Computer Communications 166 (2021) 244–253

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Energy-efficient computation offloading for vehicular edge computing networks

Xiaohui Gu, Guoan Zhang *

School of Information Science and Technology, Nantong University, Jiangsu, 226019, China

ARTICLE INFO

Keywords:
Vehicular networks
Multi-access edge computing
Computation offloading
Resource allocation
Mobility

ABSTRACT

The demanding computing capacity of emerging vehicular applications has emerged as a challenge in Internet of vehicles (IoVs). Multi-access edge computing (MEC) can significantly enhance computing capability and prolong battery life of vehicles through offloading computation-intensive tasks for edge computing. Considering the impact of vehicles' mobility on communication quality, this paper provides an energy-efficient computation offloading scheme for vehicular edge computing networks (VECN). An energy-efficiency cost (EEC) minimization problem is formulated to make a tradeoff between latency and energy consumption, for completing computational tasks in an effective manner. Since that multiple variables and time-varying channel conditions make the formulated problem difficult to solve, we transform the original non-convex problem into a two-level optimization problem and develop an iterative distributed algorithm to obtain an optimal solution. Numerical results verify the convergence and superiority of the proposed algorithm.

1. Introduction

With the development of technologies such as autonomous driving and artificial intelligence, there emerge new vehicular applications and services, accelerating the growth of computing data. However, the limited computing capacity of on-board computing units has become an inescapable fact, which facilitates the development of cloud-based vehicular networking technology. Although the cloud enjoys rich computational and storage resources, the inherent limitations of cloud computing may introduce delay fluctuation and invoke extra transmission energy cost. Thus, the efficiency of computation offloading will be degraded.

To resolve this issue, multi-access computing (MEC), pushing cloud services to wireless access networks, has been proposed as an iterative solution. MEC servers equipped at base stations (BSs) [1], in close proximity to vehicular users, can provide low latency, high bandwidth and computing agility in the computation offloading process. In vehicular edge computing networks (VECN), vehicles moving on the road can access computational and storage resources of MEC servers through vehicular-to-infrastructure (V2I) communication links, as shown in Fig. 1. Through offloading computing-demanding tasks to MEC servers, vehicular users can get faster interactive response and consume less energy.

However, the computation offloading is a very complex process [2], affected by different factors, such as radio and backhaul connection quality, users' preferences, users' capacity, or cloud capacity and availability. As a result, it is vital to decide how much and what should be offloaded, how to efficiently allocate communication and computational resources, and the impact of mobile users' mobility, when we design the offloading strategy in VECN to provide good QoS for vehicular users.

Basically, a decision on computation offloading may result in binary offloading and partial offloading, which is closely related to application model/type. Since it determines whether full or partial offloading is applicable, what could be offloaded, and how to offload [3]. A highly integrated or relatively simple task cannot be partitioned and has to be executed as a whole either locally at mobile device or offloaded to MEC server, called binary offloading. Applications/programs, composed of multiple procedures/components, can be partitioned into two parts with one executed at mobile device and other offloaded for edge execution, called partial offloading. Compared with binary offloading, partial offloading is benefit in parallel computing and lower latency. However, partial offloading is more complicated and affected by different factors, i.e. whether applications can be divided into offloadable parts or not, the offloadable part may differ in the amount of data and required computational resources, how to decide which part could be offloaded to MEC. Moreover, parallel offloading may not be applicable when the components of applications need computational results from some previous stages. For example, in the deep neural network model, the inference happens after training [4], thus parallel of local execution and MEC execution is not suitable. Therefore, we investigate the energy-efficient computation offloading under binary offloading model.

* Corresponding author.
E-mail addresses: 17110013@yjs.ntu.edu.cn (X. Gu), gzhang@ntu.edu.cn (G. Zhang).

https://doi.org/10.1016/j.comcom.2020.12.010
Received 28 July 2020; Received in revised form 2 October 2020; Accepted 2 December 2020
Available online 13 December 2020

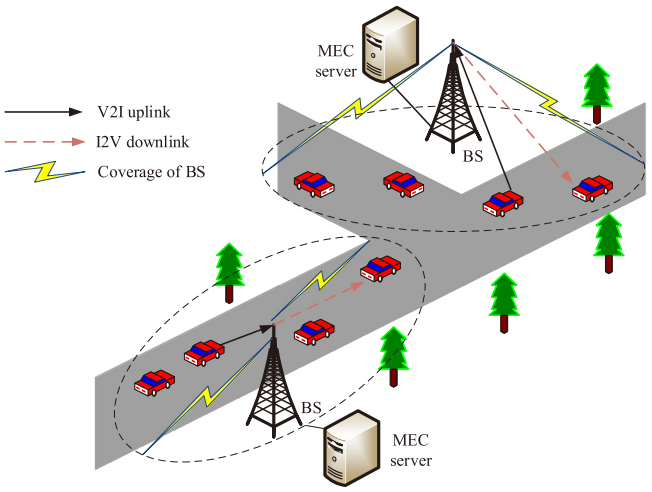


Fig. 1. System model of vehicular edge computing networks.

In this paper, we first study the relationship between the mobility of vehicles and time-varying channel conditions. The tradeoff between latency and energy consumption for completing a task is considered, before the offloading decision and resource allocation are determined. Then, we formulate an optimization problem to minimize the weighted sum of latency and energy consumption (*energy-efficiency cost (EEC)*), by joint optimizing the offloading decision, transmit power, offloading time and CPU frequency in a vehicle. Our main contribution is to conceive an energy-efficient computation offloading scheme for VECN, detailed as follows:

- With considering the impact of time-varying channel, the weighted sum of latency and energy consumption (referred to as *energy efficiency cost*) of vehicular users is minimized by jointly optimizing the transmit power, uploading time, as well as offloading decision and local CPU frequency. The tradeoff between latency and energy consumption is achieved by adjusting weight factors and enabling vehicular users to flexibly vary the supply voltage and clock frequency via dynamic voltage and frequency scaling (DVFS) technology.
- To solve the formulated non-convex problem, the original problem is transformed into an equivalent form. Then, we decompose it into a two-level optimization problem, and propose an iterative distributed algorithm to solve it. The goal of the lower-level problem is to determine the offloading decisions, transmit power, and local CPU frequency given the offloading time, while in the higher-level problem, the offloading time is optimized.
- Specifically, we derive the optimal solution in a semi-closed forms for the lower-level problem by leveraging Lagrange duality method. The structure for the optimal decision shows that whether vehicular users choose to locally compute or offload their tasks for remote computing depends on not only the specification of vehicular users such as clock frequency and transmission power, but also the latency requirements of tasks and the offloading data rate. To solve the high-level problem, one-dimensional line search method is used.
- The numerical results show that the computation performance obtained by using the proposed joint optimization scheme is better than these achieved by using disjoint optimization schemes. Moreover, it only takes several iterations for the proposed iterative algorithm to converge. Additionally, how the system parameters affect the computation offloading scheme is discussed incorporating with simulation curves.

The remainder of this paper is organized as follows. Section 2 presents related works. Section 3 describes the system model of computation offloading in VECN, and the EEC minimization problem is formulated. Then, by solving the formulated problem, the computation offloading and resources allocation algorithm is developed in Section 4. Afterwards, numerical results are presented to confirm the effectiveness of the designed algorithm in Section 5. Finally, Section 6 concludes this paper.

2. Related work

The computation offloading scheme in MEC networks has been a hot spot in recent years, specifically focusing on enhancing the system performance gain, such as reducing latency, energy consumption or system cost by optimizing offloading decision and allocating resources effectively. The *task completion time minimization* was studied in [5–7]. Specially, Hong et al. [5] optimized the resource allocation for time division multiple access (TDMA) and frequency division multiple access (FDMA) systems, respectively, Jinke et al. [6] obtained the minimum latency for both local and edge computing models in the TDMA system, and Wu et al. [7] proposed a nonorthogonal multiple access (NOMA) enabled computation offloading scheme considering the advantage of NOMA. On the aspect of *energy consumption minimization*, You et al. [8] studied the optimal resource-management for the cases of identical and arbitrary arrival-deadline orders, respectively, and Sheng et al. [9] minimized the weighted sum of terminal energy consumption via dynamically matching individual offloading behavior and group's competitive resources allocation. Lyu et al. [10] proposed a heuristic offloading decision algorithm (HODA)by jointly optimizing the offloading decision, and communication and computation resources to *maximize system utility*. Bi and Zhang [11] designed an offloading scheme for the *computation rate maximization* problem in a multi-user MEC network powered by the wireless power transfer. Moreover, some works [12–14] have considered the *tradeoff between the execution delay and energy consumption* at mobile devices by jointly optimizing the computation offloading strategy and resource allocation. More recently, the energy-delay tradeoff was achieved in [15] by self-organization of mobile devices via DVFS technology, and the cooperative offloading scheme was developed in MEC enabled FiWi enhanced HetNets. However, most of the mentioned computational offloading policies are investigated in the scenario where mobile devices are fixed and the offloading rate is constant during the offloading duration.

