEC system. Similarly, [39] also considers the downlink WPT from an AP to ground devices and proposes a bi-section search algorithm to schedule different time slots for offloading, computing and energy harvesting. Furthermore, such a technique is also integrated with UAV-assisted MEC systems [40]–[42]. In [40], Zhou *et al.* have investigated a WPT-enabled UAV-assisted MEC system with the goal of maximizing the overall computation rate. They characterize the dynamics of the WPT from a UAV to ground smart devices and present a two-stage and a three-stage algorithms for computation mode selection. In [41], a UAV-assisted sensor network is considered, where energy-hungry sensor devices are enabled to harvest power from a cloudlet-mounted UAV, and a SCA-based algorithm has been proposed to jointly optimize CPU frequencies, offloading data amount, transmission power, and UAV's trajectory. In [42], Du *et al.* have proposed an energy consumption minimization model that jointly optimizes the IoT device association and their service order, computation resource allocation, WPT duration, and UAV's hovering duration. In the scenario, the UAV needs to first power the IoT devices so as to enable them to offload collected data. As can be seen, in most of the current works [37]–[42], downlink WPT is adopted to prolong the active duration of ground IoT devices like sensors or other smart terminals. The key assumption in these above works is that a cloudlet-mounted UAV plays the role of a power supplier. Nevertheless, the UAV is an energy-restricted device itself and its flight duration may be significantly reduced when the UAV supplies energy to other terminals via downlink WPT.

C. Motivation and Contributions

From the related works above and the references therein, limited research efforts have been made and few works have been presented on joint communication and computation resource optimization for WPT-enabled platooning vehicles as mobile end users in the UAV-assisted MEC system. Toward this end, we investigate the UAV-assisted and WPT-enabled vehicular MEC system in this paper. Specifically, we focus on the computation capacity maximization of the system, in which a platoon of ground moving vehicles and a flying cloudlet-mounted UAV co-share communication and computation resources. Due to the resource constraints, the UAV is enabled to harvest energy from the ground platooning vehicles acting as a mobile array of power suppliers, while the platooning vehicles would like to gain computing services from the sky.

In this paper, we consider to incorporate the coupled dynamics of the aerial and ground nodes' mobility and the air-to-ground (A2G)/ground-to-air (G2A) communications in our joint optimization modeling. Under the communication and computation resource constraints, the transmission power of both the vehicles and the UAV and the duration of time slots allocated for different processes including task offloading over G2A links, task computing, and resulting data downloading over A2G links are jointly optimized. The system model essentially differentiates our work from existing works. The main contributions of this paper are summarized as follows.

1) We propose a system optimization model for joint communication and computation scheduling of a platoon of ground vehicles and a cloudlet-mounted UAV. We explicitly model the car-following dynamics of ground platooning vehicles and incorporate its effect on computation task offloading and result downloading into the model. Besides, we jointly take into account the effects of the A2G/G2A communications, the onboard computation, and the platoon-to-UAV WPT for resource scheduling so as to enhance the QoS of the UAV.

2) We exploit the second-order Taylor expansion and the positive-definite Hessian approximation techniques to formulate the optimization objective function and linearize the constraints at each iteration in a feasible region. Thus, we propose a successive convex approximation-based iterative method by transforming the original problem into a sequence of convex quadratic programming subproblems so as to cope with the resulting complexity of the joint optimization problem.

3) We theoretically prove that the feasibility and convergence of the proposed method can be well guaranteed. Moreover, we also conduct extensive simulations to validate the effectiveness of the proposed method under different conditions, such as different numbers of platooning vehicles, different supplied wireless power and mobility. Numerical results show the superior performance of our method over conventional schemes in terms of computation capacity improvement.

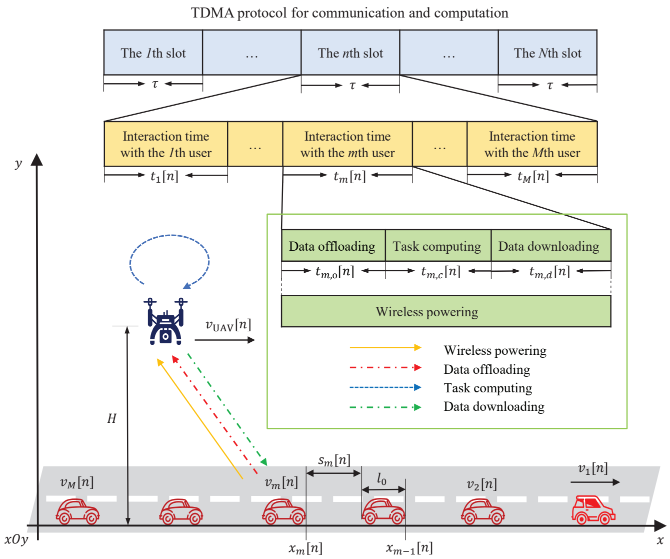


Fig. 1. A typical UAV-assisted MEC system with the cooperation of a cloudlet-mounted UAV and a platoon of ground moving vehicles as end users.

D. Organization

The rest of the paper is organized as follows. Section II develops the system model. In Section III, we formulate the joint optimization problem, and propose an algorithm based on the successive convex approximation theory in Section IV. Section V evaluates the performance. Finally, our paper is concluded and future work is outlined in Section VI.

II. System Model

A. Network Model

A UAV-assisted wireless powered MEC system for a platoon of ground moving vehicles is considered in Fig. 1, where a cloudlet is mounted on the UAV to provide MEC services to the ground platoon vehicles meanwhile the vehicles equipped with a radio frequency (RF) transmitter can supply energy and offload computation tasks to the UAV for remote processing. At this point, the vehicle platoon can form a mobile wireless power-supply array. In addition to computation offloading via G2A communication, we also consider that the UAV can feed back the results to an associated vehicle after it completes task computing via A2G communication. According to current literature [37]–[42], we consider that the UAV is flying at a constant altitude in the same direction to the vehicle platoon, which is reasonable since a fixed flight altitude can avoid unnecessary energy cost resulting from frequent descending and ascending operations. The flight height of the UAV is denoted by H, and the steady cruising velocity is v_{UAV}. Besides, the finite scheduling time horizon is discretized into N time slots each with the duration of τ seconds. The set of time slot indexes can be represented by $\mathcal{N} = \{1, \ldots, N\}$.

We also let the number of platooning vehicles be M and the vehicle set be $\mathcal{M} = \{1, \ldots, M\}$.

Similar to [39]–[41], the well-known Time Division Multiple Access (TDMA) protocol can be properly adopted to eliminate multi-user wireless interferences at the physical layer during vehicular computation offloading. To be specific, as shown in Fig. 1, a time slot τ is further divided into M time slots with a smaller duration, each of which is allocated for the interaction between a single vehicle and the UAV. The occupancy duration of the interaction session between vehicle $m \in \mathcal{M}$ and the UAV during the n-th time slot, $n \in \mathcal{N}$, is denoted by $t_m[n] \in [0, \tau]$. Accordingly, we divide such a session duration of a vehicle into three stages that are processed one by one, i.e., the computation offloading stage of the vehicle, the task computing stage of the UAV, and the resulting data downloading stage. The allocated durations of these processing stages are denoted by $t_{m,o}[n]$, $t_{m,c}[n]$ and $t_{m,d}[n]$, respectively. Hence, using the TDMA protocol, we can formulate the following equality constraints for time resource allocation among these platooning vehicles.

$$\begin{cases} t_{m,o}[n] + t_{m,c}[n] + t_{m,d}[n] = t_m[n], m \in \mathcal{M}; \\ \sum_{m=1}^{M} t_m[n] = \tau, \ n \in \mathcal{N}. \end{cases} \tag{1}$$

From the platoon perspective, we only consider the longitudinal kinematics of the moving vehicles and the UAV and thus denote the longitudinal position of a vehicle $m \in \mathcal{M}$ at time slot $n \in \mathcal{N}$ by $x_m[n]$ and that of the UAV by $x_{\text{UAV}}[n]$. At this point, the relative distance between vehicle n and the UAV can be simplified to

$$d_m[n] = \sqrt{\left(x_m[n] - x_{\text{UAV}}[n]\right)^2 + H^2}, m \in \mathcal{M}, n \in \mathcal{N}. \tag{2}$$

B. Platooning Dynamics

Current literature, like [34], [35], [40], focuses on providing MEC services for fixed ground users. However, such system scenarios may not be practical for connected vehicles as end users. Here, we take into consideration the mobility of ground moving vehicles, especially the platooning dynamics, from the perspective of traffic flow modeling. Generally, in real road traffic flows, the kinematics of a vehicle inherently involves a dependence on that of the preceding vehicles moving in the same direction. That is, the driver usually makes driving decisions in response to variations in the driving behaviors of the preceding vehicle, which is so called car-following dynamics. Therefore, to characterize such platooning dynamics, we exploit the Intelligent Driving Model (IDM), which is originally proposed by Treiber *et al.* and has been validated to well capture the micro car-following dynamics in freeway or urban traffic flows [43], [44]. It is noted that the IDM has also been widely adopted in the field of traffic flow modeling and control in both academia and industry [45].

Let the longitudinal velocity and acceleration of vehicle $m \in \mathcal{M}$ be v_m and a_m, respectively. The relative velocity and the inter-vehicle distance between the vehicle and its preceding one is $\Delta v_m = v_m - v_{m-1}$ and $s_m = x_{m-1} - x_m - l_0$, respectively, where l_0 denotes the vehicle length. Since the motion equation of vehicle m relies on the kinematics of vehicle $m-1$, its acceleration a_m can be further expressed by the following unified form

$$a_m = F_{\text{IDM}}(v_m, \Delta v_m, s_m), \tag{3}$$

where $F_{\text{IDM}}(\cdot)$ is used to describe the car-following dynamics in the vehicle platoon. Based on the IDM, we can further formulate $F_{\text{IDM}}(\cdot)$ in a nonlinear form as follows

$$a_m = a_{\max}\left[1 - \left(\frac{v_m}{v_{\max}}\right)^{\delta} - \left(\frac{s^*(v_m, \Delta v_m)}{s_m}\right)^2\right] \tag{4}$$

for all $m \in \mathcal{M}$, where $a_{\max}$ represents the allowed maximum acceleration of each vehicle, and $v_{\max}$ is the desired velocity in a steady traffic flow. The acceleration exponent δ is used to characterize the sensitivity of the driver, which is usually set from 1 to 5 [44]. The desired space s^* is defined as

$$s^*(v_m, \Delta v_m) = s_{\min} + t_r v_m + \frac{v_m \Delta v_m}{2\sqrt{a_{\max} b_{\max}}}, m \in \mathcal{M}, \tag{5}$$

where the minimum net distance $s_{\min}$ indicates the desired safe space, the minimum reaction time t_r represents the desired time headway, and the maximum deceleration $b_{\max}$ ($b_{\max} > 0$) shows the comfortable braking deceleration.

Now, based on (3), we can model the time-discrete longitudinal velocity and position of each vehicle in a platoon by a double integrator model

$$\begin{cases} v_m[n+1] = v_m[n] + a_m[n]\tau; \\ x_m[n+1] = x_m[n] + v_m[n]\tau + \dfrac{1}{2}a_m[n]\tau^2 \end{cases} \tag{6}$$

for all $m \in \mathcal{M}$ and all $n \in \mathcal{N}$. It is observed from (3) and (6) that the mobility of each vehicle in the platoon is coupled in a car-following manner, which can significantly affect the space distance of G2A/A2G links and further the network communication performance.

C. Communication Model

According to [33], [39], [40], it can be reasonably assumed that the wireless communication channel between the UAV and each ground vehicle is mainly dominated by Line of sight (LoS) when the system is deployed in a widely-open working area. Hence, the channel gain between vehicle m and the UAV at time slot n can be given as

$$g_m[n] = g_0 d_m[n]^{-\kappa}, m \in \mathcal{M}, n \in \mathcal{N}, \tag{7}$$

where g_0 is the channel power gain at a reference distance $d_0 = 1\text{m}$, and κ is the path loss exponent that depends on the dominated loss path. Given the LoS signal propagation, κ is usually set to 2 in current literature.

