Deep Learning-Assisted Energy-Efficient Task Offloading in Vehicular Edge Computing Systems

Bodong Shang, *Student Member, IEEE*, Lingjia Liu, *Senior Member, IEEE*, and Zhi Tian, *Fellow, IEEE*

***Abstract*—In this paper, we study an energy-efficient computation offloading for vehicular edge computing systems, where multiple roadside units assist vehicular users to offload computation tasks to edge servers. Our goal is to minimize the users' energy consumption by optimizing user association, data partition, transmit power, and computation resources, subject to the constraints of partial tasks offloading, user latency, maximum transmit power, outage performance, and computation capacity of edge servers. We utilize deep learning for user association to avoid combinatorial complexity, and develop an efficient optimization algorithm to optimize other variables. The resulting algorithm has scalable complexity with convergence guarantee, as confirmed by our theoretical analysis. Simulation results demonstrate that the introduced resource allocation algorithm can significantly reduce the total energy consumption of users.**

***Index Terms*—Energy-efficient communications, computation offloading, vehicular communications, deep learning.**

I. Introduction

In vehicular networks, vehicle and in-vehicle users need to process a set of computation tasks, such as road traffic services, infotainment applications, which involve the execution of data [1]. Due to the extensive workloads and the limited computation capacity at the user side, it is usually difficult to meet the latency requirements when users locally process their computation tasks. Moreover, as users are mobile, they are often subject to strict energy consumption restrictions. As such, it is expected to develop energy-efficient methods to reduce users' energy consumption and guarantee their latency requirements in computing. To this end, vehicular edge computing (VEC) has been proposed as a promising computing architecture for vehicular networks, where users' computation tasks can be offloaded to edge servers via roadside units (RSUs) that can receive computation tasks from users.

In conventional mobile edge computing systems, full channel state information (CSI) is assumed to be available at base stations [2], [3]. This assumption does not always hold in VEC systems, since the channel varies fast due to the mobility of vehicles and it is quite challenging to estimate CSI and feed back to the RSUs [4]. Under the uncertainty of small-scale fading, it remains an open question as to how to jointly allocate both communication and computation resources based on large-scale fading channel information in VEC systems.

Prior Art: To date, several works in the literature investigated the energy minimization of VEC systems. In [5], an energy-efficient workload offloading problem was studied. However, [5] considered the workload problem within the coverage of a single RSU, which lacks the cooperation among multiple edge servers. In [6], the overall system energy consumption was minimized by optimizing the offloading decisions and the number of allocated resource blocks. Note that [6] considered the binary offloading model for tasks, which is suitable only for the highly integrated or relatively simple tasks that cannot be partitioned. In [7], a cloud-fog-vehicular edge cloud architecture was proposed to holistically utilize the available computation resources in a smart city, which minimizes the total power consumption of the end-to-end architecture. However, the computation offloading for vehicles was not analyzed. [8] aimed at minimizing the total energy consumption by by optimizing the offloading decisions. The partial offloading scheme in VEC systems was not studied. [9] minimized vehicles' total energy consumption by jointly optimizing the offloading proportion and bit allocation of vehicles. However, [9] considered the nearest RSU association but did not provide any algorithms or analytical forms of solutions. Furthermore, the above works assumed full CSI or line-of-sight links between vehicle and RSUs. The large-scale shadow fading and the uncertain small-scale fading due to vehicle mobility were neglected, but these factors critically affect the resource allocation.

Contributions: The main contributions of our work are summarized in the following.

- We develop a computation offloading algorithm for VEC systems. Specifically, we minimize the total energy consumption of users by jointly optimizing the variables of user association, data partition, transmit power, and computation resources at edge servers, subject to the constraints of partial offloading, the maximum transmit power, user latency, outage performance, and computation capacity of edge servers.
- We consider the unknown small-scale fading in vehicle-to-RSU (V2R) channels and the outage performance in VEC systems. Moreover, our work is not restricted to a single RSU and an edge server. We consider multiple edge servers and users in VEC systems. Furthermore, we focus on the partial offloading model for computation tasks, making it possible to implement fine-grained computation offloading in VEC systems.
- We utilize deep learning method to obtain user association and integrate it with the developed optimization algorithm. Based on this approach, the computational complexity for obtaining the user association is very low, where the input network parameters simply go through a designed neural network model.

II. System Model

A. Network Layout

We consider a VEC system with M edge servers and K users as shown in Fig. 1. The set of edge servers is denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$, and the set of users is indicated by $\mathcal{K} = \{1, 2, \ldots, K\}$. Each RSU has a wire-connected edge server that has a certain computation resource to process users' computation tasks.

B. Communication Model

Based on the channel modeling for vehicular communications in [4], the V2R channel between the k-th user and the m-th RSU at time slot t is given by $h_{km}(t) = \overline{h}_{km} g_{km}(t)$, where $\overline{h}_{km} = h_{ref} \varsigma_{km} (L_{km})^{-\alpha}$ accounts for the large-scale fading component and $g_{km}(t)$ represents

Manuscript received December 6, 2020; revised March 22, 2021; accepted May 28, 2021. Date of publication June 17, 2021; date of current version September 17, 2021. The work of B. Shang and L. Liu was supported in part by NSF under Grant ECCS-1811497. The review of this article was coordinated by Prof. Kyunghan Lee. *(Corresponding author: Lingjia Liu.)*

Bodong Shang and Lingjia Liu are with the Bradley Department of Electrical and Computer Engineering, Virgina Polytechnic Institute and State University, VA 24061 USA (e-mail: bdshang@vt.edu; ljliu@vt.edu).