Intelligent vehicles are regarded as one of the important platforms to implement computing applications. The computation offloading under VECN has gained widespread popularity recently. With an emphasis on the *layered architecture*, a two-layer architecture composed of the vehicles layer and the roadside BSs layer was proposed in [1], a collaborative MEC framework for vehicular networks was proposed in [16]. To improve the *network management*, the authors in [17] recommended a distributed reputation management system to provide security protection and enhance network efficiency in the implementation of vehicular edge computing. To ensure the best use of underutilized vehicular computational resources, Feng et al. [18] proposed an autonomous vehicular edge (AVE) framework in managing idle computing resources of vehicles. Aiming to optimize the *computation offloading scheme*, [19] and [20] proposed a cloud-assisted computation offloading framework for vehicular networks, [21] proposed integrating load balancing with offloading and studied resource allocation for a multi-user multi-server vehicular edging computing system. Liu et al. [22] studied the task offloading problem from a matching perspective and proposed pricing based matching algorithms to optimize the total network delay. In [23], the authors studied the V2V computation offloading for MEC based on the software-defined networks, and specifically studied the communication topology relationship between vehicles for computation offloading. In [24], the concept of collaborative vehicular multi-access edge networks was proposed, which can reduce the perceived response

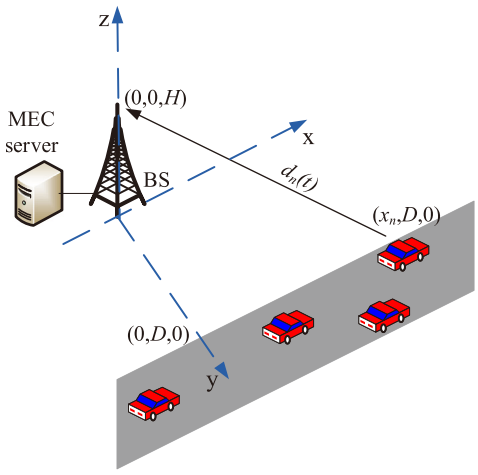


Fig. 2. The vehicular edge computing network framework.

time and improve driving safety and convenience for immersive vehicle applications. In [25], the collaboration between cloud and MEC was investigated to maximize the system utility. In [26], the authors proposed a multi-user noncooperative computation offloading game accompanied by the designed payoff function, in which multi-vehicle competition for MEC resources. Although these works neglected the delay jitter caused by the fast moving speed of vehicles, they have provided insightful understanding and valuable directions for the resource management in VECN.

In the regard of *vehicles' mobility and hard latency*, the authors in [27–29] designed the computation offloading and resource allocation policy of joint communication, caching and computing for VECN; however, they made an assumption that the offloading rate is only related to the initial location of vehicles and does not change during the computation offloading duration. In fact, these computation offloading policies may fall short of accuracy for scenarios, where multiple vehicles moving at high speed in the communication range of BS.

Different from these studies, in this paper, we concentrate on the mobility of vehicles and time-varying channel conditions. An optimal offloading scheme is proposed in this paper, to improve the energy-efficiency of VECN while guaranteeing the latency constraints of delay-sensitive and delay-tolerant computational tasks. Moreover, the DVFS technology is adopted to achieve a suitable energy-latency tradeoff on the vehicle.

3. System model and problem formulation

In this section, the framework of VECN is introduced firstly. Then, the task model and computing model are presented separately. Finally, the optimization problem is formulated.

3.1. Framework of VECN

For general presentation, we adopt a three-dimensional Euclidean coordinate as shown in Fig. 2. The height of BS antenna is denoted as H, and the distance from BS to highway is D. Thus, the location of BS is $(0, 0, H)$, the initial location of vehicle n moving along the highway at speed v_n is $(x_n, D, 0)$. The instant distance $d_n(t)$ from vehicle n to BS is shown as

$$d_n(t) = \sqrt{H^2 + D^2 + \left(x_n + v_n t\right)^2}, \tag{1}$$

With the development of intelligent transportation system (ITS), more roadside units (or BSs) will be built on the roadside to meet vehicular users' demands [29]. Note that we assume vehicular users in the VECN use TDMA protocol to offload computational bits, with no interference between each other during the offloading process. Hence, the distance between the vehicle and BS is the main influence factor for the transmission performance. As in [29], we adopt a simple channel model, and a more complicated channel model will be studied in our future work. The channel $g_n(t)$ from vehicle n to the BS is defined as

$$g_n(t) = \beta_0 d_n(t)^{-\theta} = \frac{\beta_0}{\left[H^2 + D^2 + \left(x_n + v_n t\right)^2\right]^{\frac{\theta}{2}}}, \tag{2}$$

where β_0 denotes the channel power gain at the reference distance $d_0 =$ 1m and θ denotes the path-loss exponent for V2I links. The bandwidth of different offloading channels is the same, which is denoted as W. Let $p_n > 0$ represent the transmit power of vehicle n. Then, the instant transmission rate $r_n(t)$ (in bits/second (bps)) from the vehicle to BS is given by

$$r_n(t) = W \log_2\left(1 + \frac{p_n g_n(t)}{\sigma^2}\right), \tag{3}$$

$$= W \log_2\left(1 + \frac{p_n \rho_0}{\left[H^2 + D^2 + \left(x_n + v_n t\right)^2\right]^{\frac{\theta}{2}}}\right). \tag{4}$$

where $\rho_0 = \frac{\beta_0}{\sigma^2}$ and σ^2 is the noise power at BS receiver. Since in this paper the main concern of communication performance is adapting to the mobility of vehicles and adjusting the transmit power, which means the bandwidth of different offloading channels is the same.

3.2. Task model

The set of vehicles is defined as $N = \{1, 2, \ldots, n\}$, each of which has a computation-intensive and delay-sensitive task R_n to be completed, defined as $R_n = \left(L_n, C_n, T_{n,max}\right)$. Here, L_n (in bits) denotes the input data size of task R_n, and C_n (in CPU cycles/bit) denotes the computational complexity/intensity of task R_n. As the after mentioned references about computation offloading schemes [5,6], C_n is assumed to be a fixed value whether the task is processed at the vehicle or MEC server. Let $T_{n,max}$ (in s) denote the maximum tolerable latency of task R_n, which means the time for completing task R_n should smaller than $T_{n,max}$. These parameters are related to the nature of the applications and can be estimated through task profilers [23]. The computation offloading decision of task R_n is denoted as $a_n \in \{0, 1\}$. Specifically, $a_n = 0$ indicates that vehicle n chooses to compute its computational task on its own on-board units, while $a_n = 1$ implies that vehicle n chooses to offload its computational task for edge computing. Furthermore, let $A = \left\{a_n | n = 1, \ldots, N\right\}$ denote the decision profile of all computational tasks in the system.

3.3. Computing model

3.3.1. Local computing

If vehicle n decides to compute task R_n locally, the local computing time of vehicle n is denoted by T_n^l. Then, the computing capacity of vehicle n is denoted by f_n^l (CPU cycles per second). The latency of local computing T_n^l for task R_n is given by

$$T_n^l = \frac{L_n C_n}{f_n^l}, \tag{5}$$

As in [13], the energy consumption of local computing E_n^l is given by

$$E_n^l = k T_n^l \left(f_n^l\right)^3 = k L_n C_n \left(f_n^l\right)^2, \tag{6}$$

where k is the effective switched capacitance depending on the chip architecture. It is clear that we can adjust the clock frequency of CPU chip to achieve optimal computation time and energy consumption on the vehicle, by leveraging the dynamic voltage and frequency scaling (DVFS) technology [13]. The DVFS technique enables the CPU core to

operate at multiple different frequency levels. Then, the EEC of local computing Z_n^l is given by

$$Z_n^l = \beta_n^T T_n^l + \beta_n^E E_n^l. \tag{7}$$

where $0 \le \beta_n^T \le 1$ and $0 \le \beta_n^E \le 1$ denote the weights of task completion time and energy consumption for vehicle n making decision on task R_n, respectively.