Let the available bandwidth be B, and the transmission power of vehicle m for offloading computation tasks to the UAV and that of the UAV for transmitting resulting data to vehicle m at time slot n be $P_{m,o}[n]$ and $P_{m,d}[n]$, respectively. Based on the channel gain formulation (7), the transmission data rates for the uplink and the downlink at time slot n are given based on the channel capacity theory as follows

$$\begin{cases} R_{m,o}[n] = B\log_2\left(1+\dfrac{P_{m,o}[n]g_m[n]}{N_0}\right); \\ R_{m,d}[n] = B\log_2\left(1+\dfrac{P_{m,d}[n]g_m[n]}{N_0}\right) \end{cases} \tag{8}$$

for $m \in \mathcal{M}, n \in \mathcal{N}$, where N_0 is the average noise power.

Based on (8), the energy consumption of the UAV during vehicle m offloading computation tasks at time slot n, $E_{m,o}[n]$, and its energy consumption during feeding back resulting data, $E_{m,d}[n]$, are formulated as follows, respectively,

$$\begin{cases} E_{m,o}[n] = P_{m,o}[n]g_m[n]t_{m,o}[n]; \\ E_{m,d}[n] = P_{m,d}[n]t_{m,d}[n], m \in \mathcal{M}, n \in \mathcal{N}. \end{cases} \tag{9}$$

Note that in the task offloading stage, the receiving power consumption of the UAV needs to take into account the channel fading. Thus, the power $P_{m,o}[n]$ is multiplied with the channel gain $g_m[n]$ to evaluate the power consumption of the UAV when vehicle m performs task offloading.

D. Computation Model

After the vehicles offload raw computation data to the UAV, the UAV computes the tasks with the CPU frequency f. Denote the number of CPU cycles for computing one bit of raw data by C. Similar to [40], [41], the local computation rate of the UAV, $R_{\mathrm{UAV},c}$, can be formulated by

$$R_{\mathrm{UAV},c} = \frac{f}{C}. \tag{10}$$

Note that the amount of the task data offloaded from the m-th platooning vehicle should be equal to the data amount to be computed by the UAV at time slot n when considering reliability-guaranteed computation offloading. We can propose the following equality constraint equation to establish the relationship between the duration of the task offloading stage, $t_{m,o}[n]$, and that of the task computing stage, $t_{m,c}[n]$,

$$R_{m,o}[n]t_{m,o}[n] = R_{\mathrm{UAV},c}t_{m,c}[n], m \in \mathcal{M}, n \in \mathcal{N}. \tag{11}$$

Besides, the energy consumption of the UAV for local execution also depends on the CPU frequency of its computing processor f. Similar to [39], [40], [46], we formulate the energy consumption of the UAV for processing the tasks offloaded from vehicle m at time slot n as follows

$$E_{m,c}[n] = \lambda_c f^3 t_{m,c}[n], m \in \mathcal{M}, n \in \mathcal{N}, \tag{12}$$

where λ_c is an effective capacitance coefficient dependent on the UAV's processor chip architecture.

E. Energy Harvesting Model

In our considered scenario, the platooning vehicles can power the UAV at each time slot. To characterize the WPT dynamics of the platooning vehicles, a linear energy harvesting model can be employed according to current literature [39]–[41]. Specifically, let η_0 $(0 < \eta_0 < 1)$ be the energy conservation efficiency, and P_{sup} be the energy supply power of each vehicle. The UAV's harvested energy from the m-th vehicle at the n-th slot is expressed as

$$E_{m,p}[n] = \eta_0 P_{\mathrm{sup}} g_m[n]\tau, m \in \mathcal{M}, n \in \mathcal{N}. \tag{13}$$

With WPT, the UAV is able to exploit its harvested energy from the platooning vehicles to perform communication and computation-related operations. At this point, we can further propose the inequality on the energy consumption associated with the task offloading, computing and data downloading and the harvested energy as follows

$$\sum_{m=1}^{M}\left(E_{m,o}[n] + E_{m,c}[n] + E_{m,d}[n]\right) \le \sum_{m=1}^{M} E_{m,p}[n] + \epsilon, \tag{14}$$

for all $n \in \mathcal{N}$, where ϵ denotes the energy carried by the UAV that is reserved for additional circuit overheads.

III. Problem Formulation

In this work, we aim to maximize the overall computation rate of the aerial-ground cooperative network subject to the energy and time allocation constraints. The transmission power of the platooning vehicles for computation offloading via G2A communication, $P_{m,o}[n]$, that of the UAV for feeding back resulting data via A2G communication, $P_{m,d}[n]$, and the time durations allocated for the three processing stages, i.e., task offloading, task computing and data downloading, $t_{m,o}[n], t_{m,c}[n], t_{m,d}[n]$, should be jointly optimized to achieve the system goal. For simplicity, we define different column vectors to lump the communication and computation-related decisions by $\mathbf{p}_o = \mathrm{col}\{P_{m,o}[n], m \in \mathcal{M}, n \in \mathcal{N}\}$, $\mathbf{p}_d = \mathrm{col}\{P_{m,d}[n], m \in \mathcal{M}, n \in \mathcal{N}\}$, $\mathbf{t}_o = \mathrm{col}\{t_{m,o}[n], m \in \mathcal{M}, n \in \mathcal{N}\}$, and $\mathbf{t}_d = \mathrm{col}\{t_{m,d}[n], m \in \mathcal{M}, n \in \mathcal{N}\}$. Let the global decision variable be a column vector $\mathbf{X}$, i.e., $\mathbf{X} = \mathrm{col}\{\mathbf{p}_o, \mathbf{p}_d, \mathbf{t}_o, \mathbf{t}_d\}$. Based on (8), the total computation rate of the system can be formulated by combining the offloading rates from the platooning vehicles and the downloading rate from the UAV as follows, which reflects the system computation capacity and is treated as the optimization goal,

$$\theta(\mathbf{X}) = \sum_{n=1}^{N}\sum_{m=1}^{M}\left[\begin{array}{l} B\log_2\left(1+\dfrac{P_{m,o}[n]g_m[n]}{N_0}\right)t_{m,o}[n] \\ +B\log_2\left(1+\dfrac{P_{m,d}[n]g_m[n]}{N_0}\right)t_{m,d}[n] \end{array}\right]. \tag{15}$$

It is noted that since the time duration allocated for the local task computing $t_{m,c}[n]$ can be expressed as a function of $t_{m,o}[n]$ as in (11), we only need to consider $t_{m,o}[n]$ and $t_{m,d}[n]$ as the decision variables related to time scheduling.

Combining the mobility, physical-layer communication, WPT, and computation related models from (1) to (15) above, the computation rate maximization by jointly scheduling communication and computation resources is formulated as $\mathbf{P}_1$

$$
\begin{aligned}
\mathbf{P}_1 : \max_{\mathbf{X}} :\ & \theta(\mathbf{X}) && \text{(16a)}\\
\text{s.t. }& (1), (11), (14) && \text{(16b)}\\
& P_{m,d}[n] \leqslant P_{m,o}[n] && \text{(16c)}\\
& 0 \leqslant P_{m,d}[n] t_{m,d}[n] \leqslant P_{m,o}[n] t_{m,o}[n] && \text{(16d)}\\
& 0 \leqslant t_{m,o}[n],\ t_{m,d}[n] \leqslant \tau && \text{(16e)}\\
& 0 \leqslant P_{m,o}[n],\ P_{m,d}[n] \leqslant P_{\max} && \text{(16f)}\\
& m = 1, ..., M; \quad n = 1, ..., N && \text{(16g)}
\end{aligned}
$$

where $P_{\max}$ denotes the maximum transmission power. It is remarked that in general application scenarios, the amount of the resulting data is much smaller than that of the raw task data. Hence, we impose the inequality (16c) on the uplink and the downlink transmission power levels such that the transmission data rate of the offloading link should be higher than that of the downloading link. Limited to the finite battery capacity, the transmission energy consumption of the UAV is lower than that of vehicles but always positive, which is expressed by the constraint condition (16d). (16e) and (16f) correspond to the boundary constraints on the allocated time duration and the transmission power, respectively.

From the model $\mathbf{P}_1$, it is seen that the joint optimization model is a highly nonlinear and non-convex problem. In addition, the A2G/G2A wireless channel gains heavily depend on the time-varying spatial relative distance between each platooning vehicle and the UAV as presented by (2) and (7). Thus, the dynamics of the platooning vehicles and that of the flying UAV are coupled and significantly affect the end-to-end communication performance, which is incorporated into the goal formulation of the model $\mathbf{P}_1$, i.e., $\theta(\mathbf{X})$ in (15). The coupled complexity can make directly solving $\mathbf{P}_1$ challenging. It is quite costly or even impossible to search a global optimum of such a strongly non-convex problem. Therefore, in the following section, we would like to propose an iterative method to converge to a local optima efficiently.

IV. Proposed Optimization Method

Motivated by the theory of successive convex approximation, we first propose a convex programming subproblem at each iterate point to approximate the original model, and obtain a feasible direction for searching the locally optimal solution within the neighbor of each iterate point. For simplicity, we rearrange the equality and the inequality constraints of $\mathbf{P}_1$ by $C_i(\mathbf{X}) = 0$ and $C_j(\mathbf{X}) \geq 0$ where i is used to denote the index of an equality constraint while j denotes the index of an inequality constraint. Specifically, we denote the equality constraints on the time resource allocation in (1) by

$$
C_i(\mathbf{X}) = \sum_{m=1}^{M} \left(t_{m,o}[n] + t_{m,c}[n] + t_{m,d}[n] \right) - \tau \tag{17}
$$

for $i = 1, \ldots, N$, and the equality constraints in (11) by

$$
C_i(\mathbf{X}) = R_{m,o}[n] t_{m,o}[n] - R_{\text{UAV},c}[n] t_{m,c}[n] \tag{18}
$$

for $i = N + 1, \ldots, N + N \times M$. The set of indexes associated with the equality constraints can be denoted by $\mathcal{I} = \{1, \ldots, N + N \times M\}$. Similarly, we let the set of indexes associated with the inequality constraints be $\mathcal{J}$. Note that the total number of the inequalities is $N + 11NM$. Thus, the cardinal number of $\mathcal{J}$ is $|\mathcal{J}| = N + 11NM$. We rearrange the inequality constraints on the energy consumption in (14) as

$$
C_j(\mathbf{X}) = \epsilon + \sum_{m=1}^{M} \left(E_{m,p}[n] - E_{m,o}[n] - E_{m,c}[n] - E_{m,d}[n] \right) \tag{19}
$$

for $j = 1, \ldots, N$, and denote other inequality constraints from (16c) to (16f) by $C_j(\mathbf{X})$ with the index j ranging from $N+1$ to $N + 11NM$. The expressions of $C_j(\mathbf{X})$ are detailed in Appendix A. Now, with the above notations, we can rearrange $\mathbf{P}_1$ into a compact form $\mathbf{P}_2$ as follows

$$
\begin{aligned}
\mathbf{P}_2 :\ \min_{\mathbf{X}} :\ & f(\mathbf{X}) = -\theta(\mathbf{X})\\
\text{s.t.} & \begin{cases} C_i(\mathbf{X}) = 0, i \in \mathcal{I}; \\ C_j(\mathbf{X}) \geq 0, j \in \mathcal{J}. \end{cases}
\end{aligned} \tag{20}
$$

For $\mathbf{P}_2$, let $\mathbf{X}_k$ be a feasible point obtained at the previous iteration $k-1$. We can generate a new iterate $\mathbf{X}_{k+1}$ at k by

$$
\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha_k \mathbf{r}_k \tag{21}
$$

where $\mathbf{r}_k$ represents the feasible search direction within a certain neighborhood of $\mathbf{X}_k$, and α_k denotes a proper step size in the search direction $\mathbf{r}_k$ at iteration k. Additionally, for $k = 1$, $\mathbf{X}_1$ is an initial feasible point that can be specified according to a uniform resource allocation scheme. To realize such an iterative paradigm (21), it is the key point to generate a sequence of feasible search directions $\{\mathbf{r}_k, k = 1, \ldots\}$ and a series of proper step size $\{\alpha_k, k = 1, \ldots\}$. To achieve this goal, we resort to the theory of successive convex approximation. In the following subsection, we would like to detail the algorithm design based on successive convex approximation.