Zhi Tian is with the Department of Electrical and Computer Engineering, George Mason University, VA 22030 USA (e-mail: ztian1@gmu.edu).

Digital Object Identifier 10.1109/TVT.2021.3090179

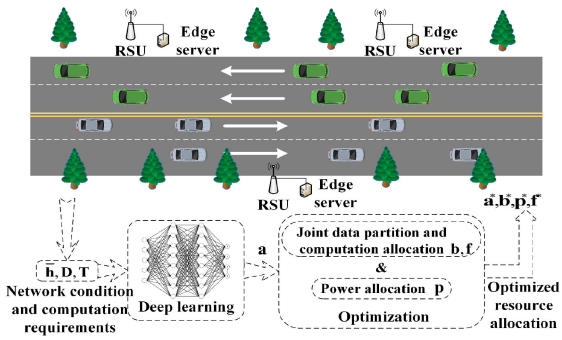

Fig. 1. An architecture of vehicular edge computing systems.

the small-scale fading component at time slot t. Specifically, h_{ref} is channel power gain at reference distance, ς_{km} is the shadowing component, L_{km} denotes the distance between k-th user and m-th RSU, and α is the pathloss exponent. The instantaneous uplink data rate $R_{km}(t)$ of the k-th user connected to the m-th RSU at the time slot t is given by $R_{km}(t) = B\log_2(1 + \frac{h_{km}(t)p_k}{N_0})$, where B is the channel bandwidth, p_k indicates the transmit power of the k-th user and N_0 represents the noise power. We consider that users are allocated with orthogonal resources for uplink transmissions in a specific road segment. The adjacent road segments operate on different frequency bands. Thus, the interference from other road segments on the same frequency band is neglected. The maximum transmit power of the k-th user is denoted by $p_k^{\max}$.

In the time domain, time is equally divided by time slots of length T on the order of hundreds of microseconds. Many consecutive time slots construct a time block on the order of hundreds of milliseconds. The large-scale fading component is typically determined by users' locations which vary little within each time block. We assume that the large-scale fading component is known at RSUs, because the locations of vehicles are usually available at RSUs. However, the small-scale fading component varies rapidly during a time block due to the high mobility of vehicles, which is unavailable at RSUs, but its statistical characterization is assumed to be known. We assume that the small-scale fading component remains constant during one time slot but fluctuates as an independent and identically distributed (i.i.d.) random variable across different time slots. We consider the Rayleigh distribution for small-scale fading with parameter λ_g [4]. The time average data rate of the k-th user connected to the m-th RSU in a time block is given by $R_{km} = \int_0^\infty B\log_2(1 + \frac{\bar{h}_{km}p_k}{N_0}x)f_g(x)dx = \frac{B\lambda_g}{\ln 2}\Phi(\frac{N_0\lambda_g}{\bar{h}_{km}p_k})$, where $f_g(x) = \lambda_g e^{-\lambda_g x}$, $\Phi(x) = e^x E_1(x)$, $E_1(x) = \int_x^\infty \frac{e^{-y}}{y}dy (x \geqslant 0)$ is the exponential integral function, and the exponential distribution of small-scale fading component (i.e., $f_g(x)$) is introduced.

C. Computation Offloading

In partial computation offloading, the k-th user offloads $\beta_k (\beta_k \in [0, 1])$ portion of its data to an edge server, while the remaining $1 - \beta_k$ portion of data is executed locally at the k-th user. The association vector of the k-th user is given by $\mathbf{a}_k = \{a_{k1}, \ldots, a_{kM}\}$, where $a_{km} = 1$ is defined if partial task of the k-th user is offloaded on the m-th server, otherwise, $a_{km} = 0$. Each user can offload partial data to only one edge server, obeying the constraint as follows

$$\sum_{m=1}^{M} a_{km} = 1, \quad \forall k \in \mathcal{K}. \tag{1}$$

Note that if the k-th user executes all of its data locally, we can set β_k to zero, regardless of the values of $\mathbf{a}_k$. In (1), each user is associated with one RSU, facilitating the control information exchange in VEC systems.

The computation task of the k-th user is expressed as $V_k = (D_k, T_k, F_k), \forall k \in \mathcal{K}$, where D_k is the data size of its computation task, T_k is the latency requirement of this task and F_k is the required number of central processing unit (CPU) cycles for processing this task. In general, the required number of CPU cycles is given by $F_k = c_{bc}D_k$, where c_{bc} is the coefficient for bit-to-cycle conversion [2].

D. Computation Model

Each edge server has a limited computation capacity, upper bounded by $f_m^{\max}$, which indicates the maximum number of allocated CPU cycles per second at the m-th edge server. The allocated computation resource for the k-th user at the m-th edge server is denoted by f_{km}, $\forall k, m$. The computation capacity constraint at the m-th edge server is given by

$$\sum_{k=1}^{K} a_{km}f_{km} \leqslant f_m^{\max}, \quad \forall m \in \mathcal{M}. \tag{2}$$

We consider that the k-th user transmits at the rate of R_{km} in a time block to reduce implementation complexity. Accordingly, the latency constraint of the k-th user is given by

$$\max\left\{\sum_{m=1}^{M} a_{km}\beta_k\left(\frac{D_k}{R_{km}} + \frac{F_k}{f_{km}}\right), \frac{(1-\beta_k)F_k}{f_{k0}}\right\} \leqslant T_k, \tag{3}$$

where f_{k0} denotes the local computation capacity of the k-th user. The computing power consumption of k-th user is given by $p_k^c = \rho(f_{k0})^\varsigma$, $\forall k \in \mathcal{K}$, where ρ and ς are constants that depend on the average switched capacitance and the average activity factor, respectively [2].