Note that the ECC is defined as a linear combination of energy consumption and computation completion time, to coincidentally reflect the energy-efficiency cost of executing a task, i.e., both higher energy consumption and longer computation completion time lead to higher energy-efficiency cost. To meet user specific demands, we allow different vehicular users to choose different weights in decision making. For example, a device with low battery energy would like to choose a larger β_n^E to save more energy. When a mobile user is running some delay sensitive applications (e.g., online movies), it may prefer to set a larger β_n^T to reduce the delay.

3.3.2. Multi-access edge computing

When vehicle n decides to offload its computational task R_n to MEC server for execution, this process includes three consecutive phases: transmitting phase, computing phase, and receiving phase. Since the offloading rate is time-varying, the relationship between transmission time t_n^{ot} and instant transmission rate $r_n(t)$ is given by

$$\int_0^{t_n^{ot}} r_n(\tau) d\tau = L_n, \tag{8}$$

The execution delay at MEC server t_n^{oe} is given by

$$t_n^{oe} = \frac{L_n C_n}{f_{MEC}}, \tag{9}$$

where f_{MEC} is the computing capacity of MEC server (CPU cycles per second), which is assumed to be a constant. In this paper, the MEC server is assumed to enjoy abundant computing resources.

For the facts [12,13] that the BS generally has large transmission power, the download rate is generally high, and the data size of the results is much smaller than of input data size, so we ignore the download delay in the rest of the paper. Accordingly, the latency of edge computing T_n^o is given by

$$T_n^o = t_n^{ot} + t_n^{oe}, \tag{10}$$

The energy consumption of vehicle n uploading its computational task is given by

$$E_n^o = p_n t_n^{ot}, \tag{11}$$

Therefore, the EEC of MEC is given by

$$Z_n^o = \beta_n^T T_n^o + \beta_n^E E_n^o. \tag{12}$$

Finally, the EEC spent to complete task R_n is expressed as

$$\begin{aligned} Z_n &= \left(1 - a_n\right) Z_n^l + a_n Z_n^o, \qquad (13)\\ &= \beta_n^T \left[\left(1 - a_n\right) \frac{L_n C_n}{f_n^l} + a_n \left(\frac{L_n C_n}{f_{MEC}} + t_n^{ot}\right)\right] \\ &\quad + \beta_n^E \left[\left(1 - a_n\right) k L_n C_n \left(f_n^l\right)^2 + a_n p_n t_n^{ot}\right]. \qquad (14) \end{aligned}$$

where a_n denotes the offloading decision, specifically $a_n = 1$ indicates the task is offloaded for remote computing and $a_n = 0$ denotes the task is locally computed. The left part of Eq. (14) is the product of latency weight factor and task completion time consumed by either local computing or remote processing, and the right part is the product of energy weight factor and energy consumption for completing the task by either local computing or remote computing.

3.4. Problem formulation

In this subsection, the offloading decisions and resources allocation is formulated as an optimization problem, with the objective function of minimizing the total EEC of users in the system as well as latency and computing capacity constraints. The optimization problem is expressed as

$$\begin{aligned} &\min_{A,F,P,T} \sum_{n=1}^{N} Z_n \\ &s.t. C1: \ a_n \in \{0,1\}, \\ &\quad C2: \ \left(1 - a_n\right) T_n^l + a_n T_n^o \le T_{n,max}, \\ &\quad C3: \ a_n L_n \le \varphi\left(p_n, t_n^{ot}\right), \\ &\quad C4: \ a_n \sqrt{D^2 + \left(x_n + \upsilon_n T_n^o\right)^2} \le R_{max}, \\ &\quad C5: \ 0 \le f_n^l \le f_{n,max}^l, \\ &\quad C6: \ 0 \le p_n \le p_{n,max}. \end{aligned} \tag{15}$$

where $A = \{a_n | n \in N\}$, $F = \{f_n^l | n \in N\}$, $P = \{p_n | n \in N\}$, $T = \{t_n^{ot} | n \in N\}$, $\varphi\left(p_n, t_n^{ot}\right) = \int_0^{t_n^{ot}} r_n(\tau) d\tau$ and R_{max} is the maximum communication range of the BS. In Eq. (15), the objective function can make a tradeoff between the latency and energy consumption by dynamically adjust weighting parameters. When a vehicular user is at a low battery state, the user may set $\beta_n^T = 0$ and $\beta_n^E = 1$ to prolong the battery life; when a vehicular user is running the application that is sensitive to latency (i.e., video streaming), the user may set $\beta_n^T = 1$ and $\beta_n^E = 0$ to minimize the latency. In this model, we can adjust weight factors $\beta_n^T, \beta_n^E \in (0,1)$ flexibly; such that a vehicular user can take care of both latency and energy consumption when designing the offloading policy.

Constraint C1 means that the computational task of vehicle n is either executed by local computing or the MEC server; C2 ensures the task completion time not violate latency constraint; C3 denotes the task is successfully offloaded to the MEC server within the offloading time duration; C4 accounts for the mobility of vehicles, which indicates that the task should be successfully offloaded to MEC before the vehicle leaves the coverage of the connected BS, to ensure V2I links not broken during the offloading duration; C5 and C6 bound the feasible value of local computing speed and transmit power of vehicle n.

4. Problem decomposition and solution

In order to facilitate the optimal solution, we simplify the constraint function of C4 as

$$a_n T_n^o \le c_n, \tag{16}$$

and

$$c_n \triangleq \frac{\sqrt{R_{max}^2 - D^2} - x_n}{\upsilon_n}. \tag{17}$$

Hence, the original problem in Eq. (15) is rewritten as

$$\begin{aligned} &\min_{A,F,P,T} \sum_{n=1}^{N} Z_n \\ &s.t. \ C7: \ a_n T_n^o \le c_n, \\ &\quad C2, C3, C5, C6. \end{aligned} \tag{18}$$

4.1. Problem decomposition

Since the multiple variables of problem in Eq. (18) are closely coupled with each other, the optimization problem (18) is non-convex, and difficult to solve directly.

Assertion 1. *Problem in Eq. (18) can be transformed into a two-level optimization problem, by iteratively finding the optimal A^*, F^*, P^* and T^*.*

Proof. Since $\min_{A,F,P,T} \sum_{n=1}^{N} Z_n$ is equivalent to $\min_T \min_{A,F,P} \sum_{n=1}^{N} Z_n$, one can first find A^*, F^*, P^* for given T, and then substitute A^*, F^*, P^* into Eq. (18) to obtain T^*. Therefore, in the higher-level problem, A, F and P are jointly optimized for given T. Subsequently, T is optimized in the lower-level problem, by substituting A^*, F^* and P^* into Eq. (18). That is, the optimal offloading decisions, local computing frequency and transmit power are obtained under given offloading time, and then the optimal offloading time is identified under the obtained offloading decisions, transmit power and local CPU frequency. This process is iterated until convergence.

4.2. A two-level solution approach

4.2.1. The higher-level problem

Problem in Eq. (18) contains the integer variable $a_n \in \{0,1\}$ and linear variables f_n^l, p_n, t_n^{ot}, as well as the non-convex objective function, making Eq. (18) a non-convex and NP-hard problem. As in [13,21,27], we can relax the binary offloading decision variable as $0 \le a_n \le 1$. Then, for given t_n^{ot}, the resulting sub-problem is given by

$$\begin{aligned} &\min_{A,F,P} \beta_n^T \left[(1-a_n) \frac{L_n C_n}{f_n^l} + a_n \left(\frac{L_n C_n}{f_{MEC}} + t_n^{ot} \right) \right] \\ &\quad + \beta_n^E \left[(1-a_n) k L_n C_n \left(f_n^l\right)^2 + a_n p_n t_n^{ot} \right] \\ &s.t.\ C8:\ 0 \le a_n \le 1, \\ &\quad C2, C3, C5, C6, C7. \end{aligned} \tag{19}$$

Lemma 1. *The optimization problem (19) is convex.*

Proof. Please see Appendix A.

Lemma 1 reveals that the convex optimization problem in Eq. (19) satisfies the Slater's constraint qualification and its duality gap is zero. The result of zero-duality-gap yields a method to solve Eq. (19) by solving its corresponding duality problem [30].