A. Convex Approximation Transformation

Specifically, let the Lagrangian function of $\mathbf{P}_2$ be

$$
L(\mathbf{X}, \boldsymbol{\lambda}) = f(\mathbf{X}) - \sum_{i \in \mathcal{I}} \lambda_i C_i(\mathbf{X}) - \sum_{j \in \mathcal{J}} \lambda_j C_j(\mathbf{X}), \tag{22}
$$

where $\boldsymbol{\lambda} = \text{col}\{\lambda_l, l \in \mathcal{I} \cup \mathcal{J}\}$ is the column vector collecting the Lagrange multipliers. Based on (22), given the feasible point $\mathbf{X}_k$ and the Lagrange multipliers $\boldsymbol{\lambda}_k$ at k, we define the positive definite quasi-Newton approximation of the Hessian matrix of the Lagrangian function $L(\mathbf{X}_k, \boldsymbol{\lambda}_k)$, $\nabla^2_{\mathbf{X}\mathbf{X}} L(\mathbf{X}_k, \boldsymbol{\lambda}_k)$, by $\mathbf{Q}_k$. Now, based on the second-order Taylor expansion, we can establish a convex quadratic function as a local approximation of $f(\mathbf{X})$ at k, $f(\mathbf{X}) \approx g(\mathbf{X}) = f(\mathbf{X}_k) + \nabla_{\mathbf{X}} f(\mathbf{X}_k)^{\mathrm{T}} (\mathbf{X} - \mathbf{X}_k) + 0.5(\mathbf{X} - \mathbf{X}_k)^{\mathrm{T}} \mathbf{Q}_k (\mathbf{X} - \mathbf{X}_k)$ where $\mathbf{X}$ is within a neighborhood of $\mathbf{X}_k$. Similarly, we can also use the first-order Taylor expansion to linearize the nonlinear constraints. Namely, at $\mathbf{X}_k$, we have $C_l(\mathbf{X}) \approx C_l(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}} (\mathbf{X} - \mathbf{X}_k)$ for $l \in \mathcal{I} \cup \mathcal{J}$. By letting $\mathbf{r}$ be $\mathbf{r} = \mathbf{X}_{k+1} - \mathbf{X}_k$, we can develop a convex quadratic programming subproblem $\mathbf{P}_3$ to obtain a search direction for $\mathbf{X}_{k+1}$ as follows

$$\mathbf{P}_3: \min_{\mathbf{r}} :\nabla_{\mathbf{X}} f(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} + 0.5\mathbf{r}^{\mathrm{T}}\mathbf{Q}_k\mathbf{r}$$
$$\text{s.t.} \begin{cases} C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} = 0, i \in \mathcal{I}; \\ C_j(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_j(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j \in \mathcal{J}. \end{cases} \quad (23)$$

Besides, $\mathbf{P}_3$ may be infeasible due to the fact that its constraints are approximative to those of $\mathbf{P}_2$. To deal with such a potential issue, we further introduce an auxiliary variable η to construct new feasible constraints. That is, we can present a linear programming as

$$\mathbf{P}_4: \max_{\eta,\mathbf{r}} :\eta$$
$$\text{s.t.} \begin{cases} \eta C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} = 0, i \in \mathcal{I}; \\ \eta C_{j_1}(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_{j_1}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j_1 \in \mathcal{J}_1; \\ C_{j_2}(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_{j_2}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j_2 \in \mathcal{J}_2; \\ \eta \in [0,1], \end{cases} \quad (24)$$

where $\mathcal{J}_1 = \{j \in \mathcal{J} : C_j(\mathbf{X}_k) < 0\}$ and $\mathcal{J}_2 = \{j \in \mathcal{J} : C_j(\mathbf{X}_k) \geq 0\}$. It can be observed from $\mathbf{P}_4$ that if $\eta = 0$, $\mathbf{r} = \mathbf{0}$ is always a feasible solution of $\mathbf{P}_4$. Therefore, the existence of an optimal solution is always guaranteed by $\mathbf{P}_4$ since its feasible region is not empty. We can denote an optimal value of $\mathbf{P}_4$ by $\overline{\eta}$ and have $\overline{\eta} \in [0,1]$. Accordingly, we can further obtain the following results on $\overline{\eta}$:

i) When $\overline{\eta} = 0$, it means that the constraints of the original problem $\mathbf{P}_2$ are incompatible themselves. In such a situation, there usually exist no feasible points for the formulated original model. Nevertheless, we can always guarantee a non-empty feasible region of the system model as long as the physical scenario is properly defined and characterized.

ii) When $\overline{\eta} = 1$, it means that the constraints of the subproblem $\mathbf{P}_3$ guarantee a non-empty feasible region, from which we can solve a feasible search direction $\mathbf{r}_k$.

iii) When $\overline{\eta} \in (0,1)$, we can specify a real value for $\eta \in (0,\overline{\eta}]$ and substitute the constraints of $\mathbf{P}_3$ by those of $\mathbf{P}_4$, namely, re-constructing a new programming subproblem $\mathbf{P}_5$ to obtain a feasible search direction $\mathbf{r}_k$ as follows

$$\mathbf{P}_5: \min_{\mathbf{r}} :\nabla_{\mathbf{X}} f(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} + 0.5\mathbf{r}^{\mathrm{T}}\mathbf{Q}_k\mathbf{r}$$
$$\text{s.t.} \begin{cases} \eta C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} = 0, i \in \mathcal{I}; \\ \eta C_{j_1}(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_{j_1}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j_1 \in \mathcal{J}_1; \\ C_{j_2}(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_{j_2}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j_2 \in \mathcal{J}_2. \end{cases} \quad (25)$$

In summary, based on the convex programming subproblem, i.e., either $\mathbf{P}_3$ or $\mathbf{P}_5$, we are enabled to solve a feasible search direction $\mathbf{r}_k$ for (21) to generate a new iterate.

B. Positive-Definite Hessian Approximation

To proceed, another key step is to adapt the positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, $\mathbf{Q}_k$. Here, we exploit the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method to construct the updating formula as follows

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k - \frac{\mathbf{Q}_k\mathbf{D}_k\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k^{\mathrm{T}}}{\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k} + \frac{\mathbf{z}_k\mathbf{z}_k^{\mathrm{T}}}{\mathbf{z}_k^{\mathrm{T}}\mathbf{D}_k}, \quad (26)$$

where $\mathbf{D}_k$ is defined as $\mathbf{D}_k = \mathbf{X}_{k+1} - \mathbf{X}_k$ and $\mathbf{z}_k$ is formulated as a linear combination of $\mathbf{Q}_k\mathbf{D}_k$ and

$$\mathbf{y}_k = \nabla_{\mathbf{X}} L(\mathbf{X}_{k+1}, \boldsymbol{\lambda}_k) - \nabla_{\mathbf{X}} L(\mathbf{X}_k, \boldsymbol{\lambda}_k), \quad (27)$$

namely, $\mathbf{z}_k = \omega\mathbf{y}_k + (1-\omega)\mathbf{Q}_k\mathbf{D}_k$ where $\omega \in [0,1]$. The combination coefficient ω is designed so as to keep the Hessian approximation positive-definite, which is given as follows.

$$\omega = \begin{cases} 1, \mathbf{y}_k^{\mathrm{T}}\mathbf{D}_k \geq \xi\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k; \\ \frac{(1-\xi)\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k}{\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k - \mathbf{y}_k^{\mathrm{T}}\mathbf{D}_k}, \text{otherwise}, \end{cases} \quad (28)$$

where we let $\xi \in (0,1)$.

Using (28), we can obtain the following result.

Theorem 1: Suppose that $\mathbf{Q}_k$ is a positive-definite Hessian matrix and $\xi \in (0,1)$ is given. The positive definiteness of $\mathbf{Q}_{k+1}$ generated from (26) is also well guaranteed.

Proof: According to the BFGS formula, the positive definiteness of $\mathbf{Q}_{k+1}$ is maintained as long as keeping $\mathbf{z}_k^{\mathrm{T}}\mathbf{D}_k > 0$. Hence, we only need to show $\mathbf{z}_k^{\mathrm{T}}\mathbf{D}_k > 0$ in both cases of (28) so as to complete the proof. First, it can be seen that $\mathbf{z}_k = \mathbf{y}_k$ when $\omega = 1$ in the first case of (28). According to the positive definiteness of $\mathbf{Q}_k$ and $\xi > 0$, it holds that $\mathbf{z}_k^{\mathrm{T}}\mathbf{D}_k = \mathbf{y}_k^{\mathrm{T}}\mathbf{D}_k \geq \xi\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k > 0$. Otherwise, in the second case of (28), by substituting the second expression on ω into $\mathbf{z}_k$, we can yield $\mathbf{D}_k^{\mathrm{T}}\mathbf{z}_k = \xi\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k > 0$. At this point, the theorem is proven. ■

C. Line Search for Step Size

From (21), it is seen that the step size α_k also plays an important role in constructing an iterative point. To determine a proper step size that not only generates a sufficient decrease in the objective function of $\mathbf{P}_2$, i.e., $f(\mathbf{X})$, along with the direction $\mathbf{r}_k$ at each iteration k but also makes the iteration converge to the feasible region of $\mathbf{P}_2$, we transform a single-variable constrained searching problem for α_k into an unconstrained optimization by using an exact penalty function-based method. To be specific, we formulate an exact l_1-penalty function as a metric function for linear searching α_k as follows

$$W(\mathbf{X}, \boldsymbol{\mu}) = f(\mathbf{X}) + \sum_{i\in\mathcal{I}} \mu_i |C_i(\mathbf{X})| + \sum_{j\in\mathcal{J}} \mu_j \left[-C_j(\mathbf{X})\right]^{+} \quad (29)$$

where $\boldsymbol{\mu} = \mathrm{col}\{\mu_l, l \in \mathcal{I} \cup \mathcal{J}\}$ are penalty coefficients and $[x]^{+} = \max\{0, x\}$. The idea of the exact penalty function (29) is to incorporate the inequality and the equality constraints into the metric function via combining a penalty parameters $\boldsymbol{\lambda}$. In this way, any violation of the constraints will be penalized. To adapt these penalty coefficients during iterations, we further exploit Powell's updating formula based on the Lagrangian multipliers for updating these penalty coefficients [47]

$$\mu_{l,k} = \begin{cases} |\lambda_{l,k}|, k = 1; \\ \max\left\{|\lambda_{l,k}|, \frac{1}{2}\left(|\lambda_{l,k}| + \mu_{l,k-1}\right)\right\}, k \geq 2, \end{cases} \quad (30)$$

for all $l \in \mathcal{I} \cup \mathcal{J}$, where $\lambda_{l,k}$ and $\mu_{l,k}$ are the Lagrangian multiplier and the penalty coefficient corresponding to the l-th constraint at iteration k, respectively. It is noted that