E. Outage Probability

The outage probability of data transmission for the k-th user connected to the m-th RSU in a time slot is expressed as $\mathbb{P}_{km}^o = \mathbb{P}\{R_{km}(t) \leqslant R_{km}\}$. In addition, the required number of time slots for data transmission is denoted by N_{km}, where $N_{km} = \frac{\beta_k D_k}{TR_{km}}$. We consider that the user's average number of outage time slots is less than a certain threshold μ_o, i.e., $N_{km}\mathbb{P}\{R_{km}(t) \leqslant R_{km}\} \leqslant \mu_o, \forall k \in \mathcal{K}$, where μ_o denotes the threshold of the expected maximum outage time slots for transmitting β_k portion of data to the RSU.

Lemma 1: The outage constraint of the k-th user connected to the m-th RSU is given by

$$\sum_{m=1}^{M}\left\{1 - \exp\left[-\left(2^{\frac{\lambda_g}{\ln 2}\Phi\left(\frac{N_0\lambda_g}{\bar{h}_{km}p_k}\right)} - 1\right)\frac{N_0\lambda_g}{\bar{h}_{km}p_k}\right]\right\} \times \frac{a_{km}\beta_k D_k \ln 2}{TB\lambda_g\Phi\left(\frac{N_0\lambda_g}{\bar{h}_{km}p_k}\right)} \leqslant \mu_o, \quad \forall k \in \mathcal{K}. \tag{4}$$

Proof: Given a_{km}, we have $N_{km} = \frac{\beta_k D_k}{T\frac{B\lambda_g}{\ln 2}\Phi(\frac{N_0\lambda_g}{\overline{h}_{km}p_k})}$. Considering the distribution of small-scale fading, we have

$$\begin{aligned} & N_{km}\mathbb{P}\left\{R_{km}(t) \leqslant R_{km}\right\} \\ &= N_{km}\mathbb{P}\left\{\log_2\left(1+\frac{\overline{h}_{km}p_k g_{km}(t)}{N_0}\right) \leqslant \frac{\lambda_g}{\ln 2}\Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)\right\} \\ &= N_{km}\mathbb{P}\left\{g_{km}(t) \leqslant \left(2^{\frac{\lambda_g}{\ln 2}\Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)}-1\right)\frac{N_0}{\overline{h}_{km}p_k}\right\} \\ &= N_{km}\left\{1-\exp\left[-\left(2^{\frac{\lambda_g}{\ln 2}\Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)}-1\right)\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right]\right\}. \end{aligned} \tag{5}$$

Then, according to $N_{km}\mathbb{P}\{R_{km}(t) \leqslant R_{km}\} \leqslant \mu_o, \quad \forall k \in \mathcal{K}$, we have

$$\frac{1-\exp\left[-\left(2^{\frac{\lambda_g}{\ln 2}\Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)}-1\right)\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right]}{\frac{TB\lambda_g}{\beta_k D_k \ln 2}\Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)} \leqslant \mu_o, \quad \forall k \in \mathcal{K}. \tag{6}$$

Combining a_{km} in (6), we obtain the desired result. ■

F. Problem Formulation

In this paper, we focus on minimizing the total energy consumption of users in VEC systems, which is expressed as follows

$$\min_{\mathbf{a},\mathbf{b},\mathbf{p},\mathbf{f}} \sum_{k=1}^{K}\frac{(1-\beta_k)F_k p_k^c}{f_{k0}} + \sum_{k=1}^{K}\sum_{m=1}^{M}\frac{a_{km}p_k \ln 2\beta_k D_k}{B\lambda_g \Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km}p_k}\right)} \tag{7}$$

$$\text{s.t.}(1),(2),(3),(4),$$

$$a_{km} = \{0,1\}, \quad \forall k \in \mathcal{K}, \quad \forall m \in \mathcal{M}, \tag{7a}$$

$$0 \leqslant p_k \leqslant p_k^{\max}, \quad \forall k \in \mathcal{K}, \tag{7b}$$

$$0 \leqslant \beta_k \leqslant 1, \quad \forall k \in \mathcal{K}, \tag{7c}$$

$$0 \leqslant f_{km}, \quad \forall k \in \mathcal{K}, \quad \forall m \in \mathcal{M}, \tag{7d}$$

where $\mathbf{a} = \{a_{km}\}_{k\in\mathcal{K},m\in\mathcal{M}}$ is user association, $\mathbf{b} = \{\beta_k\}_{k\in\mathcal{K}}$ indicates users' data partition, $\mathbf{p} = \{p_k\}_{k\in\mathcal{K}}$ denotes users' transmit power, and $\mathbf{f} = \{f_{km}\}_{k\in\mathcal{K},m\in\mathcal{M}}$ represents computation resource allocation. In the objective function of problem (7), the first term is the energy consumption for users' local computing, and the second term captures the energy consumption of users' data transmission.

III. Proposed Algorithm

We study an efficient algorithm to solve the users' energy consumption minimization problem (7) in VEC systems.