(1) Duality problem formulation

Let $\lambda = \left[\lambda_n, n = 1..., N\right]^T$, $\chi = \left[\chi_n, n = 1, \ldots, N\right]^T$ denote Lagrange multipliers associated with constraints C2 and C3, respectively. The dual function is given by

$$\begin{aligned} &L(\lambda, \chi, \omega, A, F, P) \\ &\quad = (1-a_n) Z_n^l + a_n Z_n^o \\ &\quad\quad + \lambda_n \left[(1-a_n) \frac{L_n C_n}{f_n^l} + a_n \left(\frac{L_n C_n}{f_{MEC}} + t_n^{ot} \right) - T_{n,max} \right] \\ &\quad\quad + \chi_n \left[a_n L_n - \varphi\left(p_n, t_n^{ot}\right) \right]. \end{aligned} \tag{20}$$

Then the duality problem is denoted as

$$\max_{\lambda,\chi>0} \min_{A,F,P} L(\lambda, \chi, \omega, A, F, P) \tag{21}$$

Based on the Layering as Optimization Decomposition (LOD) method [30,31], the duality problem in Eq. (21) is broken down into two layers: the inner layer is the minimization problem in Eq. (21) which can be solved in a distributed manner; the outer layer is the maximization problem in Eq. (21).

Lemma 2. *There exists an optimal solution $\left(a_n^*, f_n^{l*}, p_n^*, \lambda^*, \chi^*\right)$ of the Distributed algorithm proposed by LOD approach.*

Proof. It is known from Lemma 1 that the lower-level optimization problem in Eq. (19) is convex and there is a zero duality gap between Eqs. (19) and (21). In addition, we can observe from Eqs. (19) and (21) that the optimization variables a_n, f_n^l and p_n are independent of each other, which implies that the dual problem in Eq. (21) is separable. The authors in [31] have proved that the decomposed subproblems by applying the Layering as Optimization Decomposition (LOD) approach have an optimal solution if and only if there is no duality gap between the primal problem and its corresponding dual problem. Thus, the Distributed algorithm proposed by LOD approach can yield an optimal solution $\left(a_n^*, f_n^{l*}, p_n^*, \lambda^*, \chi^*\right)$.

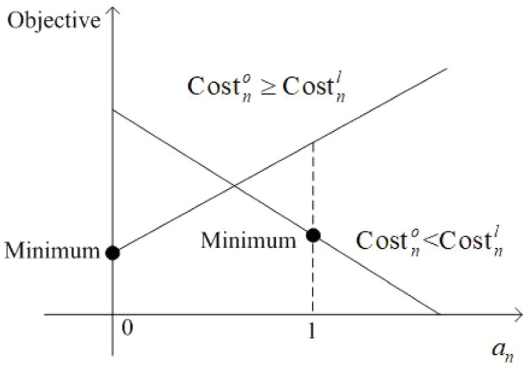


Fig. 3. The value of objective function versus the offloading decision.

In the following, we present distributed sub-algorithms to obtain an optimal computation offloading decision, local CPU frequency and transmit power.

(2) Computation offloading decision

Computation offloading decision subalgorithm aims to determine the task of vehicle n is offloaded or locally processed such that the energy efficiency cost of completing the task is minimized, while the latency requirement is preserved. Let $\text{Cost}_n^l = Z_n^l + \lambda_n \frac{L_n C_n}{f_n^l}$ and $\text{Cost}_n^o = Z_n^o + \lambda_n \left(t_n^{ot} + \frac{L_n C_n}{f_{MEC}} \right)$ respectively denote the computational cost on the vehicle and MEC server. By solving the minimization problem below, the optimal computation offloading decision can be determined

$$\min_{a_n} \text{Cost}_n^l + a_n \left(\text{Cost}_n^o - \text{Cost}_n^l \right) \tag{22}$$

Lagrangian multiplier $\lambda = \left[\lambda_n, n = 1, \ldots, N\right]^T$ in Eq. (20) is for the completion time constraint C2, where λ_n denotes the price of the completion time of task n no more than the required maximum completion time. Specifically, the multiplier is penalty factor on the objective function to make it evolve toward its optimum under the corresponding constraint.

We observe that if $\text{Cost}_n^0 \ge \text{Cost}_n^l$, the linear objective function is minimized when $a_n \in [0,1]$ reaches the minimum; whereas, if $\text{Cost}_n^0 < \text{Cost}_n^l$, the objective function achieves minimum when $a_n \in [0,1]$ reaches the maximum. The analysis is illustrated in Fig. 3. Therefore, the computation offloading decision is summarized as follows

$$a_n^* = \begin{cases} 1, \text{if } \text{Cost}_n^0 < \text{Cost}_n^l, \\ 0, \text{Otherwise}. \end{cases} \tag{23}$$

This indicates that it is beneficial that task R_n is executed on the platform (the on-board unit or MEC server) whose computational cost is smaller. From the definitions of Cost_n^l and Cost_n^0, we can make the conclusion that not only the local CPU frequency f_n^l and transmit power p_n, but also the offloading time t_n^{ot} and latency deadline $T_{n,max}$ affect the computation offloading decision.

(3) Local CPU frequency

The control of clock frequency aims to set the optimal CPU frequency for vehicle n to minimize the EEC of local computing ($a_n = 0$). By solving the optimization problem below, the optimal clock frequency control can be determined

$$\begin{aligned} &\min_{f_n^l} \left(\beta_n^T + \lambda_n\right) \frac{L_n C_n}{f_n^l} + \beta_n^E k L_n C_n \left(f_n^l\right)^2 \\ &s.t.\ C5. \end{aligned} \tag{24}$$

Due to the convexity of the objective function, the optimization problem (24) is convex in $0 \le f_n^l \le f_{n,max}^l$. The KKT conditions [30] are applied to obtain the optimal CPU frequency. Let the first derivative of

the objective function of Eq. (24) equal 0, the optimal CPU frequency is given by

$$f_n^{l*} = \begin{cases} \sqrt[3]{\frac{\beta_n^T + \lambda_n}{2\beta_n^E k}}, & \text{if } \sqrt[3]{\frac{\beta_n^T + \lambda_n}{2\beta_n^E k}} \le f_{n,max}^l, \\ f_{n,max}^l, & \text{Otherwise.} \end{cases} \tag{25}$$

We can make the observation that the local CPU frequency increases monotonically with the weight of latency β_n^T and the price for meeting latency deadline λ_n, and decreases with the weight of energy consumption β_n^E.

(4) Transmit power

The objective of transmit power allocation is to minimize the EEC of MEC by optimally allocating the transmit power for vehicle n. Clearly, this analysis makes sense only when the task is offloaded, *i.e.*, $a_n = 1$. By solving the minimization problem below, the optimal transmit power can be determined

$$\begin{aligned} &\min_{p_n} \beta_n^E p_n t_n^{ot} - \chi_n \varphi\left(p_n, t_n^{ot}\right) \\ &s.t.\ C6. \end{aligned} \tag{26}$$

Lemma 3. *The optimal solution of Eq. (26) is given by*

$$p_n^* = \begin{cases} \hat{p}_n, & \text{if } 0 \le \hat{p}_n \le p_{n,max}, \\ p_{n,max}, & \text{if } p_{n,max} < \hat{p}_n. \end{cases} \tag{27}$$

where $\hat{p}_n$ is the root of the equation $\beta_n^E t_n^{ot} - \chi_n \varphi'\left(p_n, t_n^{ot}\right) = 0$.

Proof. Please see Appendix B

Algorithm 1: Distributed algorithm for solving the higher-level problem

require:
ε: an infinitesimal number;
Initialize: L_n, C_n, $T_{n,max}$, $f_{n,max}^l$, f_{MEC}, $p_{n,max}$, β_n^T, β_n^E, $\psi(t)$, λ_n and χ_n, f_n^l, p_n and iteration index $t \leftarrow 1$;
/* *Local CPU frequency* */
$t = 1$;
Compute the clock frequency $f_n^l(t+1)$ by Eq. (25);
Update Lagrangian multiplier $\lambda_n(t+1)$ by Eq. (28);
$t = t + 1$;
Until $|\lambda_n(t+1) - \lambda_n| < \varepsilon$
Compute T_n^l, E_n^l, Z_n^l by Eqs. (5)-(7), respectively;
Calculate $\text{Cost}_n^l = Z_n^l + \omega_n \frac{L_n C_n}{f_n^l}$;
/* *Transmit power* */
$t = 1$;
Compute $\hat{p}_n$ based on the bisection search method;
Compute the transmit power $p_n(t+1)$ by Eq. (27);
Update Lagrangian multipliers $\lambda_n(t+1)$ and $\chi_n(t+1)$ by Eqs. (29) and (30), respectively;
$t = t + 1$;
Until $|\mu_n(t+1) - \mu_n(t)| < \varepsilon$
Compute T_n^o, E_n^o, Z_n^o by Eqs. (10)–(12), respectively;
Compute c_n by Eq. (17)
Calculate $\text{Cost}_n^o = Z_n^o + \omega_n \left(t_n^{ot} + \frac{L_n C_n}{f_{MEC}}\right)$;
/* *Computation offloading decision* */
if $\text{Cost}_n^o < \text{Cost}_n^l$ **then**
$a_n = 1$
else
$a_n = 0$
end if