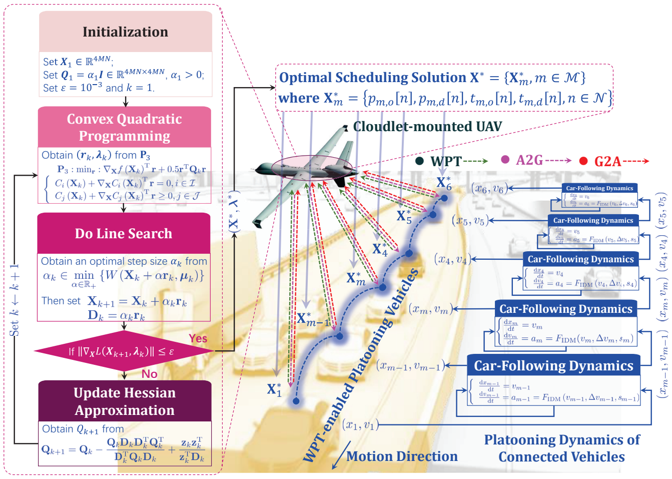

Initialization
Set $X_1 \in \mathbb{R}^{4MN}$;
Set $Q_1 = \alpha_1 I \in \mathbb{R}^{4MN\times 4MN}$, $\alpha_1 > 0$;
Set $\varepsilon = 10^{-3}$ and $k = 1$.
Convex Quadratic Programming
Obtain (r_k, λ_k) from P_3
$\mathbf{P}_3 : \min_{\mathbf{r}} : \nabla_{\mathbf{X}} f(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} + 0.5\mathbf{r}^{\mathrm{T}}\mathbf{Q}_k\mathbf{r}$
$C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} = 0, i \in \mathcal{I}$
$C_j(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_j(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r} \geq 0, j \in \mathcal{J}$
Do Line Search
Obtain an optimal step size α_k from
$\alpha_k \in \min_{\alpha\in\mathbb{R}_+} \{W(\mathbf{X}_k + \alpha\mathbf{r}_k, \boldsymbol{\mu}_k)\}$
Then set $\mathbf{X}_{k+1} = \mathbf{X}_k + \alpha_k\mathbf{r}_k$
$\mathbf{D}_k = \alpha_k\mathbf{r}_k$
If $\|\nabla_X L(X_{k+1}, \lambda_k)\| \leq \varepsilon$
Yes
No
Update Hessian Approximation
Obtain Q_{k+1} from
$\mathbf{Q}_{k+1} = \mathbf{Q}_k - \frac{\mathbf{Q}_k\mathbf{D}_k\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k^{\mathrm{T}}}{\mathbf{D}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{D}_k} + \frac{\mathbf{z}_k\mathbf{z}_k^{\mathrm{T}}}{\mathbf{z}_k^{\mathrm{T}}\mathbf{D}_k}$
Set $k \leftarrow k+1$
$(\mathbf{X}^*, \boldsymbol{\lambda}^*)$
Optimal Scheduling Solution $\mathbf{X}^* = \{\mathbf{X}_m^*, m \in \mathcal{M}\}$
where $\mathbf{X}_m^* = \{p_{m,o}[n], p_{m,d}[n], t_{m,o}[n], t_{m,d}[n], n \in \mathcal{N}\}$
Cloudlet-mounted UAV
● WPT
● A2G
● G2A
$\mathbf{X}_6^*$
$\mathbf{X}_5^*$
$\mathbf{X}_4^*$
$\mathbf{X}_m^*$
$\mathbf{X}_{m-1}^*$
$\mathbf{X}_1^*$
(x_6, v_6)
(x_5, v_5)
(x_4, v_4)
(x_m, v_m)
(x_{m-1}, v_{m-1})
(x_1, v_1)
Car-Following Dynamics
$\frac{\mathrm{d}x_6}{\mathrm{d}t} = v_6$, $\frac{\mathrm{d}v_6}{\mathrm{d}t} = a_6 = F_{\mathrm{IDM}}(v_6, \Delta v_6, s_6)$
Car-Following Dynamics
$\frac{\mathrm{d}x_5}{\mathrm{d}t} = v_5$, $\frac{\mathrm{d}v_5}{\mathrm{d}t} = a_5 = F_{\mathrm{IDM}}(v_5, \Delta v_5, s_5)$
Car-Following Dynamics
$\frac{\mathrm{d}x_4}{\mathrm{d}t} = v_4$, $\frac{\mathrm{d}v_4}{\mathrm{d}t} = a_4 = F_{\mathrm{IDM}}(v_4, \Delta v_4, s_4)$
Car-Following Dynamics
$\frac{\mathrm{d}x_m}{\mathrm{d}t} = v_m$, $\frac{\mathrm{d}v_m}{\mathrm{d}t} = a_m = F_{\mathrm{IDM}}(v_m, \Delta v_m, s_m)$
Car-Following Dynamics
$\frac{\mathrm{d}x_{m-1}}{\mathrm{d}t} = v_{m-1}$, $\frac{\mathrm{d}v_{m-1}}{\mathrm{d}t} = a_{m-1} = F_{\mathrm{IDM}}(v_{m-1}, \Delta v_{m-1}, s_{m-1})$
WPT-enabled Platooning Vehicles
Motion Direction
Platooning Dynamics of Connected Vehicles

Fig. 2. The implementation framework of the proposed method.

such penalty coefficients allow positive contribution from the constraints that are inactive in the solution of the subproblem $\mathbf{P}_3$ but were recently active. The updating of $\mu_{l,k}$ based on (30) can make $W(\mathbf{X}, \boldsymbol{\mu})$ behave in a similar way as the Lagrangian function, such that $\mathbf{r}_k$ can also guarantee a decrease in $W(\mathbf{X}, \boldsymbol{\mu})$.

Using (29) and (30), we further propose a line search model for obtaining a proper α_k at k as follows

$$\alpha_k \in \min_{\alpha\in\mathbb{R}_+} \left\{W\left(\mathbf{X}_k + \alpha\mathbf{r}_k, \boldsymbol{\mu}_k\right)\right\}, \tag{31}$$

which can be efficiently solved by using some existing unconstrained optimization algorithms such as the Newton's method, the conjugate gradient method, the gradient projection method, the quadratic interpolation methods, and some others [48].

D. Convergence and Complexity Analysis

Following the above subsections, we further show the overall implementation framework of the proposed algorithm in Fig. 2. As illustrated in the proposed framework, the key computation complexity lies in solving the convex quadratic programming $\mathbf{P}_3$ (or $\mathbf{P}_5$), which, nonetheless, can be effectively handled by existing convex optimization techniques such as the active set method, the interior point method [48], the distributed alternating direction method of multipliers technique [49], etc. In addition, the line search for obtaining an appropriate step size α_k at each k can also be achieved by using a simple quadratic interpolation-based one-dimensional search method. In the following, we show that the convergence of the proposed method in Fig. 2 can be well guaranteed and discuss its computational complexity in detail.

To proceed, let $\mathcal{V} = \mathcal{I} \cup \mathcal{J}$ and further denote $\Phi(\mathbf{X}) = \max_{l\in\mathcal{V}}\{C_l(\mathbf{X})\}$. We can provide two lemmas as follows, the proofs of which are detailed in Appendix B.

Lemma 1: For any $|C_l(\mathbf{X})|$, $l \in \mathcal{V}$, it always holds that

$$|C_l(\mathbf{X})| = \max\left\{C_l(\mathbf{X}), -C_l(\mathbf{X})\right\}, \ l \in \mathcal{V}. \tag{32}$$

Lemma 2: Given any directional column vector $\mathbf{d} \in \mathbb{R}^{4MN}$, the directional derivative of $\Phi(\mathbf{X})$ with respect to $\mathbf{d}$ exists and can be expressed as follows

$$\Phi'(\mathbf{X}, \mathbf{d}) = \max_{l\in\mathcal{V}(\mathbf{X})}\left\{\nabla_{\mathbf{X}} C_l(\mathbf{X})^{\mathrm{T}}\mathbf{d}\right\}, \tag{33}$$

where $\mathcal{V}(\mathbf{X}) = \{l : C_l(\mathbf{X}) = \Phi(\mathbf{X}), l \in \mathcal{V}\}$.

Combining Lemmas 1 and 2, we get the following result.

Theorem 2: Suppose that $(\mathbf{X}_k, \boldsymbol{\lambda}_k)$ is a pair of iterative points satisfying the Karush-Kuhn-Tucker (KKT) conditions, i.e., a pair of KKT points, and the penalty parameters $\boldsymbol{\mu}_k$ are updated by following (30). For any non-zero direction vector $\mathbf{r}_k \neq \mathbf{0}$ obtained from $\mathbf{P}_3$, $\mathbf{r}_k$ is also a descent direction for the metric function $W(\mathbf{X}_k, \boldsymbol{\mu}_k)$ at $\mathbf{X}_k$, i.e., $W'(\mathbf{X}_k, \boldsymbol{\mu}_k, \mathbf{r}_k) < 0$.

Proof: For the sake of simplifying notations, we introduce six different index subsets of $\mathcal{V}$ as follows

$$\begin{cases} \mathcal{V}_1 = \{i \in \mathcal{I} : C_i(\mathbf{X}) < 0\}, \\ \mathcal{V}_2 = \{i \in \mathcal{I} : C_i(\mathbf{X}) = 0\}, \\ \mathcal{V}_3 = \{i \in \mathcal{I} : C_i(\mathbf{X}) > 0\}, \\ \mathcal{V}_4 = \{i \in \mathcal{J} : C_i(\mathbf{X}) < 0\}, \\ \mathcal{V}_5 = \{i \in \mathcal{J} : C_i(\mathbf{X}) = 0\}, \\ \mathcal{V}_6 = \{i \in \mathcal{J} : C_i(\mathbf{X}) > 0\}. \end{cases} \tag{34}$$

Based on Lemmas 1 and 2, we can rearrange (29) and get

$$\begin{aligned} W'(\mathbf{X}_k, \boldsymbol{\mu}_k, \mathbf{r}_k) =& \nabla_{\mathbf{X}} f(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &- \sum_{l\in\mathcal{V}_1} \mu_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &+ \sum_{l\in\mathcal{V}_2} \mu_{l,k} \left|\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right| \\ &+ \sum_{l\in\mathcal{V}_3} \mu_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &- \sum_{l\in\mathcal{V}_4} \mu_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &+ \sum_{l\in\mathcal{V}_5} \mu_{l,k} \max\left\{0, -\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right\}. \end{aligned} \tag{35}$$

Since $(\mathbf{X}_k, \boldsymbol{\lambda}_k)$ is a KKT pair of $\mathbf{P}_3$, we have the following gradient condition

$$\nabla_{\mathbf{X}} f(\mathbf{X}_k) + \mathbf{Q}_k\mathbf{r}_k - \sum_{l\in\mathcal{V}} \lambda_{l,k} C_l(\mathbf{X}_k) = 0. \tag{36}$$

Combining (35) and (36), we can further yield

$$\begin{aligned} W'(\mathbf{X}_k, \boldsymbol{\mu}_k, \mathbf{r}_k) =& -\mathbf{r}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{r}_k + \sum_{l\in\mathcal{V}_1} \left(\lambda_{l,k} - \mu_{l,k}\right) \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &+ \sum_{l\in\mathcal{V}_2} \left[\lambda_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k + \mu_{l,k} \left|\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right|\right] \\ &+ \sum_{l\in\mathcal{V}_3} \left(\lambda_{l,k} + \mu_{l,k}\right) \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &- \sum_{l\in\mathcal{V}_4} \left(\lambda_{l,k} - \mu_{l,k}\right) \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \\ &+ \sum_{l\in\mathcal{V}_5} \left[\lambda_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k + \mu_{l,k} \max\left\{0, -\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right\}\right] \\ &+ \sum_{l\in\mathcal{V}_6} \lambda_{l,k} \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k. \end{aligned} \tag{37}$$

According to the condition that $(\mathbf{X}_k, \boldsymbol{\lambda}_k)$ is a KKT pair of $\mathbf{P}_3$, we can have the equality $C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = 0$ and the following complementary slackness conditions

$$\begin{cases} \lambda_{j,k} \geq 0, \\ C_j(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_j(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \geq 0, \\ \lambda_{j,k}\left(C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right) = 0, \end{cases} \tag{38}$$

for all $j \in \mathcal{J}$.

Recall that the specific formula (30) guarantees $\mu_{l,k} \geq |\lambda_{l,k}|$ for all $l \in \mathcal{V}$ and $\mathbf{Q}_k$ is positive definite. We can examine the sign of $W'(\mathbf{X}_k, \boldsymbol{\mu}_k, \mathbf{r}_k)$ in the following six cases:

i) With $l \in \mathcal{V}_1$, we get $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = -C_l(\mathbf{X}_k) > 0$ and thus see $(\lambda_{l,k} - \mu_{l,k})\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \leq 0$.

ii) With $l \in \mathcal{V}_2$, we get $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = -C_l(\mathbf{X}_k) = 0$.

iii) With $l \in \mathcal{V}_3$, we get $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = -C_l(\mathbf{X}_k) < 0$ and thus see $(\lambda_{l,k} + \mu_{l,k})\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \leq 0$.

iv) With $l \in \mathcal{V}_4$, we get $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \geq -C_l(\mathbf{X}_k) > 0$ and thus see $(\lambda_{l,k} - \mu_{l,k})\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \leq 0$.

v) With $l \in \mathcal{V}_5$, we get $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \geq -C_l(\mathbf{X}_k) = 0$ and thus see $\max\left\{0, -\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right\} = 0$. In such a case, it also holds that $\lambda_{l,k}(C_l(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k) = 0$, which further leads to $\lambda_{l,k}\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = -\lambda_{l,k} C_l(\mathbf{X}_k) = 0$.

vi) With $l \in \mathcal{V}_6$, we can get $\lambda_{l,k}\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = -\lambda_{l,k} C_l(\mathbf{X}_k) \leq 0$.