A. Joint Data Partition and Computation Resource Allocation

Given the user's association $\mathbf{a}$ and power allocation $\mathbf{p}$, we jointly optimize users' data partition $\mathbf{b}$ and computation resource allocation $\mathbf{f}$, where the sub-problem is given by

$$\min_{\mathbf{b},\mathbf{f}} \sum_{k=1}^{K}\frac{(1-\beta_k)F_k p_k^c}{f_{k0}} + \sum_{k=1}^{K}\frac{a_{km}p_k\beta_k D_k}{R_{km}}$$

$$\text{s.t.}(2),(4),(7c),(7d), \tag{8}$$

$$\frac{a_{km}\beta_k D_k}{R_{km}} + \frac{a_{km}F_k\beta_k}{f_{km}} - T_k \leqslant 0, \quad \forall k \in \mathcal{K}, \tag{8a}$$

$$\frac{(1-\beta_k)F_k}{f_{k0}} - T_k \leqslant 0, \quad \forall k \in \mathcal{K}. \tag{8b}$$

It can be verified that the left hand side (LHS) of constraint (8a) is not jointly convex for β_k and f_{km}. To circumvent this issue, we introduce $\gamma_k = \sqrt{\beta_k}$ to transform (8) as follows

$$\min_{\mathbf{c},\mathbf{f}} -\sum_{k=1}^{K}{\gamma_k}^2 c_k \tag{9}$$

$$\text{s.t.}(2),(7d),$$

$$\frac{a_{km}{\gamma_k}^2 D_k}{R_{km}} + \frac{a_{km}{\gamma_k}^2 F_k}{f_{km}} - T_k \leqslant 0, \quad \forall k \in \mathcal{K}, \tag{9a}$$

$$\gamma_k^{lb} \leqslant \gamma_k \leqslant \gamma_k^{ub,1}, \forall k \in \mathcal{K}, \tag{9b}$$

where $c_k = \frac{F_k}{f_{k0}}p_k^c - \frac{a_{km}p_k\beta_k D_k}{R_{km}}$, $\gamma_k^{lb} = \max\{1-\frac{T_k f_{k0}}{F_k}, 0\}^{\frac{1}{2}}$, $\gamma_k^{ub,1} = \min\{1, \sqrt{\frac{\mu_o TB\lambda_g}{\ln 2 D_k}\Theta(\frac{N_0\lambda_g}{\overline{h}_{km}p_k})}\}$, $\Theta(\cdot)$ is given by

$$\Theta(x) = \Phi(x)\left[1-\exp\left(1-e^{\left(1-2^{\lambda_g\Phi(x)/\ln 2}\right)x}\right)\right]^{-1}. \tag{10}$$

It can be verified that the LHS of constraint (9a) is jointly convex for γ_k and f_{km}. However, the convexity of the objective function of problem (9) depends on $c_k, k \in \mathcal{K}$. We partition the set of users $\mathcal{K}$ into two subsets $\mathcal{K}_-$ and $\mathcal{K}_+$, where $\mathcal{K}_- = \{k|k \in \mathcal{K}, c_k \leqslant 0\}$, $\mathcal{K}_+ = \{k|k \in \mathcal{K}, c_k > 0\}$. With the successive convex optimization technique, in each iteration, the objective concave functions are approximated by more tractable functions at given local points. Recall that any concave function is globally upper-bounded by its first-order Taylor expansion at any point. For $k \in \mathcal{K}_+$, we have

$$-\sum_{k\in\mathcal{K}_+}{\gamma_k}^2 c_k \leqslant -\sum_{k\in\mathcal{K}_+}\left(\left(\gamma_k^i\right)^2 c_k + 2\gamma_k^i c_k\left(\gamma_k - \gamma_k^i\right)\right), \tag{11}$$

where γ_k^i is the value of γ_k in the ith iteration. Given γ_k^i, problem (9) is reformulated by

$$\min_{\mathbf{c},\mathbf{f}} -\sum_{k\in\mathcal{K}_-}{\gamma_k}^2 c_k - \sum_{k\in\mathcal{K}_+}\gamma_k^i c_k\left(2\gamma_k - \gamma_k^i\right),$$

$$\text{s.t.}(2),(7d),(9a),(9b), \tag{12}$$

which is a convex problem. In the sequel, we apply the Lagrangian dual method to solve the problem in (12) efficiently by investigating the analytical form of solutions. The Lagarange function of (12) is given by

$$\min_{\mathbf{c},\mathbf{f}} L\left(\mathbf{c},\mathbf{f},\varpi,\vartheta\right), \tag{13}$$

where $L(\mathbf{c},\mathbf{f},\varpi,\vartheta)$ is given in (14) at the top of the page, $\{\varpi_m\}_{m\in\mathcal{M}}$ and $\{\vartheta_k\}_{k\in\mathcal{K}}$ are the Lagrangian multipliers.

$$\begin{aligned} L\left(\mathbf{c},\mathbf{f},\varpi,\vartheta\right) = & -\sum_{k\in\mathcal{K}_-}{\gamma_k}^2 c_k - \sum_{k\in\mathcal{K}_+}\gamma_k^i c_k\left(2\gamma_k - \gamma_k^i\right) \\ & + \sum_{m=1}^{M}\varpi_m\left(\sum_{k=1}^{K}a_{km}^* f_{km} - f_m^{\max}\right) \\ & + \sum_{k=1}^{K}\vartheta_k\left(\frac{a_{km}^*{\gamma_k}^2 D_k}{R_{km}^*} + \frac{a_{km}^*{\gamma_k}^2 F_k}{f_{km}} - T_k\right), \end{aligned} \tag{14}$$