(5) Lagrangian multipliers update

Based on the sub-gradient method, the outer layer as the master problem in Eq. (21) can be effectively solved. For given A, F, P, the Lagrange multipliers are updated in the following

For local computing:

$$\lambda_n(k+1) = \left\{ \lambda_n(k) + \psi(k) \left[T_{n,max} - \frac{L_n C_n}{f_n^l} \right] \right\}^+. \tag{28}$$

For edge computing:

$$\lambda_n(k+1) = \left\{ \lambda_n(k) + \psi(k) \left[T_{n,max} - \left(\frac{L_n C_n}{f_{MEC}} + t_n^{ot} \right) \right] \right\}^+. \tag{29}$$

$$\chi_n(k+1) = \left\{ \chi_n(k) + \psi(k) \left[\varphi\left(p_n, t_n^{ot}\right) - a_n L_n \right] \right\}^+. \tag{30}$$

where $(x)^+ = \max(x, 0)$, which normalizes Lagrange multipliers to be non-negative, index $k > 0$ is the iteration index and $\psi(k)$ is positive iteration step size. The details of the proposed distributed algorithm are illustrated in Algorithm 1.

The convergence of the proposed iterative algorithm is guaranteed [30] when the iteration step size $\psi(k)$ satisfies $\psi(k) \to 0$, $\sum_{k=1}^{\infty} \psi(k) \to \infty$, and $\sum_{k=1}^{\infty} \psi(k)^2 < \infty$.

4.2.2. The lower-level problem

The objective of the lower-level problem is to minimize $\zeta\left(t_n^{ot}\right)$ by

$$\begin{aligned} &\min_{t_n^{ot}} \zeta\left(t_n^{ot}\right) \\ &s.t.\ 0 \le t_n^{ot} \le \bar{t}_n^{ot}. \end{aligned} \tag{31}$$

where $\bar{t}_n^{ot} = c_n$ is determined by constraint C7 of problem in Eq. (18) and $\zeta\left(t_n^{ot}\right)$ is the value of Eq. (18) after submitting A^*, F^*, P^* onto Eq. (18)

$$\begin{aligned} \zeta\left(t_n^{ot}\right) = &\sum_{n=1}^{N} \beta_n^T \left[\left(1 - a_n^*\right) \frac{L_n C_n}{f_n^{l*}} + a_n^* \left(t_n^{ot} + \frac{L_n C_n}{f_{MEC}} \right) \right] \\ &+ \beta_n^E \left[\left(1 - a_n^*\right) k L_n C_n \left(f_n^{l*}\right) + a_n^* p_n^* t_n^{ot} \right]. \end{aligned} \tag{32}$$

The one-dimensional linear search method is adopted for solving problem in Eq. (31) with only a single-variable t_n^{ot} and $t_n^{ot} \in [0, c_n]$. With enough small step size of the search, the global optimal solution can be obtained. The iterative optimization process for solving the original problem is summarized in Algorithm 2.

Algorithm 2: The two-level iterative optimization algorithm for vehicle n

Initialize: ζ^{temp} =Inf, $t_n^{ot,temp} = 0$, and Δ.
Repeat 1:
for $t_n^{ot} = 0 : \Delta : c_n$ **do**
Repeat 2:
Carry out Algorithm 1;
Calculate $\zeta\left(t_n^{ot}\right)$ by Eq. (32).
end Repeat 2
if $\zeta\left(t_n^{ot}\right) < \zeta^{temp}$ **then**
Set $\zeta^{temp} = \zeta\left(t_n^{ot}\right)$ and $t_n^{ot,temp} = t_n^{ot}$.
end if
end for
Set $t_n^{ot*} = t_n^{ot,temp}$.
end Repeat 1
Output: The optimal solution t_n^{ot*} to Eq. (31) and corresponding $\left(a_n^*, p_n^*\right)$.

To further clarify the decomposition and solution process, a diagram is given in Fig. 4.

4.3. Complexity analysis

The complexity of Algorithm 2 comes from four aspects. The first aspect contains the computation of optimal CPU frequency and the sub-gradient method for updating dual variable associated with constraint C2. The second aspect is the bisection method for obtaining the optimal offloading time and the subgradient method for updating of Lagrange multipliers associated with constraint C2 and C3. Let L_1 and L_2 denote

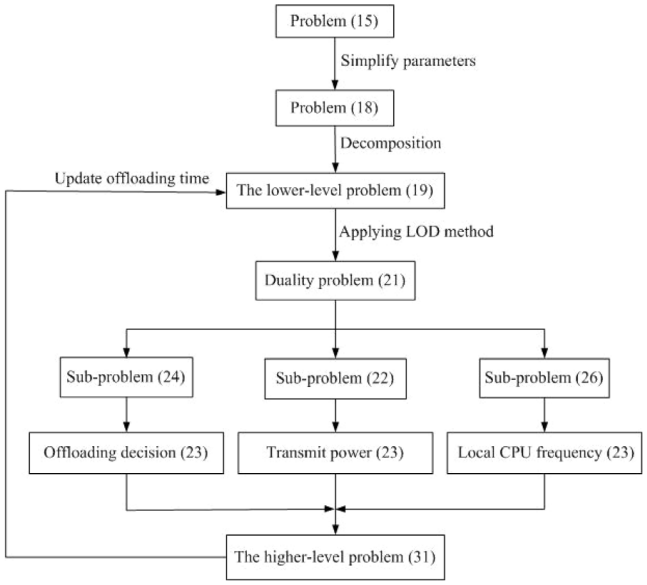


Fig. 4. The flowchart of problem decomposition.

Table 1
Default simulation parameters.

Parameter	Description	Value
N	Number of moving vehicles	10
H	Height of the BS antenna	25 m
D	Distance between BS and highway	35 m
x_n	The initial location of vehicle n	[80, 160] m
θ	Path-loss exponent	4
β_0	Reference channel gain	−30 dB
W	Bandwidth	5 MHz
σ^2	Noise power	−110 dBm
$p_{n,max}$	Maximum transmit power	23 dBm
v_n	Moving speed	100 Km/h
R_{max}	Maximum transmission range of BS	200 m
L_n	Task input data size	1 MB
c_n	Task complexity	[200, 1400] cycles/bit
$f^l_{n,max}$	Maximum computing capacity	10 GHz
f_{MEC}	MEC computing capacity	50 GHz
β^T_n	Weight of latency	0.5
β^E_n	Weight of energy consumption	0.5

the number of iterations required for the outer loop and the inner loop of Algorithm 2, respectively. Let l_1 and l_2 denote the tolerance error for the bisection method and the subgradient method, respectively. Thus, according to the works in [32,33], the total complexity of Algorithm 2 is $O\left[L_1 L_2 \left(\frac{1}{l_2} + \log_2\left(\frac{l_1}{p_{n,max}}\right)\frac{1}{l_2^2}\right)\right]$ and $O(.)$ is the big-O notation [33].

5. Performance evaluation

This section provides simulation results to verify the performance of our proposed computation offloading scheme.

5.1. Experiment setup

In the simulation, the task is assumed to be generated by a face recognition application in [13], which is employed to evaluate the effectiveness of the proposed algorithm. The task profile refers to [34], and the computing profile refers to Ref. [10]. The default simulation settings are listed in Table 1. To obtain the solution in polynomial time, we set $\psi(k) = \frac{1+\beta}{k+\beta}$, where β is a positive constant.

We will first investigate the convergence of the proposed algorithm, then show the impact of various system parameters on the optimal offloading decision and resources allocation, then compare the proposed computation offloading scheme with the following benchmark schemes. To ensure fair comparisons among variants, for the reference schemes we assume that the vehicle is always located at the midpoint between the initial point and allowable maximum location, and each vehicle has only one task to be processed.