Now, combining the results of the above cases with (37) can obtain

$$W'(\mathbf{X}_k, \boldsymbol{\mu}_k, \mathbf{r}_k) \leq -\mathbf{r}_k^{\mathrm{T}}\mathbf{Q}_k\mathbf{r}_k < 0, \tag{39}$$

which proves the theorem. ■

Lemma 3: Given $\varphi > 0$ and $\{\epsilon_k > 0, \forall k\}$ where $\sum_{k=1}^{\infty}\epsilon_k < +\infty$, it always holds that

$$W(\mathbf{X}_k + \alpha_k\mathbf{r}_k, \boldsymbol{\mu}_k) \leq \min_{0 \leq \alpha \leq \varphi} W(\mathbf{X}_k + \alpha\mathbf{r}_k, \boldsymbol{\mu}_k) + \epsilon_k, \tag{40}$$

for any α_k obtained from (31).

Lemma 3 can be immediately proven by the definition of α_k given in (31) and Theorem 2. Based on the results above, we can further obtain the global convergence as follows.

Theorem 3: Suppose that ζ_1 and ζ_2 are two positive real numbers such that

$$\zeta_1\|\mathbf{X}\|^2 \leq \mathbf{X}^{\mathrm{T}}\mathbf{Q}_k\mathbf{X} \leq \zeta_2\|\mathbf{X}\|^2 \tag{41}$$

holds for any $\mathbf{X} \in \mathbb{R}^{4MN}$ and all $k \geq 1$. Let the sequence of the penalty parameters $\{\boldsymbol{\mu}_k, \forall k \geq 1\}$ follow (30) and $\boldsymbol{\lambda}_k$ are the Lagrangian multipliers of (22) at each $k \geq 1$, such that they always guarantee

$$|\lambda_{l,k}| \leq \mu_{l,k} \tag{42}$$

for all $l \in \mathcal{V}$ and all $k \geq 1$. The sequence of iterative points $\{\mathbf{X}_k, \forall k \geq 1\}$ generated from the proposed algorithm in Fig. 2 can either stop at a KKT point of the original problem $\mathbf{P}_2$ or converge to a KKT point.

Proof: If there exists a specific iteration k such that $\mathbf{r}_k = \mathbf{0}$ holds, based on the KKT conditions of $\mathbf{P}_3$, we can obtain following results

$$\begin{cases} \nabla_{\mathbf{X}} f(\mathbf{X}_k) + \mathbf{Q}_k\mathbf{r}_k - \sum\limits_{l \in \mathcal{V}} \lambda_{l,k}\nabla_{\mathbf{X}} C_l(\mathbf{X}_k) = \mathbf{0}, \\ C_i(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_i(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k = 0, i \in \mathcal{I}, \\ \lambda_{j,k}\left[C_j(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_j(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k\right] = 0, j \in \mathcal{J}, \\ C_j(\mathbf{X}_k) + \nabla_{\mathbf{X}} C_j(\mathbf{X}_k)^{\mathrm{T}}\mathbf{r}_k \geq 0, j \in \mathcal{J}, \\ \lambda_{j,k} \geq 0, j \in \mathcal{J}. \end{cases} \tag{43}$$

Combining $\mathbf{r}_k = \mathbf{0}$ and (43) given above, we can see that $\mathbf{X}_k$ is also a KKT point of the problem $\mathbf{P}_2$. Hence, in such a situation, the algorithm stops and the solution sequence arrives at the KKT point.

If $\mathbf{r}_k \neq \mathbf{0}$ holds for any $k \geq 1$, $\{\mathbf{X}_k, \forall k \geq 1\}$ is an infinite sequence. According to the boundedness of the positive definite Hessian approximation $\mathbf{Q}_k$ and the Lagrangian multipliers $\boldsymbol{\lambda}_k$, we denote the accumulation points of $\{\mathbf{X}_k, \forall k \geq 1\}$, $\{\mathbf{Q}_k, \forall k \geq 1\}$ and $\{\boldsymbol{\lambda}_k, \forall k \geq 1\}$ as follows

$$\begin{cases} \lim\limits_{\substack{k \in \mathbb{N}_+ \\ k \to \infty}} \mathbf{X}_k = \mathbf{X}^*, \\ \lim\limits_{\substack{k \in \mathbb{N}_+ \\ k \to \infty}} \mathbf{Q}_k = \mathbf{Q}^*, \\ \lim\limits_{\substack{k \in \mathbb{N}_+ \\ k \to \infty}} \boldsymbol{\lambda}_k = \boldsymbol{\lambda}^*, \end{cases} \tag{44}$$

where $\mathbb{N}_+$ is an infinite subsequence of the natural number sequence $\{1, 2, \ldots\}$. Besides, according to the KKT conditions (43) and based on the continuity of $\nabla_{\mathbf{X}} f(\mathbf{X}_k)$ and $\nabla_{\mathbf{X}} C_l(\mathbf{X}_k)$ for $l \in \mathcal{V}$ and the positive definiteness of $\mathbf{Q}_k$, the sequence $\{\mathbf{r}_k, \forall k \in \mathbb{N}_+\}$ also converges to a vector $\mathbf{r}^*$ such that $(\mathbf{r}^*, \boldsymbol{\lambda}^*)$ is the KKT pair of $\mathbf{P}_3$ where $\mathbf{X}_k$ and $\mathbf{Q}_k$ are substituted by $\mathbf{X}^*$ and $\mathbf{Q}^*$, respectively. At this point, we only need to show $\mathbf{r}^* = \mathbf{0}$ so as to prove the theorem. For this goal, we provide the proof by contradiction as follows.

Assuming $\mathbf{r}^* \neq \mathbf{0}$, we can set an $\overline{\alpha} > 0$ such that

$$W(\mathbf{X}^* + \overline{\alpha}\mathbf{r}^*, \boldsymbol{\mu}) = \min_{0 \leq \alpha \leq \varphi} W(\mathbf{X}^* + \alpha\mathbf{r}^*, \boldsymbol{\mu}). \tag{45}$$

According to Theorem 2, we can see

$$W(\mathbf{X}^* + \overline{\alpha}\mathbf{r}^*, \boldsymbol{\mu}) < W(\mathbf{X}^*, \boldsymbol{\mu}). \tag{46}$$

Thus, we can define a value ΔW by

$$\Delta W = W(\mathbf{X}^*, \boldsymbol{\mu}) - W(\mathbf{X}^* + \overline{\alpha}\mathbf{r}^*, \boldsymbol{\mu}) > 0. \tag{47}$$

Due to $\mathbf{X}_k + \overline{\alpha}\mathbf{r}_k \to \mathbf{X}^* + \overline{\alpha}\mathbf{r}^*$ for $k \in \mathbb{N}_+$ and according to (47), we have

$$W(\mathbf{X}_k + \overline{\alpha}\mathbf{r}_k, \boldsymbol{\mu}) + \frac{\Delta W}{2} < W(\mathbf{X}^*, \boldsymbol{\mu}). \tag{48}$$

On the other side, according to

$$W(\mathbf{X}_{k+1}, \boldsymbol{\mu}) \leq W(\mathbf{X}_k, \boldsymbol{\mu}) + \epsilon_k, \forall k, \tag{49}$$

and

$$\sum_{q=k}^{\infty}\epsilon_q < \frac{\Delta W}{2} \tag{50}$$

for a sufficiently large $k \in \mathbb{N}_+$, we can derive the following result based on Lemma 3

$$\begin{aligned} W(\mathbf{X}^*, \boldsymbol{\mu}) &\leq W(\mathbf{X}_{k+1}, \boldsymbol{\mu}) + \sum_{q=k+1}^{\infty} \epsilon_q \\ &\leq \min_{0 \leq \alpha \leq \varphi} W(\mathbf{X}_k + \alpha \mathbf{r}_k, \boldsymbol{\mu}) + \epsilon_k + \sum_{q=k+1}^{\infty} \epsilon_q \\ &< W(\mathbf{X}_k + \overline{\alpha} \mathbf{r}_k, \boldsymbol{\mu}) + \frac{\Delta W}{2} < W(\mathbf{X}^*, \boldsymbol{\mu}), \end{aligned} \tag{51}$$

which is contradictory. Therefore, $\mathbf{r}^* = \mathbf{0}$ must hold and $\mathbf{X}^*$ is the KKT point of $\mathbf{P}_2$. The theorem is proven. ■

Additionally, we also remark that since the proposed method exploits the quasi-Newton Hessian approximation $\mathbf{Q}_k$ to replace the Lagrangian Hessian $\nabla^2_{\mathbf{XX}} L(\mathbf{X}_k, \boldsymbol{\lambda}_k)$ and according to Theorem 1, the iterative algorithm can guarantee a superlinear convergence rate, i.e., $\mathbf{X}_k$ converging to $\mathbf{X}^*$ with a superlinear rate under $k \to \infty$ [48] (See Theorem 18.5 in [48]). The computational complexity of the proposed algorithm is dominated by the convex quadratic programming $\mathbf{P}_3$ and the line search problem (31). We remark that the size of $\mathbf{Q}_k$ is $4MN \times 4MN$ and the total number of the linear constraints is $|\mathcal{I}| + |\mathcal{J}| = 2N + 12MN$. By using the interior point method for $\mathbf{P}_3$, the computational complexity can be in the order of $\mathcal{O}(\sqrt{4MN}L)$ where L denotes the total number of input data bits for the algorithm implementation [50] and is in the order of $\mathcal{O}(4MN + 16M^2N^2 + (2N + 12MN) \times 4MN + (2N + 12MN))$ in our problem setup. The number of the platooning vehicles, M, and the number of available time slots, N, indicate the network scale and the resource scale, respectively. It is seen above that both the levels of M and N jointly dominate the complexity order.

To analyze the computational complexity of the line search procedure involved in the proposed method, we first let $\{\alpha[t], \forall t \geq 1\}$ be a sequence of iterative points generated by a line search method, which can converge to the optimal step size α_k as given in (31), i.e., $\lim_{t\to\infty} \alpha[t] = \alpha_k$. Here, t is the index of the iterations in the line search. By using a two-point quadratic interpolation-based method for solving (31), the superlinear convergence rate of $\psi = \frac{1+\sqrt{5}}{2}$ can also be well guaranteed [48]. Thus, according to the property of the superlinear convergence, we can see that there must exist a real constant $\chi \in \mathbb{R}$ such that $|\alpha[t+1] - \alpha_k| \leq \chi |\alpha[t] - \alpha_k|^{\psi}$. Based on this result, given a specific numerical tolerance ε, the computational complexity of the line search using the two-point quadratic interpolation is

$$K_{\text{ls}} = \mathcal{O}\left(\frac{\ln\left(\frac{\ln(\varepsilon) - \frac{\ln(\chi)}{1-\psi}}{\ln(|\alpha[1] - \alpha_k|) - \frac{\ln(\chi)}{1-\psi}} \right)}{\ln(\psi)} \right), \tag{52}$$

where $\alpha[1]$ is the initial iterate in the line search.