Since (13) is convex, we use the coordinate descent method to find the optimal solution to (13). Specifically, given $\mathbf{f}$, we first optimize $\mathbf{c}$; then, given the optimized $\mathbf{c}$, we optimize $\mathbf{f}$, which are shown as follows

$$\mathbf{c}^{j+1} = \arg\min_{\mathbf{c}} L\left(\mathbf{c}, \mathbf{f}^{j}, \varpi, \vartheta\right), \tag{15}$$

$$\mathbf{f}^{j+1} = \arg\min_{\mathbf{f}} L\left(\mathbf{c}^{j+1}, \mathbf{f}, \varpi, \vartheta\right), \tag{16}$$

where $\mathbf{c}^{j+1}$ and $\mathbf{f}^{j+1}$ denote the optimized $\mathbf{c}$ and $\mathbf{f}$ in the $(j+1)$th iteration, respectively.

Theorem 1: Given $\mathbf{f}^{j}$, for k-th user, if $c_k \leqslant 0$, we have $\gamma_k^{j+1} = \gamma_k^{lb}$; if $c_k > 0$, we have

$$\gamma_k^{j+1} = \begin{cases} \gamma_k^{ub}, \text{ if } \gamma_k^{ub} \leqslant \gamma_k^{opt} \\ \gamma_k^{opt}, \text{ if } \gamma_k^{lb} \leqslant \gamma_k^{opt} < \gamma_k^{ub} \\ \gamma_k^{lb}, \text{ if } \gamma_k^{opt} < \gamma_k^{ub} \end{cases}, \tag{17}$$

$$\text{where } \gamma_k^{opt} = \gamma_k^{i} c_k \left(\vartheta_k \Upsilon_k^{j}\right)^{-1}, \tag{18}$$

$$\gamma_k^{ub} = \min\left\{T_k \left(\Upsilon_k^{j}\right)^{-1}, \gamma_k^{ub,1}\right\}^{\frac{1}{2}}, \tag{19}$$

$$\text{and } \Upsilon_k^{j} = \sum_{m=1}^{M} \frac{a_{km} D_k}{R_{km}} + \frac{a_{km} F_k}{f_{km}^{j}}. \tag{20}$$

Given the optimized $\mathbf{c}^{j+1}$, the optimal f_{km}^{j+1} is given by

$$f_{km}^{j+1} = \gamma_k^{j+1} \left(\frac{\vartheta_k F_k}{\varpi_m}\right)^{\frac{1}{2}}. \tag{21}$$

Proof: The results can be obtained by exploring the first-order optimality conditions of the quadratic function $L(\mathbf{c}, \mathbf{f}, \varpi, \vartheta)$. The detailed proof is omitted here to save space. ■

Once obtaining the optimized $\mathbf{c}$ and $\mathbf{f}$, we update Lagrangian multipliers, as follows

$$\varpi_m^{i+1} = \left[\varpi_m^{i} + \pi_{\varpi}\left(\sum_{k=1}^{K} a_{km} f_{km} - f_m^{\max}\right)\right]^{+}, \tag{22}$$

$$\vartheta_k^{i+1} = \left[\vartheta_k^{i} + \pi_{\vartheta}\left(\gamma_k{}^{2} a_{km}\left(\frac{D_k}{R_{km}} + \frac{F_k}{f_{km}}\right) - T_k\right)\right]^{+}, \tag{23}$$

where π_{ϖ} and π_{ϑ} are the chosen step-sizes.

By optimizing $\mathbf{c}$, $\mathbf{f}$, and updating ϖ, ϑ, we obtain the optimal $\mathbf{c}$ and $\mathbf{f}$. Then, we calculate back $\mathbf{b}$ from $\mathbf{c}$.

B. Power Allocation

Given the user's association $\mathbf{a}$, data partition $\mathbf{b}$, and computation resource allocation $\mathbf{f}$, the power allocation sub-problem is given by

$$\min_{\mathbf{p}} \sum_{k=1}^{K}\sum_{m=1}^{M} \frac{a_{km} p_k \ln 2 \beta_k D_k}{B\lambda_g \Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km} p_k}\right)} \tag{24}$$

$$\text{s.t.} (4), (7b),$$

$$\sum_{m=1}^{M} \frac{a_{km} \ln 2 \beta_k D_k}{B\lambda_g \Phi\left(\frac{N_0\lambda_g}{\overline{h}_{km} p_k}\right)} + \frac{a_{km}\beta_k F_k}{f_{km}} \leqslant T_k, \quad \forall k \in \mathcal{K}. \tag{24a}$$

Problem (24) can be decomposed into K sub-problems. For ease of analysis, we introduce the variable $\eta_k = \frac{N_0\lambda_g}{\overline{h}_{km} p_k}$. The power allocation sub-problem for the k-th user is given by

$$\min_{\eta_k} \frac{\ln 2 \beta_k D_k N_0}{B \overline{h}_{km} \Phi(\eta_k) \eta_k} \tag{25}$$

$$\text{s.t.} \frac{\ln 2 \beta_k D_k}{B\lambda_g \Phi(\eta_k)} + \frac{\beta_k F_k}{f_{km}} \leqslant T_k, \tag{25a}$$

$$\left(1 - e^{\left(1 - 2^{\lambda_g \Phi(\eta_k)/\ln 2}\right)\eta_k}\right) \frac{\beta_k D_k \ln 2}{TB\lambda_g \Phi(\eta_k)} \leqslant \mu_o, \tag{25b}$$

$$\eta_k \geqslant \frac{N_0\lambda_g}{\overline{h}_{km} p_k^{\max}}. \tag{25c}$$

In the following Theorem, we provide the optimal expression of η_k regarding the problem (25).