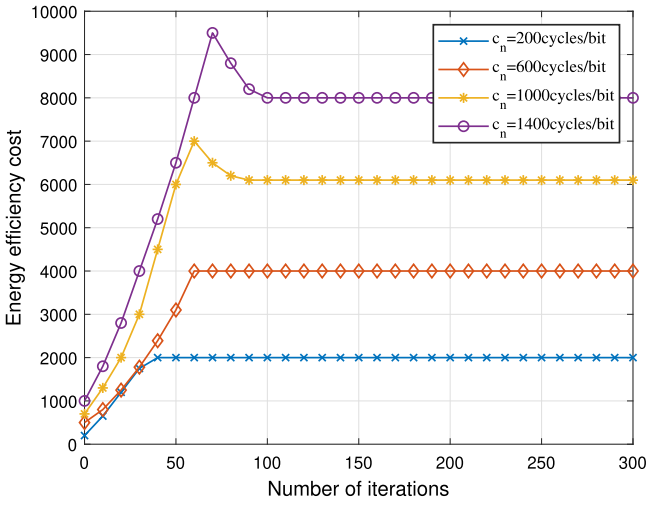


Fig. 5. Convergence curves of the proposed algorithm.

- *ELE with fixed CPU frequency:* Entire local execution without DFVS technology. Here, we set $f^l_n = 0.7 f^l_{n,max}$.
- *ELE with DFVS:* Entire local execution with DFVS technology. The optimal CPU frequency of vehicle n corresponds to Eq. (25).
- *BO with transmit power control:* Binary offloading with transmit power control and fixed local CPU frequency. The optimal transmit power of vehicle n is corresponds to Eq. (27). Here, we set $f^l_n = 0.7 f^l_{n,max}$.
- *SDR-based scheme* [12]: Binary offloading with DFVS and fixed transmit power. The optimal CPU frequency of local computing corresponds to Eq. (25). Here, we set $p_n = p_{n,max}$.

5.2. Convergence

To analyze the convergence of the two-level iterative optimization algorithm, Fig. 5 shows how EEC changes with the number of iterations, when the task complexity is 200, 600, 1000 and 1400cycles/bit, respectively. It is observed that it needs about 40–100 iterations to converge for the proposed algorithm with different task complexity. An iteration takes 0.2–0.5 ms, therefore, our algorithm takes at most 50 ms to obtain the optimal offloading scheme for the most complicated tasks. The results show that the proposed algorithm has good convergence properties in various situations.

5.3. Impact of task input data size

In the cases where the computing capacity of MEC server is 100 GHz, 50 GHz, 40 GHz, 30 GHz and 20 GHz, respectively, we present the percentage of tasks offloaded to MEC with respect to an increasing size of input data in Fig. 6. As shown in Fig. 6, the percentage of offloaded tasks grows as L_n. This is because when the task input data size becomes large, there will be more vehicles tending to offload their computational tasks to MEC server, considering the EEC of MEC and local computing. Moreover, Fig. 6 also indicates that the server computing time plays an essential role in the optimal offloading policy, especially in the case where multiple computation-intensive tasks are offloaded to MEC servers with limited computing capacity.

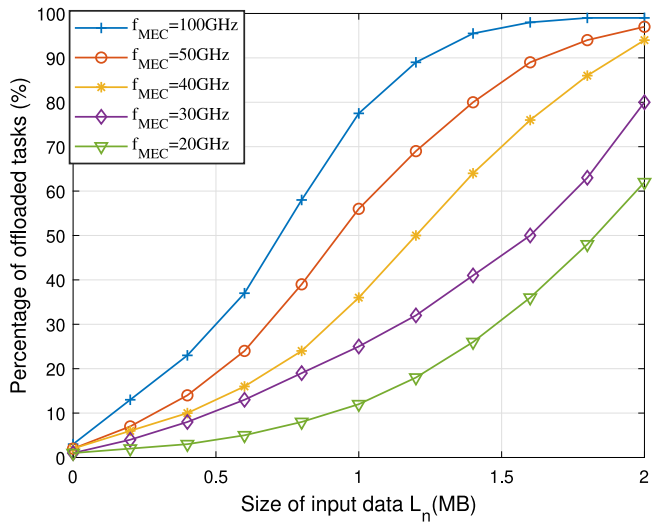


Fig. 6. The percentage of offloaded tasks versus the size of input data.

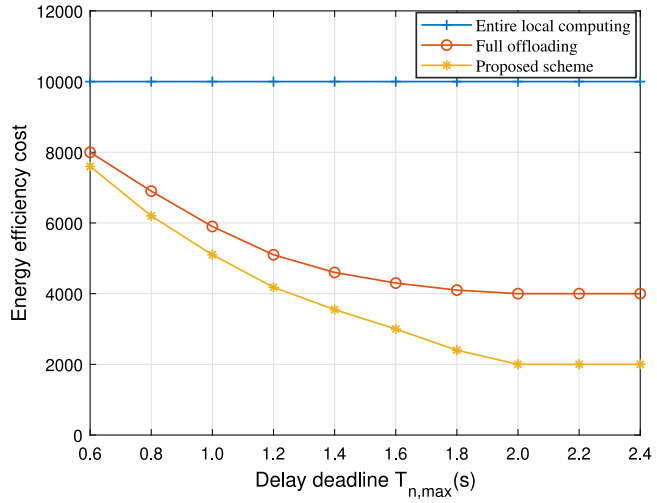


Fig. 7. Energy-efficiency cost versus the delay deadline.

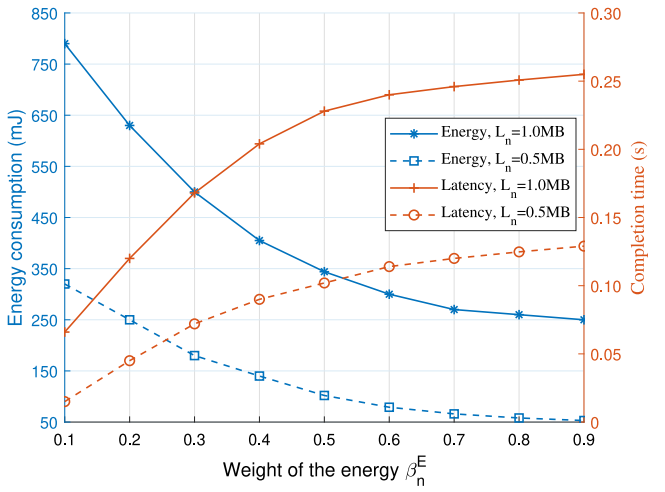


Fig. 8. Performance of the proposed scheme versus weight factor β_n^E in terms of energy consumption and task completion time.

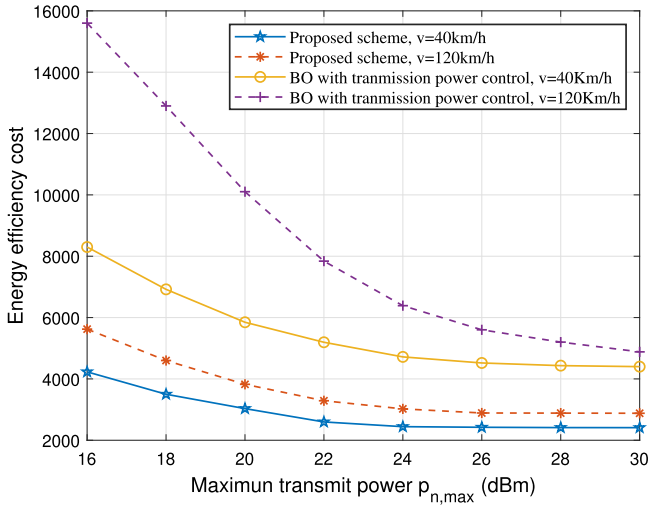


Fig. 9. Performance of the proposed scheme versus the weight factor β_n^E in terms of energy consumption and latency.

5.4. Impact of delay deadline

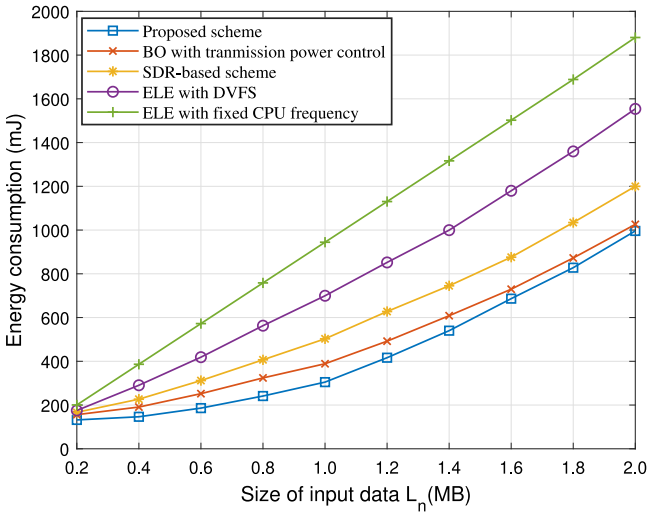
Fig. 7 depicts that different delay deadlines $T_{n,max}$ affect the EEC of the proposed scheme, entire local computing scheme and full offloading scheme [35] for same task profile. The following observations are drawn from Fig. 7. First, only the task data size, task complexity and vehicle's computing capability affect the EEC of the entire local computing scheme. Therefore, the changes in delay deadline do not influence the EEC of the entire local computing scheme. Second, compared with local computing, the proposed scheme reduces EEC significantly. This is because the proposed scheme optimally offloads computation-intensive tasks for remote execution, according to the EEC on MEC and vehicle. Third, the proposed scheme has a lower EEC for a long delay deadline compared with full offloading scheme. This is reasonable given that the proposed scheme adopts the optimal offloading decision and transmit power allocation. Finally, when the maximum tolerable latency is larger than the time spent by the vehicle leaving its connecting RSU, *i.e.*, $T_n^o > c_n$, the maximum tolerable latency of offloading task R_n is replaced by c_n and the EEC of full offloading scheme as well as the proposed scheme will not change with increasing $T_{n,max}$. This process guarantees smooth data transmission in the MEC processing model.