Let the total number of iterations needed in the outer loop of the proposed algorithm be K_{ol}. Based on the results above, the overall computational complexity of the proposed algorithm is $K_{\text{ol}} K_{\text{ls}} \mathcal{O}(\sqrt{4MN}L)$, where $L \sim \mathcal{O}(4MN + 16M^2N^2 + (2N + 12MN) \times 4MN + (2N + 12MN))$. As can be seen, the dimension of the feasible solution $\mathbf{X}$ relies on the number of the platooning vehicles M, and thus the computational complexity of the proposed method is significantly influenced by the vehicle number M. Increasing M will result in higher computational complexity. Nevertheless, the above analysis results show that the proposed method can be implemented effectively with polynomial-time complexity. From Fig. 2, our proposed method transforms the original problem into a quadratic programming subproblem at each iterate, which itself can guarantee the implementation robustness of the algorithm since it can circumvent the nonlinearity and complexity associated with the original resource optimization problem. Besides, we need to remark that in order to guarantee the robust performance of the proposed system framework as shown in Fig. 2, the accurate and real-time mobility information of the flying UAV in the air and the platooning vehicles on the ground is needed and the sensor information is treated as the parametric input for the practical implementation of the proposed optimization algorithm. The motion of the UAV and the platooning vehicles should also be coordinated by a robust flight controller and a car-following controller, respectively. The performance of the entire air-ground cooperative system will depend on the sensor information accuracy and the stability of the nodes' controllers. The system state estimation and platoon control issues can be addressed by exploiting some existing filter design (e.g., the Bayesian Kalman filter [51] and particle filter [52]) and robust platoon control techniques (e.g., the robust H-infinity controller [3], [4] and consensus controller [5]). Thus, our proposed joint optimization method as in Fig. 2 can be integrated with these robust modern control techniques to facilitate the practical system deployment.

V. Performance Evaluation

This section presents the simulation experiments and evaluates the convergence and performance of our proposed method. We first provide details on our simulation settings. Then, the global convergence of the proposed method is validated. Finally, we compare our proposed method with three other benchmarks in terms of the system computation capacity. Besides, our simulation experiments have been conducted by using a MATLAB/Simulink-based desktop software development platform, in which a high-performance workstation with the processor configurations of Intel Core i9-9900K CPU @ 3.6GHz-5.0GHz and RAM 64GB is adopted, and hybrid programming and code automatic generation techniques are jointly used to implement the proposed system models and the proposed joint resource optimization algorithm. The mobility models of the UAV and the platooning vehicles, the physical-layer A2G/G2A communication model and the nodes' local computation model are integrated to realize our simulations.

A. Simulation Settings

In our simulation scenario, a platoon of moving vehicles on the ground and a UAV cruising in the same direction as that of the vehicle platoon are considered to cooperate in task computation. An specific application case is shown in Fig. 1. We properly configure the simulation parameters related to the

TABLE I
SIMULATION PARAMETERS

Parameter	Symbol	Value
Number of time slots	N	200
Slot duration	τ	0.2 s
Communication bandwidth	B	4×10^7 Hz
Noise power	N_0	10^{-9} W
Channel power gain at $d_0 = 1$ m	g_0	-50 dB
UAV's maximum CPU frequency	f	2×10^9 cyc/s
Number of CPU cycles	C	10^3 cyc/bit
Energy conservation efficiency	η_0	0.8
Power of energy supply	P_{sup}	0.2 W
UAV's reserved energy at a slot	ϵ	0.2 J
Effective capacitance coefficient	λ_c	10^{-28}
Desired velocity of platoon	$v_{\max}$	30 m/s
Platoon's maximum acceleration	$a_{\max}$	0.73 m/s^2
Platoon's maximum deceleration	$b_{\max}$	1.67 m/s^2
Platoon's minimum reaction time	t_r	1.5 s
Platoon's minimum net distance	$s_{\min}$	2 m
Acceleration exponent	δ	4
Vehicle length	l_0	5 m.

mobility model of the car-following-based vehicle platoon by referring to the literature [43]–[45]. Besides, we also refer to [33]–[35], [39], [40] to set the simulation data for the parameters of the UAV's dynamic model, the physical-layer A2G/G2A communication model, the wireless power transfer model, and the on-board computation model. Specifically, the leading vehicle of the platoon is moving at a constant longitudinal velocity $10\,\text{m/s}$ while the other following vehicles are moving in a car-following behavior. The initial space headway between the platooning vehicles is uniformly and randomly generated from $8\,\text{m}$ to $20\,\text{m}$. The UAV in the air is initially located behind the vehicle platoon. The initial longitudinal distance between the UAV and the tail vehicle in the end of the platoon is set to $20 \times M$ meters. The flight height of the UAV is fixed at $H = 10\,\text{m}$ and its constant cruising velocity is $v_{\text{UAV}} = 13\,\text{m/s}$. The main simulation settings on the physical-layer communication, on-board computation and platooning mobility models are summarized in Table I.

B. Convergence Analysis

The evolution of the computation bits per time slot of the vehicle platoon during iterations is illustrated under different platooning vehicle numbers in Fig. 3. As can be seen, the computation bits of the system, as the optimization metric, are boosted by our proposed method and can converge finally to a locally optimal point after a proper number of iterations. The proposed method can arrive at the steady state under either a small-scale platooning vehicle situation (e.g., with $M = 5$) or a large-scale platooning vehicle situation (e.g., with $M = 12$). For example, when increasing M from $M = 5$ to $M = 12$, the number of iterations needed for guaranteeing convergence does not grow rapidly. The computation bits can still reach the local optimum after about 10 iterations. Regardless of the modeling complexity, the results in Fig. 3 validate the convergence performance of our method.

C. Performance Comparison

In the following subsection, we further compare our proposed Joint Communication and Computation Resource Scheduling method based on successive convex approximation (JointCCRS) with other three conventional approaches as performance benchmarks. Specifically, the compared approaches include i) the optimal resource scheduling method based on the maximum power control (OptRS-MPC) that aims at optimizing the time resource allocation among the platooning vehicles using the maximum power level, ii) the optimal resource scheduling method based on the stochastic power control (OptRS-SPC) that optimally schedules the time resource while the transmission power of the platooning vehicles are stochastically controlled, and iii) the optimal power control method with uniform time allocation (OptPC-URS) that aims at optimizing the power allocation among the platooning vehicles using uniform time allocation.

1) Impact of Platooning Vehicle Number: We first compare the performance of our method with that of the other methods under different settings on the platooning vehicle number M in Fig. 4 and Fig. 5, respectively. Besides, the duration of a time slot τ is also set to different values, i.e., setting $\tau = 0.1\,\text{s}$ in Fig. 4 while $\tau = 0.2\,\text{s}$ in Fig. 5. As can be seen from Fig. 4 and Fig. 5, the total computation bits achieved by the compared methods decline when the vehicle number increases. Since the total computation and communication resources are limited, increasing M reduces the available resources per each node and thus makes the computation-oriented QoS of the system degrade. Besides, since OptRS-SPC adopts stochastic power control mechanism, its performance curve is shown to be slightly fluctuating in these figures. It is also seen that the total computation bits obtained by all the methods under the setting $\tau = 0.1\,\text{s}$ in Fig. 4 are lower than those under $\tau = 0.2\,\text{s}$ in Fig. 5. This is because that a larger slot duration indicates more available time resources. By comparison, the total computation bits achieved by our proposed method, JointCCRS, are much higher than those achieved by the other three schemes with either a small-scale vehicle platoon (e.g., with $M = 5$) or a large-scale vehicle platoon (e.g., with $M = 15$) in both Fig. 4 and Fig. 5. To be specific, the proposed method can improve the system computation capacity by more than 1.47 times on average when compared with the other methods in Fig. 4 and by more than 1.73 times on average in Fig. 5.

2) Impact of Time Slot Number: In Fig. 6 and Fig. 7, we illustrate the impacts of different time slot numbers N on the total computation bits of the system under a small-scale platooning scenario ($M = 6$) and a large-scale platooning scenario ($M = 12$), respectively. From Fig. 6 and Fig. 7, it is seen that the total computation bits achieved by different methods rise with increasing the time slot number N. This is because more time slots indicate more time resources available for processing more computation tasks. Another fact can also be observed from the performance curve obtained by our method, JointCCRS, is that the increment in the system computation performance is reduced with increasing N. For instance, with $M = 6$ in Fig. 6, the growth rate of the total computation bits is slow down after about $N = 300$, while the critical point is $N = 500$ with $M = 12$ in Fig. 7. The main reason is that more platooning vehicles need more time slots to dispose their computational load. Nevertheless, our proposed method can achieve the best performance among the compared

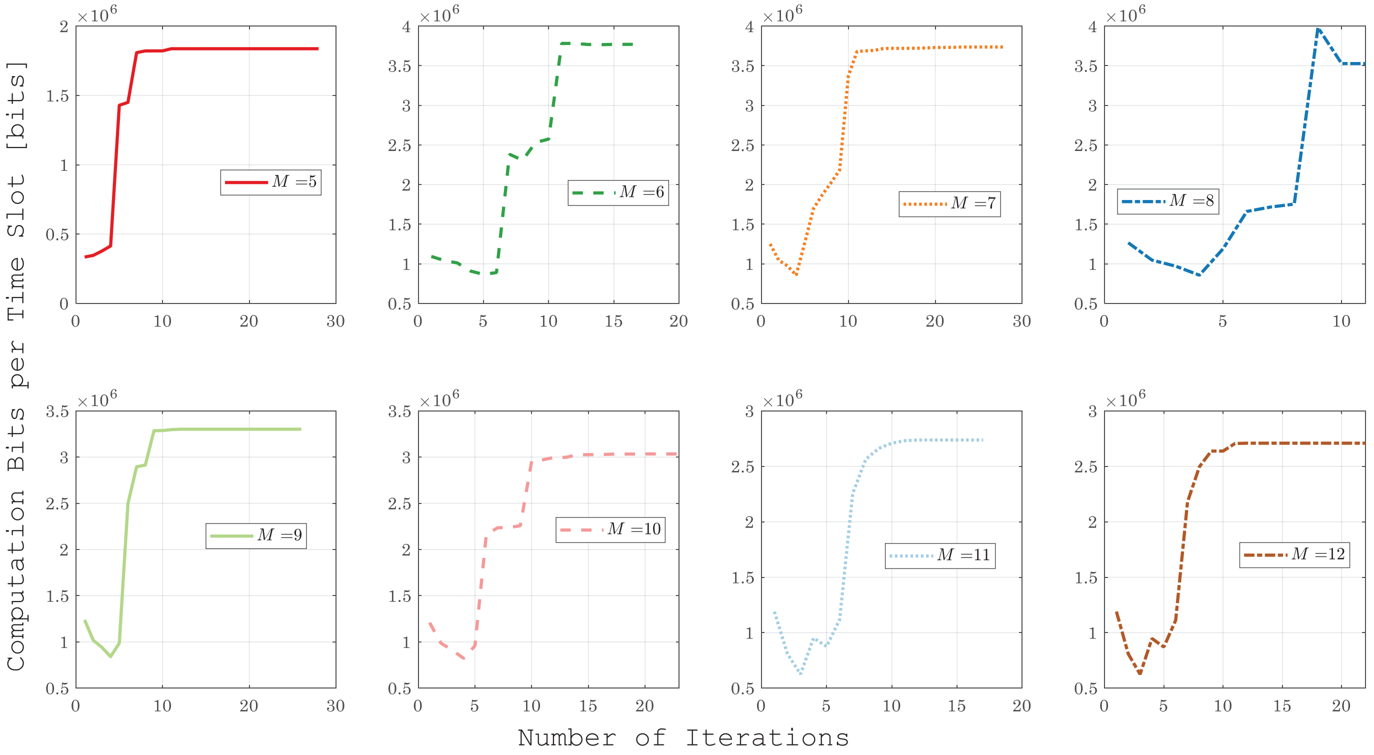


Fig. 3. The computation bits per time slot under different numbers of platooning vehicles.

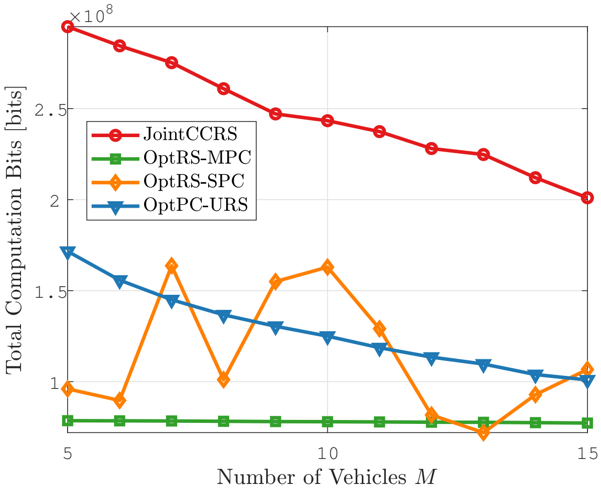


Fig. 4. The total computation bits under different numbers of platooning vehicles with the slot duration $\tau = 0.1\,\mathrm{s}$.