Theorem 2: The optimal η_k in (25) is given by

$$\eta_k^{*} = \max\left\{\eta_k^{lb}, \min\left\{\eta_k^{ub,1}, \eta_k^{ub,2}\right\}\right\}, \tag{26}$$

where $\eta_k^{lb} = \frac{N_0\lambda_g}{\overline{h}_{km} p_k^{\max}}$, $\eta_k^{ub,1} = \Phi^{-1}\left(\frac{\ln 2 \beta_k D_k f_{km}}{B\lambda_g (T_k f_{km} - \beta_k F_k)}\right)$, and $\eta_k^{ub,2} = \Theta^{-1}\left(\frac{\beta_k \ln 2 D_k}{\mu_o T B \lambda_g}\right)$. $\Phi^{-1}(\cdot)$ and $\Theta^{-1}(\cdot)$ are the inverse functions of $\Phi(\cdot)$ and $\Theta(\cdot)$ (defined in (10)), respectively.

Proof: Due to the fact that $\frac{\partial \Phi(x) x}{\partial x} = \Phi(x) x + \Phi(x) - 1 \geqslant 0$, the function $\Phi(x)x$ increases with x. Thus, the objective value in (25) decreases with η_k, and the optimal η_k^{*} takes as large value as possible. Since $\frac{\partial \Phi(x)}{\partial x} = \Phi(x) - \frac{1}{x} \leqslant 0$, the LHS of (25a) increases with η_k. The upper bound of η_k in (25a), i.e., $\eta_k^{ub,1}$, is obtained by taking equality in (25a). In addition, the LHS of (25b) is an increasing function with respect to η_k. Thus, the upper bound of η_k in (25b), i.e., $\eta_k^{ub,2}$, is obtained by taking equality in (25b). Combining the upper bounds (i.e., $\eta_k^{ub,1}$, $\eta_k^{ub,2}$) and the lower bound of η_k (i.e., $\eta_k^{lb} = \frac{N_0\lambda_g}{\overline{h}_{km} p_k^{\max}}$), we obtain the desired results. ■

Then, we can reach the optimal p_k by using $p_k = \frac{N_0\lambda_g}{\overline{h}_{km}\eta_k}$.

C. User Association

We apply a deep neural network (DNN) to obtain user association schemes. The motivation is two-fold: a) this achieves a low computational complexity by simply going through a neural network model implemented in a real-time manner; b) we can enlarge and update the training dataset to obtain the desired system performance. The inputs of the DNN are the large-scale fading components $\overline{\mathbf{h}} = \{\overline{h}_{km}\}$, users' data size $\mathbf{D} = \{D_k\}$, and user latency requirements $\mathbf{T} = \{T_k\}$, while the output of the DNN is the user association scheme.

We utilize a exploitation and exploration policy to generate the dataset. Specifically, given $\overline{\mathbf{h}}, \mathbf{D}, \mathbf{T}$, we first obtain a user association scheme based on the nearest RSU association scheme. Then, we develop one-step exploration, where we change the association scheme of one of the K users while keeping other user associations the same. Since each user can access the other $M-1$ RSUs in the one-step exploration, there are $N_{one} = K(M-1)$ possible schemes in the whole one-step exploration. Next, N_{ran} user association schemes are generated with random exploration, where each user randomly selects one of M RSUs with probability $\frac{1}{M}$. Given each user association scheme, by optimizing $\mathbf{b}, \mathbf{f}$, and $\mathbf{p}$, we select the one with the lowest energy consumption from $1 + N_{one} + N_{ran}$ schemes as the output of the data pair.

In Fig. 2, we show the deep learning model for obtaining $\mathbf{a}$ in VEC systems. In the output layer, we obtain the output values with the value range of $[0, 1]$ by using Sigmoid functions. Then, we obtain the binary output values by selecting the RSU with the maximum output value for each user. Considering the difference of input values, we execute the data pre-processing procedure, including the integration and

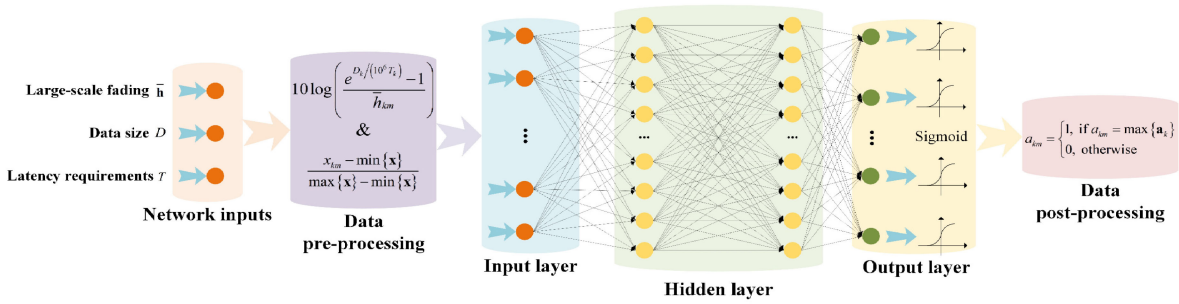

Fig. 2. Deep learning model for obtaining user association $\mathbf{a}$ in VEC systems.