5.5. Impact of weight factors

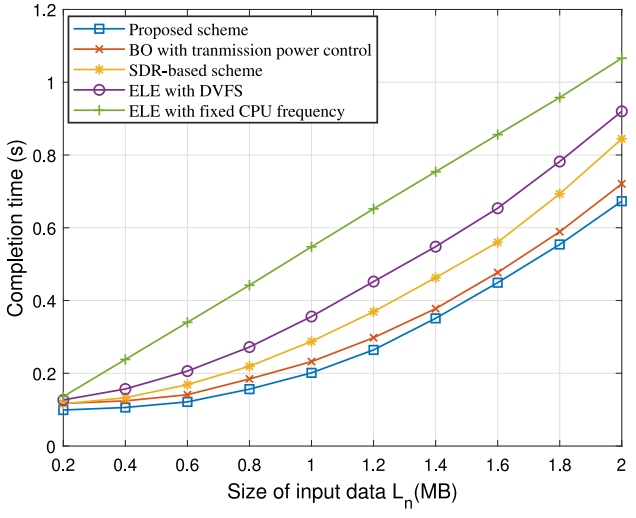
Fig. 8 plots the energy consumption and latency when the weight factor of energy β_n^E increases from 0.1 to 0.9, while the weight factor of latency $\beta_n^T = 1 - \beta_n^E$ decreases from 0.9 to 0.1. We can observe that the energy consumption reduces when β_n^E increases, at the cost of larger latency. That is, the lower the energy consumption is, the greater the latency. It just shows the tradeoff between latency and energy consumption for completing the computational task.

5.6. Impact of maximum transmit power

The impact of the maximum transmit power on the performance of two schemes, *i.e.*, the proposed scheme and the binary offloading scheme (with transmit power control and fixed local CPU frequency) is discussed in Fig. 9. We can observe that the EEC decreases as $p_{n,max}$ increases. Besides, when $p_{n,max}$ is large enough, the proposed scheme will approach EEC saturation under different speed v_n, due to the fact that *(1)* the increase in the maximum transmit power makes more vehicles offload their computational tasks to the MEC server; *(2)* as the transmit power keeps growing, the EEC of MEC is closer to that of local computing, thus the number of offloading tasks keeps steady and the



(a)



(b)

Fig. 10. Comparison of energy consumption and task completion time for different schemes. (**b**)Energy consumption. (**b**)Task completion time.

optimal transmit power is constant with Eq. (27). Therefore, the EEC does not decrease with further increasing $p_{n,max}$. Moreover, the curves in Fig. 9 indicate that the faster the vehicle move, the larger the EEC spent.

5.7. Comparison of energy consumption and completion time

For different values of L_n, we compare the energy consumption and task completion time of several related schemes in Fig. 10. We can observe that both the energy consumption and computation completion increase with L_n. As expected, the proposed scheme enjoys better performance than the other four schemes, since it makes the best of DVFS technology and transmit power control. On the one hand, the performance gap between two entire local execution schemes, and that between BO with fixed local CPU frequency scheme and the proposed scheme show the superiority of imposing DVFS technology on the vehicle. On the other hand, the proposed scheme surpasses SDR-based scheme, which verifies the benefit of transmit power control. Finally, the superiority of the proposed scheme shows that the performance achieved by jointly optimizing the communication and computation resources is superior to that obtained by optimizing these resources separately.

6. Conclusion

In this paper, the changing channel conditions, caused by the mobility of vehicles, have been mainly considered in the problem formulation. The tradeoff between latency and energy consumption in the vehicular edge computing network has been investigated. Then the computation offloading and resource allocation problem has been formulated as an EEC minimization problem, by jointly optimizing the offloading decision and resources allocation. Despite the non-convex formulated problem, we have transformed it into a two-level problem, and solved it in a distributed manner. Furthermore, an iterative distributed algorithm has been developed, which is composed of the sub-algorithms of computation offloading decision, transmit power allocation, CPU frequency control and transmission time adjustment. Through numerical results, the effectiveness and superiority of the proposed algorithm has been revealed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China No. 61971245 and 61801249.

Appendix A. Proof of the convexity

In the objective function of Eq. (19), the function $\left(1-a_n\right)\frac{L_nC_n}{f_n^l}$ and $\left(1-a_n\right)kL_nC_n\left(f_n^l\right)^2$ are convex with respect to $\left\{a_n, f_n^l\right\}$, respectively; $a_n\left(t_n^{ot}+\frac{L_nC_n}{f_{MEC}}\right)$ and $a_np_nt_n^{ot}$ are linear in $\left\{a_n, p_n\right\}$, respectively. Thus, the objective function is convex.

To investigate the convexity of $\varphi\left(p_n, t_n^{ot}\right)$ in constraint C3, we first define

$$h(G) \triangleq \int_0^{t_n^{ot}} Q\log_2(G)d\tau \tag{A.1}$$

The first-order and second-order derivatives of $h(G)$ with respect to G are

$$\frac{\partial h(G)}{\partial G} = \frac{t_n^{ot}Q}{G\log 2} \geq 0 \tag{A.2}$$

and

$$\frac{\partial^2 h(G)}{\partial G^2} = -\frac{t_n^{ot}Q}{G^2\log 2} \leq 0 \tag{A.3}$$

Thus, $h(G)$ is concave and non-decreasing monotonically with respect to p_n. Next, we define

$$g\left(p_n\right) \triangleq G = 1+\frac{p_n\rho_0}{\left[H^2+D^2+\left(x_n+v_nt\right)^2\right]^2} \tag{A.4}$$

It is clear that $g(p_n)$ is an affine function with respect to p_n. Therefore, the composite function $f(p_n)=\varphi\left(p_n, t_n^{ot}\right)=h\left(g\left(p_n\right)\right)$ is concave with respect to p_n [30, chap. 3]. Hence, the constraint function in C3 is convex.

Moreover, the other constraint functions of Eq. (19) are affine functions with respect to $\left\{a_n, p_n, f_n^l\right\}$, which are also convex. In this way, the optimization problem (19) is convex.

Appendix B. Proof of Lemma 2

Similar to the proof of Lemma 1, the objective function in Eq. (26) is strictly convex in $0 \leq p_n \leq p_{n,max}$. Subsequently, the KKT conditions [30] are used to take the optimal communication resource allocation P^*. By differentiating Eq. (26) with respect to p_n, and making it equal 0, it can be achieved that the unique root $\hat{p}_n$ of equation $\beta_n^E t_n^{ot} - \chi_n \varphi'\left(p_n, t_n^{ot}\right) = 0$ is the optimal to minimize the objective function of Eq. (26), where $\varphi'\left(p_n, t_n^{ot}\right) \triangleq \frac{\partial(p_n, t_n^{ot})}{\partial p_n}$. Recall the definition of $\varphi\left(p_n, t_n^{ot}\right)$ in Section 3.4 and $r_n(t)$ in Eq. (3), and we can conduct that $\varphi\left(p_n, t_n^{ot}\right)$ monotonically increases with p_n. Next, we will analyze the optimal solution of Eq. (26) with constraint C6.

If $\hat{p}_n < 0$, $p_n^* = 0$, since the objective function of Eq. (26) monotonically increases in $\left[0, p_{n,max}\right]$ and reaches the minimum at $p_n = 0$. If $\hat{p}_n > p_{n,max}$, $p_n^* = p_{n,max}$, since the objective function of Eq. (26) monotonically decreases in $\left[0, p_{n,max}\right]$ and reaches the minimum at $p_n = p_{n,max}$. If $0 \leq \hat{p}_n \leq p_{n,max}$, $p_n = \hat{p}_n$, since the objective function of Eq. (26) monotonically decreases in $\left[0, \hat{p}_n\right]$, increases in $\left[\hat{p}_n, p_{n,max}\right]$, and reaches the minimum at $p_n = \hat{p}_n$.