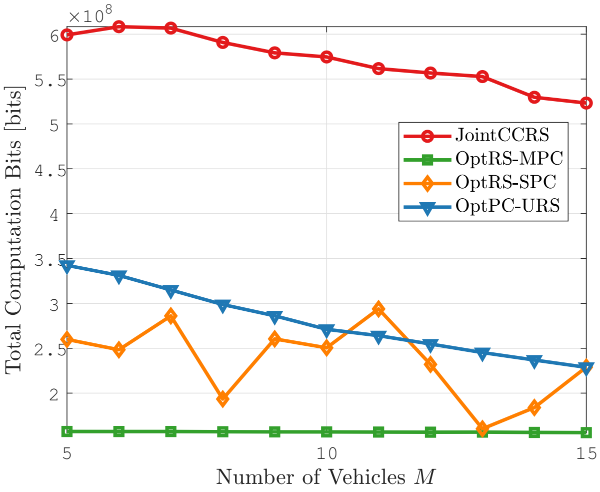


Fig. 5. The total computation bits under different numbers of platooning vehicles with the slot duration $\tau = 0.2\,\mathrm{s}$.

methods, with an average improvement of 136.53% in the total computation bits in Fig. 6 and 158.14% in Fig. 7.

3) Impact of WPT Power: The impacts of different supply power levels P_{sup} used by the platooning vehicles on the system performance are shown in Fig. 8 and Fig. 9. In these two figures, the platooning vehicle number is also set to $M = 6$ and $M = 12$, respectively. We can see that the performance curve obtained by different methods slightly fluctuates against the varying supply power. When comparing both the figures, it is seen that the average performance of the methods under $M = 12$ is lower than that under $M = 6$. This is due to the fact that more platooning vehicles can consume more scheduled resources for their computation tasks. However, the total computation bits achieved by our method are the largest among the compared methods, which are more than 2.03 times of the performance metric achieved by the others on average in Fig. 8 and more than 1.90 times on average in Fig. 9.

4) Impact of Mobility: Fig. 10 shows the total computation bits of different methods versus the relative velocity of the UAV and the leading vehicle $\frac{v_{\text{UAV}}}{v_1[0]}$. The velocity ratio is ranged from 0.1 to 1.9 at the initial slot. We can observe from Fig. 10 that all the compared methods experience an increase and

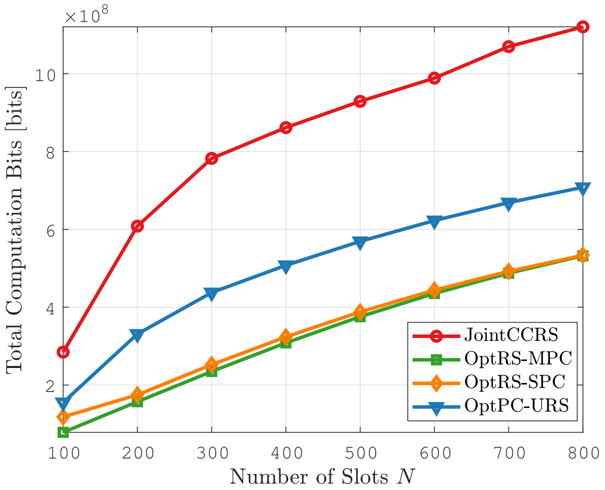


Fig. 6. The total computation bits under different time slot numbers with the platooning vehicle number $M = 6$.

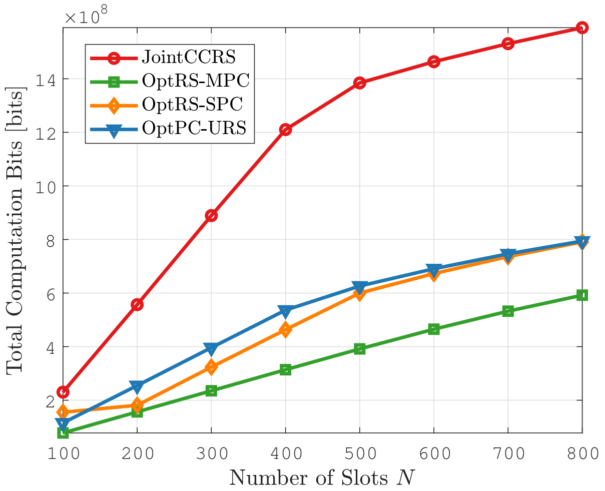


Fig. 7. The total computation bits under different time slot numbers with the platooning vehicle number $M = 12$.

then a decrease in the system performance along with the increasing velocity ratio. There exists a peak in the system performance of each method. It is confirmed that the mobility can significantly affect the G2A/A2G communications so as to affect the computation offloading performance. Based on our method, the peak of the total computation bits can be reached at the initial velocity ratio of about 1. By comparison, our method performs best and our computation capacity is enhanced by about 137.59% on average.

Fig. 11 illustrates the total computation bits of different methods versus the relative initial longitudinal position of the UAV and the leading vehicle. The performance curves obtained by JointCCRS and OptPC-URS show an obvious upward and then downward trends as the relative distance decreases. Specifically, when the initial relative position is ranged within $[90, 120]$ meters, the total computation bits achieved by our method can exceed $600\,\mathrm{MB}$ during $N = 200$ available scheduling slots, and our method can promote the

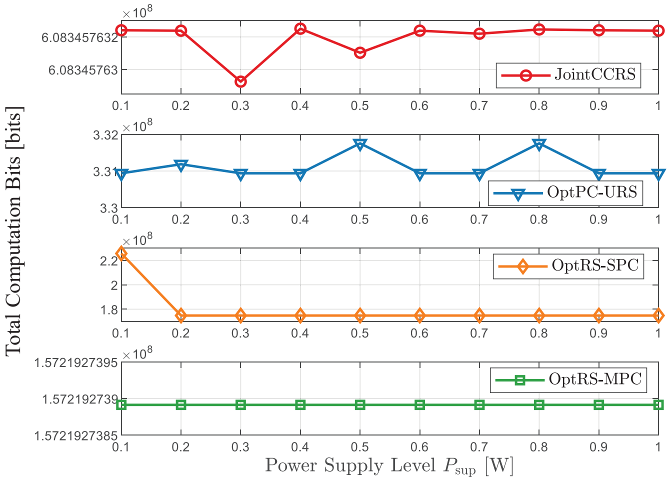


Fig. 8. The total computation bits under different supply power levels with the platooning vehicle number $M = 6$.

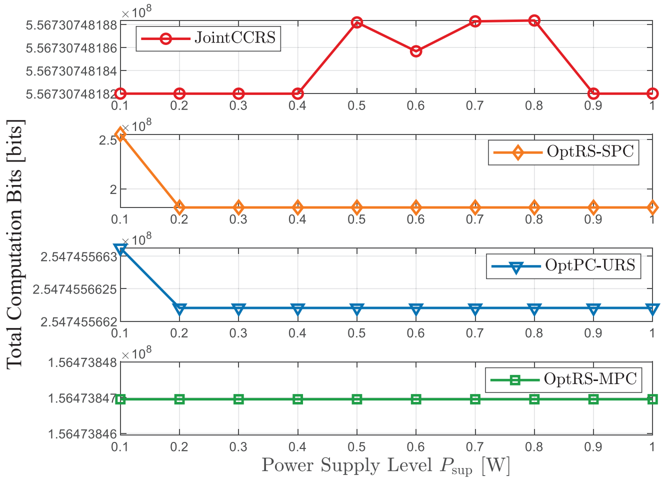


Fig. 9. The total computation bits under different supply power levels with the platooning vehicle number $M = 12$.

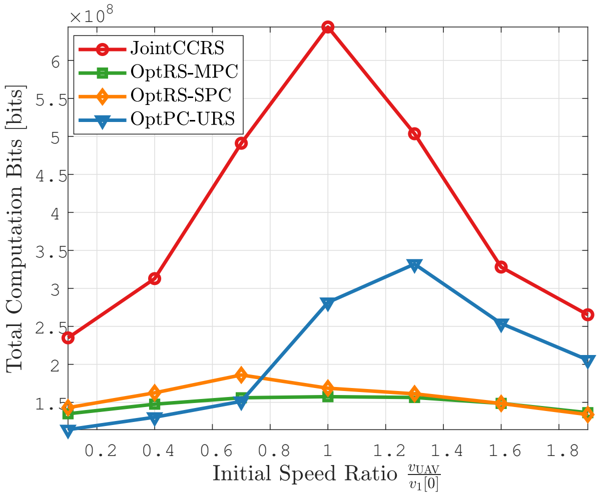


Fig. 10. The total computation bits under different velocity ratios $\frac{v_{\mathrm{UAV}}}{v_1[0]}$.

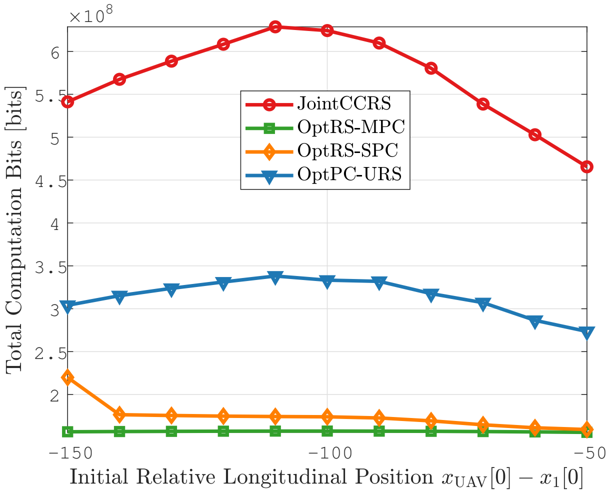


Fig. 11. The total computation bits under different initial longitudinal relative distances $x_{\mathrm{UAV}}-x_1[0]$.

system performance with an average increment of 189.95% when compared with the others.

VI. Conclusion and Future Work

In this paper, we have investigated a UAV-assisted MEC system for platooning vehicles, in which both the UAV and the vehicle platoon can cooperate in communication and computation while the platooning vehicles are enabled with WPT to supply power to the UAV. We have formulated a system optimization model for jointly scheduling the communication and computation resources by optimizing the transmission power and time slot allocation. The coupled dynamics of the platooning dynamics and the A2G/G2A communications is taken into consideration. To handle the complexity in the non-convex optimization problem, we have further proposed a Joint Communication and Computation Resource Scheduling optimization method, jointCCRS, based on successive convex approximation, the convergence of which is proven theoretically and validated numerically. Simulation results obtained under different situations are also provided to demonstrate the superior performance of the proposed method in terms of improving the overall computation capacity of the aerial-ground cooperative network. Our work can offer a joint optimal resource scheduling paradigm for WPT-enabled UAV-CAV cooperative MEC systems considering vehicle platooning dynamics. In the future, we will focus on the joint optimization modeling and algorithm design of the system in terms of cooperative communication, computation, and control.