Algorithm 1: Energy-efficient RSU-assisted VEC algorithm.

REQUIRE:
The tolerance ε_{obj}, the maximum iteration number $N_{itera}^{\max}$.
ENSURE:
1: Given network conditions, obtain $\mathbf{a}$ with deep learning;
2: **while** $i \leqslant N_{itera}^{\max}$ **do**
3: Updating iteration index $i = i + 1$;
4: With fixed $\mathbf{p}^{(i-1)}$, jointly optimize $\mathbf{b}^{(i)}$ and $\mathbf{f}^{(i)}$ according to problem (8);
5: With fixed $\mathbf{b}^{(i)}$ and $\mathbf{f}^{(i)}$, optimize $\mathbf{p}^{(i)}$ in problem (24);
6: Obtain the objective value $E(\mathbf{b}^i, \mathbf{p}^i, \mathbf{f}^i)$;
7: If $(E^{i-1} - E^i)/E^i \leqslant \varepsilon_{obj}$: Break; End If;
8: **end while**
9: **return** $(\mathbf{a}^*, \mathbf{b}^*, \mathbf{p}^*, \mathbf{f}^*)$; The energy consumption E^*.

normalization methods, as shown in Fig. 2. Specifically, for any k and m, we obtain $x_{km} = 10\log(\frac{e^{D_k/(10^6 T_k)}-1}{\bar{h}_{km}})$ by considering the units of the variables. Then, we normalize the inputs based on $\frac{x_{km}-\min\{\mathbf{x}\}}{\max\{\mathbf{x}\}-\min\{\mathbf{x}\}}$ to scale input values between 0 and 1. The numbers of neurons in the input and output layers are equal to KM. After the training phase, we can use the trained DNN to calculate $\mathbf{a}$ for any $\bar{\mathbf{h}}, \mathbf{D}, \mathbf{T}$.

D. Algorithm, Convergence and Complexity

1) Algorithm: Algorithm 1 shows an iterative algorithm for solving problem (7), where $E^i = E(\mathbf{b}^i, \mathbf{p}^i, \mathbf{f}^i)$ represents the total energy consumption of users in the i-th iteration.

2) Convergence: Algorithm 1 has theoretical guarantee of convergence which is shown as follows

$$E\left(\mathbf{b}^{i-1}, \mathbf{p}^{i-1}, \mathbf{f}^{i-1}\right) \overset{(a)}{=} E^{ub}\left(\mathbf{b}^{i-1}, \mathbf{p}^{i-1}, \mathbf{f}^{i-1}\right) \overset{(b)}{\geqslant} E^{ub}\left(\mathbf{b}^{i}, \mathbf{p}^{i-1}, \mathbf{f}^{i}\right) \overset{(c)}{\geqslant} E\left(\mathbf{b}^{i}, \mathbf{p}^{i-1}, \mathbf{f}^{i}\right) \overset{(d)}{\geqslant} E\left(\mathbf{b}^{i}, \mathbf{p}^{i}, \mathbf{f}^{i}\right), \quad (27)$$

where E^{ub} denotes the total energy consumption of users based on problem (12), (a) holds since the first-order Taylor expansion in (11) is tight at given local points, (b) holds since $\mathbf{b}^i$ and $\mathbf{f}^i$ are jointly solved optimally in problem (12); (c) holds since E^{ub} is an upper bound of E with $\mathbf{b}^i$ and $\mathbf{f}^i$; (d) holds since $\mathbf{p}^{(i)}$ is the optimal solution to problem (24). Therefore, we have $E(\mathbf{b}^{i-1}, \mathbf{p}^{i-1}, \mathbf{f}^{i-1}) \geqslant E(\mathbf{b}^i, \mathbf{p}^i, \mathbf{f}^i)$. Since E is always positive, Algorithm 1 converges.

3) Complexity: Algorithm 1 incurs polynomial complexity in computation. The complexity to obtain $\mathbf{a}$ by DNN is $C_{DL} = \sum_{l=1}^{Layers}(n^{(l)}n^{(l-1)} + n^{(l)}) + KM$, where $n^{(l)}$ is the number of neurons including the bias unit in the l-th layer. To solve problem (8), the complexity of updating $\mathbf{c}$ and $\mathbf{f}$ in (13) is $O(K)$ due to the closed-form solutions based on Theorem 1. The complexity of updating dual variables is $O(K + M)$. The total complexity of problem (8) is $O(N_{\text{bf}}(K + M))$, where N_{bf} denotes the number of iterations of problem (8).

To solve problem (24), we denote the complexity of inverse functions $\Phi^{-1}(\cdot)$ and $\Theta^{-1}(\cdot)$ as $O(\frac{1}{\varepsilon_1})$ and $O(\frac{1}{\varepsilon_2})$, respectively. Since the solution to (25) is in closed-form, the total complexity of problem (24) is $O(K\log_2(\frac{1}{\varepsilon_1\varepsilon_2}))$.

The total complexity of Algorithm 1 is given by

$$O\left(N_{\text{I}}\left(N_{\text{bf}}\left(K+M\right)+K\log_2\left(\frac{1}{\varepsilon_1\varepsilon_2}\right)\right)+C_{DL}\right), \quad (28)$$

where N_{I} is the number of outer iterations of Algorithm 1.

TABLE I
DEFAULT PARAMETERS SETUP

Parameter	Value
K, M, p_k^{max}	10, 2, 0.5 W
Server's computation capacity f_m^{max}	1×10^{10} cycles/s
Data size D_k	[0.5,1.5] Mbits
Latency requirement T_k, time slot T	500 ms, 1 ms
Bit and cycle conversion c_{bc}	1×10^3 cycles/bit
User's computation capacity f_{k0}	2×10^9 cycles/s
Coefficients ρ, ς	1×10^{-27}, 3
Threshold of outage time slots μ_o	250
Noise spectral density n_0	-80 dBm
Path loss exponent α, bandwidth B	3, 0.5 MHz

IV. Numerical Results and Discussions

In this section, numerical results are provided to assess the proposed algorithm for VEC systems. Default parameter settings are shown in Table I. The road length is 200 m and the number of lanes is 4. RSUs are equally spaced beside the road. Users are randomly distributed on the lanes. Given an average data size $D_{ave} = 1$ Mbits, user's data size is randomly generated within $[0.5D_{ave}, 1.5D_{ave}]$. For each setup, the result is averaged over 1000 tests. The learning rate is 0.01 with decay rate 0.1. The number of samples is 2048. The number of neurons in hidden layer is 800. The size of mini-batch is 10. The number of random exploration N_{ran} is 50.

In Fig. 3, we compare the average energy consumption of users, versus the number of users, under different methods. When K increases, users will consume more energy to execute their computation tasks due to the reduced allocated computation resource for each user. In addition, we compare the proposed deep learning method with other methods, including exhaustive search, Lagrangian dual method [3], nearest RSU association, and random RSU association. It shows that the proposed deep learning-assisted computation offloading method approaches the

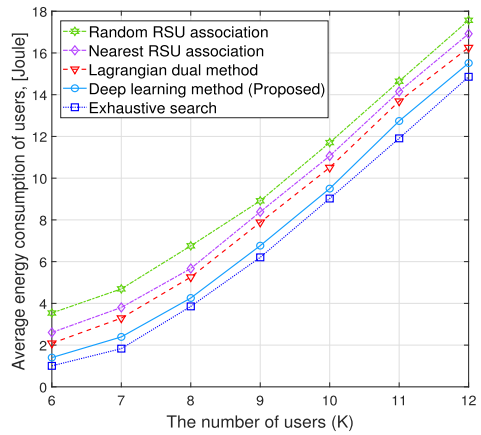

Fig. 3. Average users' energy consumption versus K.

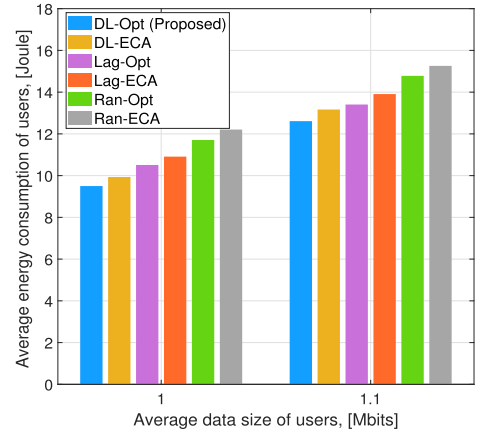

Fig. 4. Average users' energy consumption versus D_{ave}.

optimal one (i.e., exhaustive search). Moreover, it achieves a smaller E than the other three methods. This is because the DNN model can capture the relationships between input and output data pairs, and thus it gives a near-optimal solution. Moreover, note that connecting to the nearest RSUs leads to reliable data rates, but the system neglects the cooperation among edge servers for load balancing.

In Fig. 4, we examine the different algorithms in terms of E. Specifically, we compare the proposed deep learning + joint $\mathbf{b}, \mathbf{p}, \mathbf{f}$ optimization (Opt) method (labeled as 'DL-Opt'), the deep learning + equal computation resource allocation (ECA) method (labeled as 'DL-ECA'), the Lagrangian dual + Opt method (labeled as 'Lag-Opt'), the Lagrangian dual + ECA method (labeled as 'Lag-ECA'), the random RSU association + Opt method (labeled as 'Ran-Opt'), the random RSU association + ECA method (labeled as 'Ran-ECA'). It is observed that the proposed algorithm achieves the lowest energy consumption than other algorithms, which demonstrates the advantage of this work.

V. Conclusion

In this paper, we have developed a deep learning-assisted energy-efficient computation offloading algorithm for VEC systems. The algorithm can solve the complex VEC problem and find a near-optimal solution in a real-time manner with low complexity. Simulation results demonstrate the advantages of the proposed algorithm in substantially reducing users' total energy consumption compared with other methods.

References

[1] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Aug. 2019.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.

[3] B. Shang and L. Liu, "Mobile-edge computing in the sky: Energy optimization for air-ground integrated networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7443–7456, Aug. 2020.

[4] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, Jul. 2017.

[5] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5087–5099, May 2019.

[6] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, "Energy-efficient computation offloading in vehicular edge cloud computing," *IEEE Access*, vol. 8, pp. 37 632–37 644, 2020.

[7] A. A. Alahmadi, T. El-gorashi, and J. Elmirghani, "Energy efficient processing allocation in opportunistic cloud-fog-vehicular edge cloud architectures," 2020, *arXiv:2006.14659*.

[8] T. Yang, Y. Zhu, Y. Hu, and R. Mathar, "Energy minimization of delay-constrained offloading in vehicular edge computing networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop*, 2019, pp. 1–6.

[9] Y. Jang, J. Na, S. Jeong, and J. Kang, "Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–5.