References

[1] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, X.S. Shen, Toward efficient content delivery for automated driving services: An edge computing solution, IEEE Netw. 32 (1) (2018) 80–86, http://dx.doi.org/10.1109/MNET.2018.1700105.

[2] A.u.R. Khan, M. Othman, S.A. Madani, S.U. Khan, A survey of mobile cloud computing application models, IEEE Commun. Surv. Tutor. 16 (1) (2014) 393–413, http://dx.doi.org/10.1109/SURV.2013.062613.00160.

[3] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656, http://dx.doi.org/10.1109/COMST.2017.2682318.

[4] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: Paving the last mile of artificial intelligence with edge computing, Proc. IEEE 107 (8) (2019) 1738–1762, http://dx.doi.org/10.1109/JPROC.2019.2918951.

[5] H.Q. Le, H. Al-Shatri, A. Klein, Efficient resource allocation in mobile-edge computation offloading: Completion time minimization, in: 2017 IEEE International Symposium on Information Theory, ISIT, 2017, pp. 2513–2517, http://dx.doi.org/10.1109/ISIT.2017.8006982.

[6] J. Ren, G. Yu, Y. Cai, Y. He, Latency optimization for resource allocation in mobile-edge computation offloading, IEEE Trans. Wireless Commun. 17 (8) (2018) 5506–5519, http://dx.doi.org/10.1109/TWC.2018.2845360.

[7] Y. Wu, L.P. Qian, K. Ni, C. Zhang, X. Shen, Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading, IEEE J. Sel. Top. Sign. Proces. 13 (3) (2019) 392–407.

[8] C. You, Y. Zeng, R. Zhang, K. Huang, Asynchronous mobile-edge computation offloading: Energy-efficient resource management, IEEE Trans. Wireless Commun. 17 (11) (2018) 7590–7605, http://dx.doi.org/10.1109/TWC.2018.2868710.

[9] M. Sheng, Y. Wang, X. Wang, J. Li, Energy-efficient multiuser partial computation offloading with collaboration of terminals, radio access network, and edge server, IEEE Trans. Commun. 68 (3) (2020) 1524–1537, http://dx.doi.org/10.1109/TCOMM.2019.2959338.

[10] X. Lyu, H. Tian, C. Sengul, P. Zhang, Multiuser joint task offloading and resource optimization in proximate clouds, IEEE Trans. Veh. Technol. 66 (4) (2017) 3435–3447, http://dx.doi.org/10.1109/TVT.2016.2593486.

[11] S. Bi, Y.J. Zhang, Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading, IEEE Trans. Wireless Commun. 17 (6) (2018) 4177–4190, http://dx.doi.org/10.1109/TWC.2018.2821664.

[12] T.Q. Dinh, J. Tang, Q.D. La, T.Q.S. Quek, Offloading in mobile edge computing: Task allocation and computational frequency scaling, IEEE Trans. Commun. 65 (8) (2017) 3571–3584, http://dx.doi.org/10.1109/TCOMM.2017.2699660.

[13] S. Guo, J. Liu, Y. Yang, B. Xiao, Z. Li, Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing, IEEE Trans. Mob. Comput. 18 (2) (2019) 319–333, http://dx.doi.org/10.1109/TMC.2018.2831230.

[14] Y. Ding, C. Liu, X. Zhou, Z. Liu, Z. Tang, A code-oriented partitioning computation offloading strategy for multiple users and multiple mobile edge computing servers, IEEE Trans. Ind. Inf. 16 (7) (2020) 4800–4810, http://dx.doi.org/10.1109/TII.2019.2951206.

[15] A. Ebrahimzadeh, M. Maier, Cooperative computation offloading in FiWi enhanced 4G hetnets using self-organizing MEC, IEEE Trans. Wireless Commun. 19 (7) (2020) 4480–4493, http://dx.doi.org/10.1109/TWC.2020.2983890.

[16] K. Wang, H. Yin, W. Quan, G. Min, Enabling collaborative edge computing for software defined vehicular networks, IEEE Netw. 32 (5) (2018) 112–117, http://dx.doi.org/10.1109/MNET.2018.1700364.

[17] X. Huang, R. Yu, J. Kang, Y. Zhang, Distributed reputation management for secure and efficient vehicular edge computing and networks, IEEE Access 5 (2017) 25408–25420, http://dx.doi.org/10.1109/ACCESS.2017.2769878.

[18] J. Feng, Z. Liu, C. Wu, Y. Ji, AVE: Autonomous vehicular edge computing framework with ACO-based scheduling, IEEE Trans. Veh. Technol. 66 (12) (2017) 10660–10675, http://dx.doi.org/10.1109/TVT.2017.2714704.

[19] K. Zhang, Y. Mao, S. Leng, S. Maharjan, Y. Zhang, Optimal delay constrained offloading for vehicular edge computing networks, in: 2017 IEEE International Conference on Communications, ICC, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICC.2017.7997360.

[20] K. Zhang, Y. Mao, S. Leng, Y. He, Y. Zhang, Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading, IEEE Veh. Technol. Mag. 12 (2) (2017) 36–44, http://dx.doi.org/10.1109/MVT.2017.2668838.

[21] Y. Dai, D. Xu, S. Maharjan, Y. Zhang, Joint load balancing and offloading in vehicular edge computing and networks, IEEE Internet Things J. 6 (3) (2019) 4377–4387, http://dx.doi.org/10.1109/JIOT.2018.2876298.

[22] P. Liu, J. Li, Z. Sun, Matching-based task offloading for vehicular edge computing, IEEE Access 7 (2019) 27628–27640, http://dx.doi.org/10.1109/ACCESS.2019.2896000.

[23] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, H. Zhou, V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture, IEEE Access 6 (2018) 17741–17755, http://dx.doi.org/10.1109/ACCESS.2018.2820679.

[24] G. Qiao, S. Leng, K. Zhang, Y. He, Collaborative task offloading in vehicular edge multi-access networks, IEEE Commun. Mag. 56 (8) (2018) 48–54, http://dx.doi.org/10.1109/MCOM.2018.1701130.

[25] J. Zhao, Q. Li, Y. Gong, K. Zhang, Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks, IEEE Trans. Veh. Technol. 68 (8) (2019) 7944–7956, http://dx.doi.org/10.1109/TVT.2019.2917890.

[26] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, D. Zhao, A game-based computation offloading method in vehicular multiaccess edge computing networks, IEEE Internet Things J. 7 (6) (2020) 4987–4996, http://dx.doi.org/10.1109/JIOT.2020.2972061.

[27] L.T. Tan, R.Q. Hu, Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning, IEEE Trans. Veh. Technol. 67 (11) (2018) 10190–10203, http://dx.doi.org/10.1109/TVT.2018.2867191.

[28] L.T. Tan, R.Q. Hu, L. Hanzo, Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks, IEEE Trans. Veh. Technol. 68 (4) (2019) 3086–3099, http://dx.doi.org/10.1109/TVT.2019.2893898.

[29] C. Yang, Y. Liu, X. Chen, W. Zhong, S. Xie, Efficient mobility-aware task offloading for vehicular edge computing networks, IEEE Access 7 (2019) 26652–26664, http://dx.doi.org/10.1109/ACCESS.2019.2900530.

[30] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge university press, 2004, pp. 67–103.

[31] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, Proc. IEEE 95 (1) (2007) 255–312, http://dx.doi.org/10.1109/JPROC.2006.887322.

[32] F. Zhou, Y. Wu, R.Q. Hu, Y. Qian, Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems, IEEE J. Sel. Areas Commun. 36 (9) (2018) 1927–1941, http://dx.doi.org/10.1109/JSAC.2018.2864426.

[33] S. Bubeck, Convex optimization: Algorithms and complexity, Found. Trends Mach. Learn. 8 (3–4) (2015) 131–138.

[34] C. You, K. Huang, H. Chae, B. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16 (3) (2017) 1397–1411, http://dx.doi.org/10.1109/TWC.2016.2633522.

[35] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, D.O. Wu, Energy-optimal mobile cloud computing under stochastic wireless channel, IEEE Trans. Wireless Commun. 12 (9) (2013) 4569–4581, http://dx.doi.org/10.1109/TWC.2013.072513.121842.