Appendix A
$C_j(\mathbf{X})$ with j ranging from $N+1$ to $N+11NM$

For j ranges from $N+1$ to $N+11NM$, the inequality constraints from (16c) to (16f) can be expressed as $C_j(\mathbf{X})$, where $C_j(\mathbf{X})$ are given by

$$C_j(\mathbf{X}) = \begin{cases} P_{m,o}[n]-P_{m,d}[n], j=N+1,\ldots,N+NM; \\ P_{m,o}[n]t_{m,o}[n], j=N+NM+1,\ldots,N+2NM; \\ P_{m,d}[n]t_{m,d}[n] \\ \ -P_{m,o}[n]t_{m,o}[n], j=N+2NM+1,\ldots,N+3NM; \\ t_{m,o}[n], j=N+3NM+1,\ldots,N+4NM; \\ \tau-t_{m,o}[n], j=N+4NM+1,\ldots,N+5NM; \\ t_{m,d}[n], j=N+5NM+1,\ldots,N+6NM; \\ \tau-t_{m,d}[n], j=N+6NM+1,\ldots,N+7NM; \\ P_{m,o}[n], j=N+7NM+1,\ldots,N+8NM; \\ p_{\max}-P_{m,o}[n], j=N+8NM+1,\ldots,N+9NM; \\ P_{m,d}[n], j=N+9NM+1,\ldots,N+10NM; \\ p_{\max}-P_{m,d}[n], j=N+10NM+1,\ldots,N+11NM. \end{cases} \tag{53}$$

Appendix B
Proofs of Lemmas 1 and 2

The result of Lemma 1 directly follows the mathematical definition of an absolute value function. In the following, we mainly prove Lemma 2. According to the given condition, for any $\mathbf{X},\mathbf{d}\in\mathbb{R}^{4MN}$ and $\delta>0$, we can get

$$\frac{\Phi(\mathbf{X}+\delta\mathbf{d})-\Phi(\mathbf{X})}{\delta} = \max_{l\in\mathcal{V}}\left\{\frac{C_l(\mathbf{X}+\delta\mathbf{d})-\Phi(\mathbf{X})}{\delta}\right\}. \tag{54}$$

For the case of $l\notin\mathcal{V}(\mathbf{X})$, it is obviously seen that

$$\lim_{\delta\to 0}\frac{C_l(\mathbf{X}+\delta\mathbf{d})-\Phi(\mathbf{X})}{\delta} = -\infty. \tag{55}$$

For the case of $l\in\mathcal{V}(\mathbf{X})$, we further get

$$\begin{aligned} &\lim_{\delta\to 0}\frac{C_l(\mathbf{X}+\delta\mathbf{d})-\Phi(\mathbf{X})}{\delta} \\ &=\lim_{\delta\to 0}\frac{C_l(\mathbf{X}+\delta\mathbf{d})-C_l(\mathbf{X})}{\delta} \\ &=\nabla_{\mathbf{X}}C_l(\mathbf{X})^{\mathrm{T}}\mathbf{d}. \end{aligned} \tag{56}$$

Combining (54) and (55) in both the cases, Lemma 2 is proven.

References

[1] S. Darbha, S. Konduri, and P. R. Pagilla, “Benefits of v2v communication for autonomous and connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1954–1963, May 2019.

[2] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, “Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, Fall 2017.

[3] Y. Zheng, S. E. Li, K. Li, and W. Ren, “Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1353–1364, May 2018.

[4] S. E. Li, F. Gao, K. Li, L. Wang, K. You, and D. Cao, “Robust longitudinal control of multi-vehicle systems—a distributed h-infinity method,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2779–2788, Sep. 2018.

[5] Y. Li, C. Tang, S. Peeta, and Y. Wang, “Nonlinear consensus-based connected vehicle platoon control incorporating car-following interactions and heterogeneous time delays,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2209–2219, June 2019.

[6] J. Chen, H. Liang, J. Li, and Z. Lv, "Connected automated vehicle platoon control with input saturation and variable time headway strategy," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[7] Y. Li, W. Chen, S. Peeta, and Y. Wang, "Platoon control of connected multi-vehicle systems under v2x communications: Design and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1891–1902, May 2020.

[8] C. Zhai, Y. Liu, and F. Luo, "A switched control strategy of heterogeneous vehicle platoon for multiple objectives with state constraints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1883–1896, May 2019.

[9] S. Thormann, A. Schirrer, and S. Jakubek, "Safe and efficient cooperative platooning," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[10] R. Alieiev, G. Jornod, T. Hehn, A. Kwoczek, and T. Kürner, "Improving the performance of high-density platooning using vehicle sensor-based doppler-compensation algorithms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 421–432, Jan 2020.

[11] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–29, 2019.

[12] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1275–1313, Secondquarter 2019.

[13] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1243–1274, Secondquarter 2019.

[14] Z. Tan, H. Qu, J. Zhao, S. Zhou, and W. Wang, "Uav-aided edge/fog computing in smart iot community for social augmented reality," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4872–4884, June 2020.

[15] B. Li, Z. Fei, and Y. Zhang, "Uav communications for 5g and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, April 2019.

[16] J. Qiu, D. Grace, G. Ding, M. D. Zakaria, and Q. Wu, "Air-ground heterogeneous networks for 5g and beyond via integrating high and low altitude platforms," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 140–148, December 2019.

[17] Y. Zhou, N. Cheng, N. Lu, and X. S. Shen, "Multi-uav-aided networks: Aerial-ground cooperative vehicular networking architecture," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 36–44, Dec 2015.

[18] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, thirdquarter 2019.

[19] E. Turgut, M. C. Gursoy, and I. Guvenc, "Energy harvesting in unmanned aerial vehicle networks with 3d antenna radiation patterns," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2020.

[20] S. Suman, S. Kumar, and S. De, "Uav-assisted rfet: A novel framework for sustainable wsn," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 4, pp. 1117–1131, Dec 2019.

[21] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-ground integrated mobile edge networks: A survey," *IEEE Access*, vol. 8, pp. 125 998–126 018, 2020.

[22] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 140–146, February 2020.

[23] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When uav swarm meets edge-cloud computing: The qos perspective," *IEEE Network*, vol. 33, no. 2, pp. 36–43, March 2019.

[24] M. Li, F. R. Yu, P. Si, R. Yang, Z. Wang, and Y. Zhang., "Uav-assisted data transmission in blockchain-enabled m2m communications with mobile edge computing," *IEEE Network*, pp. 1–8, 2020.

[25] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K. R. Choo, "Neural-blockchain-based ultrareliable caching for edge-enabled uav networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5723–5736, Oct 2019.

[26] G. Faraci, C. Grasso, and G. Schembra, "Design of a 5g network slice extension with mec uavs managed with reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2356–2371, Oct 2020.

[27] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[28] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in mec-and uav-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2020.

[29] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for uav networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, 2020.

[30] N. Zhao, Z. Liu, and Y. Cheng, "Multi-agent deep reinforcement learning for trajectory design and power allocation in multi-uav networks," *IEEE Access*, vol. 8, pp. 139 670–139 679, 2020.

[31] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K. Wong, "Minimum throughput maximization for multi-uav enabled wpcn: A deep reinforcement learning method," *IEEE Access*, vol. 8, pp. 9124–9132, 2020.

[32] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends in Cognitive Sciences*, vol. 23, no. 5, pp. 408 – 422, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661319300610

[33] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient uav-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.

[34] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

[35] J. Zhang, L. Zhou, Q. Tang, E. C. . Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3688–3699, 2019.

[36] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for uav-enabled mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2019.

[37] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[38] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2018.

[39] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.

[40] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[41] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, "Uav-assisted wireless powered cooperative mobile edge computing: Joint offloading, cpu control, and trajectory optimization," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2777–2790, 2020.

[42] 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 Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 187–10 200, 2019.

[43] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[44] M. Treiber, A. Kesting, and D. Helbing, "Delays, inaccuracies and anticipation in microscopic traffic models," *Physica A: Statistical Mechanics and its Applications*, vol. 360, no. 1, pp. 71–88, 2006.

[45] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Springer-Verlag Berlin Heidelberg, 2013. [Online]. Available: https://www.springer.com/gp/book/9783642324598

[46] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in uav-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3147–3159, 2020.

[47] M. Powell, "A fast algorithm for nonlinearly constrained optimization calculations," in *Numerical Analysis*. Springer, 1978, pp. 144–157.

[48] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

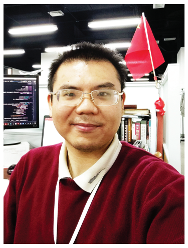
[49] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, p. 1–122, Jan. 2011. [Online]. Available: https://doi.org/10.1561/2200000016

[50] C. A. Floudas and P. M. Pardalos, Eds., *Encyclopedia of Optimization, Second Edition*. Springer, 2009.

[51] R. Dehghannasiri, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Optimal bayesian kalman filtering with prior update," *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 1982–1996, April 2018.

[52] W. Li and L. Guo, "Robust particle filtering with time-varying model uncertainty and inaccurate noise covariance matrix," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2020.

Yang Liu received her B.Sc. degree in traffic engineering from Beijing University of Technology, Beijing, China, in 2019. She is currently working towards her M.Sc. degree in Beihang University. Her research interests are focused on vehicular communications and air-ground cooperative networks, connected autonomous vehicles and intelligent transportation systems.

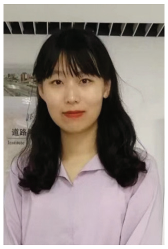


Jianshan Zhou received the B.Sc.,M.Sc., and Ph.D. degrees in traffic information engineering and control from Beihang University, Beijing, China, in 2013, 2016 and 2020, respectively. From 2017 to 2018, he was a Visiting Research Fellow with the School of Informatics and Engineering, University of Sussex, Brighton, U.K. He is currently a Postdoctoral Research Fellow supported by the Zhuoyue Program of Beihang University, and is or was the Technical Program Session Chair with the IEEE EDGE 2020 and the Youth Editorial Board Member of the Unmanned Systems Technology. He is the author or coauthor of more than 20 international scientific publications. His research interests include the modeling and optimization of vehicular communication networks and air–ground cooperative networks, the analysis and control of connected autonomous vehicles, and intelligent transportation systems. He was the recipient of the First Prize in the Science and Technology Award from the China Intelligent Transportation Systems Association in 2017, the First Prize in the Innovation and Development Award from the China Association of Productivity Promotion Centers in 2020, the National Scholarships in 2017 and 2019, the Outstanding Top-Ten Ph.D. Candidate Prize from Beihang University in 2018, and the Outstanding China-SAE Doctoral Dissertation Award in 2020.

Daxin Tian [M'13-SM'16] received his Ph.D. degree in computer application technology from Jilin University, Changchun, China, in 2007. He is currently a University Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His research is focused on intelligent transportation systems, autonomous connected vehicles, swarm intelligence, and mobile computing. He was awarded the Changjiang Scholars Program (Young Scholar) of Ministry of Education of China in 2017, the National Science Fund for Distinguished Young Scholars in 2018, and the Distinguished Young Investigator of China Frontiers of Engineering in 2018. He is also a senior member of the IEEE and served as the Technical Program Committee member/Chair/Co-Chair for several international conferences including EAI 2018, ICTIS 2019, IEEE ICUS 2019, IEEE HMWC 2020, GRAPH-HOC 2020, etc.

Zhengguo Sheng [SM'18] received the B.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006, and the M.S. and Ph.D. degrees from Imperial College London, London, U.K., in 2007 and 2011, respectively. He is currently a Senior Lecturer with the University of Sussex, Brighton, U.K. Previously, he was with UBC, Vancouver, BC, Canada, as a Research Associate and with Orange Labs, Santa Monica, CA, USA, as a Senior Researcher. He has more than 100 publications. His research interests cover IoT, vehicular communications, and cloud/edge computing.

Xuting Duan received the Ph.D. degree in traffic information engineering and control from Beihang University, Beijing, China, in 2017. He is currently an assistant professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. His current research interests are focused on vehicular ad hoc networks.



Guixian Qu received the B.Sc. degree in transportation engineering from Shandong University of Technology, Shandong, China, in 2012, the M.Sc. and Ph.D. degrees from Beijing University of Technology, Beijing, China, in 2014 and 2019, respectively. She is currently a Postdoctoral Research Fellow of Beihang University. Her research interests include the intelligent transportation systems, connected autonomous vehicles, vehicle dynamics modeling and control. She was the recipient of the First Prize in the Science and Technology Award from the China Communications and Transportation Association in 2019.

Victor C. M. Leung [S'75-M'89-SM'97-F'03] is a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, Shenzhen, China, and a Professor Emeritus at the University of British Columbia (UBC), Vancouver, Canada. Before he retired from UBC at the end of 2018, he was a Professor of Electrical and Computer Engineering and holder of the TELUS Mobility Research Chair there. His research is in the broad areas of wireless networks and mobile systems. He has coauthored more than 1300 journal/conference papers and book chapters. Dr. Leung is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, IEEE Network, and several other journals. He received the IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, and 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award. He co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Fellow of IEEE, the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers".