

Joint Sensing, Communication, and Computation Resource Allocation for Cooperative Perception in Fog-Based Vehicular Networks

Xinran Zhang, Zhimin He, Yaohua Sun, Shuo Yuan, Mugen Peng
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
 Beijing, China
 {xrzhang819, hezhimin-sgr, sunyaohua, yuanshuo, pmg}@bupt.edu.cn

Abstract—To enlarge the perception range and reliability of individual autonomous vehicles, cooperative perception has been received much attention. However, considering the high volume of shared messages, limited bandwidth and computation resources in vehicular networks become bottlenecks. In this paper, we investigate how to balance the volume of shared messages and constrained resources in fog-based vehicular networks. To this end, we first characterize sum satisfaction of cooperative perception taking account of its spatial-temporal value and latency performance. Next, the sensing block message, communication resource block, and computation resource are jointly allocated to maximize the sum satisfaction of cooperative perception, while satisfying the maximum latency and sojourn time constraints of vehicles. Owing to its non-convexity, we decouple the original problem into two separate sub-problems and devise corresponding solutions. Simulation results demonstrate that our proposed scheme can effectively boost the sum satisfaction of cooperative perception compared with existing baselines.

Index Terms—Fog-based vehicular networks, cooperative perception, joint sensing, communication, and computation resource allocation, multi-agent deep reinforcement learning

I. INTRODUCTION

With the rapid advancements of embedded systems, sensors, and artificial intelligence in recent decades, autonomous driving has evolved from an impossible dream into a foreseeable reality, and will eventually reshape future transport system and driving experience [1]. As the key enabling technology, cooperative perception [2] exploits vehicle-to-everything (V2X) communications to exchange sensory messages among neighboring vehicles and road side units, and offers an eagle view of broader road environment. In [3], a remarkable perception performance gain over traditional separate perception is validated through theoretical analysis, regardless of the additional communication and computation overhead. Nevertheless, when the number of involved vehicles is 15, the required communication capacity reaches 500 Mbps [4], [5], which overwhelms limited communication bandwidth and computation power in vehicular networks (VNETs) and restricts the implementation of cooperative perception.

To balance the exchanging message volume in cooperative perception and the limited resources in VNETs, some efforts have been made for the joint allocation of sensing, communi-

cation, and computation resource. Higuchi *et al.* [6] attempt to reduce the redundancy of cooperative perception messages by selecting more valuable ones to be delivered. Specifically, the value of messages is firstly quantified based on message history and current mobility, and then more valuable messages are transmitted via V2X communications. In [7], a quadtree-based compression mechanism is utilized to partition sensory information into different block messages. Then, the vehicular association, resource allocation, and block message selection are jointly optimized with deep reinforcement learning (DRL) to maximize the satisfaction of cooperative perception. Du *et al.* [8] go a step further by proposing a platoon-based cooperative perception framework, in which the perception scheduling, computation strategies, and communication resource allocation are jointly optimized to minimize the delay of sensing tasks.

The aforementioned works [6]–[8] commonly concentrate on the direct sharing among vehicles, leaving the more general problem of VNET-assisted cooperative perception less investigated. By taking advantages of hierarchical computation resources and flexible resource management, fog-based vehicular networks (FVNETs) [9] provide a promising edge-cloud collaboration framework. In this paper, we examine how to realize cooperative perception with constrained resources in FVNETs, where multiple vehicles offload individual sensory messages to fog access points (F-APs) or the cloud server for cooperative computation. The main contribution are summarized three-folds.

- Taking spatial-temporal values of sensory messages and latency performance into consideration, we propose a joint sensing block message, communication and computation resource allocation problem to maximize the sum satisfaction of cooperative perception, while satisfying the maximum tolerant latency and sojourn time constraints.
- The non-convex sum satisfaction maximization problem is decoupled into a joint block message and offloading mode selection problem plus a joint communication and computation resource allocation problem. By solving two problems based on multi-agent DRL and swap matching, the sum satisfaction can be maximized.

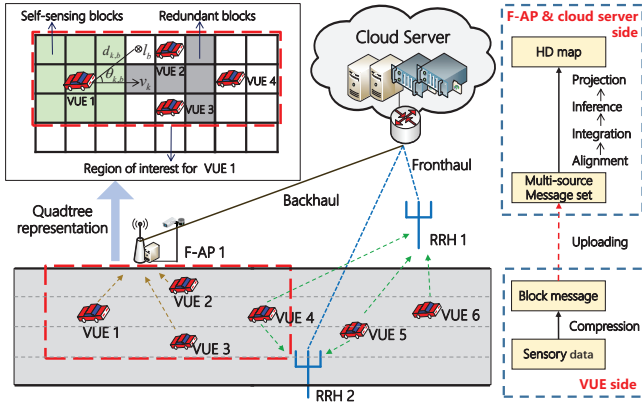


Fig. 1. Cooperative Perception in FVNETs.

- Through simulation results, we show that our proposed scheme can significantly improve the sum satisfaction of cooperative perception in FVNETs.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Considering a cooperative perception scenario in FVNETs, which consists of K vehicular user equipments (VUEs), N F-APs, and M remote radio heads (RRHs) connecting to a centralized cloud server. As shown in Fig.1, each VUE adopts region quadtree technique [7] to compress its sensory data into independent block messages $b \in \mathcal{B}_k \subseteq \mathcal{B}$. Here, \mathcal{B}_k and \mathcal{B} denote the block sets of VUE k 's sensory region and the whole region, respectively. With efficient machine learning methods, the cloud server can implicitly estimate VUE k 's trajectory $l_k(t) = (x_k(t), y_k(t))$ and sojourn time $\tau_{k,n}^{\text{soj}}(t)$ with F-AP n for time $t \in [0, T]$, where $(x_k(t), y_k(t))$ are coordinates.

For cooperative perception, each VUE requests a map constructing task \mathcal{T}_k which covers its front region with a length of d^{exp} . To this end, the cloud server chooses different VUEs to constitute independent cooperative clusters, and selects different block messages from cooperative VUEs for uploading. In this article, we assume that the VUEs associating with the same F-AP or the cloud server form a cooperative cluster, and partial messages could be migrated among F-APs and the cloud server via fiber backhaul links.

Once receiving these uploading messages, the cloud server and F-APs carry out customized maps for individual VUEs by sequentially realigning, integrating, inferring, and projecting the collected uploading messages. We define a tuple $(I, \mu, \tau_k^{\text{max}})$ to refer to VUE k 's task, whose elements denote the data size of a block message, the number of CPU-cycle frequencies required to process 1 bit input, and \mathcal{T}_k 's maximum tolerant latency, respectively.

A. Spatial-Temporal Model in Cooperative Perception

To identify the values of block messages in cooperative perception, the spatial-temporal model is detailed as follows.

From the perspective of timeliness, the value of VUE k 's block message b decreases from time of generation at $t_{k,b,0}$ until it hits deadline at $t_{k,b,0} + \tau^{\text{dll}}$. A linear descend function

is used to define the temporal value of VUE k 's block message b at time t as

$$q_{k,b}(t) = -\frac{q_{k,b,0}}{\tau^{\text{dll}}}(t - t_{k,b,0}) + q_{k,b,0}, \quad (1)$$

where $q_{k,b,0} > 0$ is the message's initial value. However, considering that the sensing time $t_{k,b,0}$ and the sensing period are unknown at the cloud server, we model the temporal value $q_{k,b}(t)$ by Markov process to characterize its randomness. For VUE k , the set of potential temporal value is defined as $\mathcal{Q}_{k,b} = \{q_{k,b,1}, q_{k,b,2}, \dots, q_{k,b,Z}\}$. $\Pr_{k,z|z'} = \Pr\{q_{k,b}(t) = q_{k,b,z} | q_{k,b}(t-1) = q_{k,b,z'}\}$ is the probability that the temporal value of VUE k 's block message b at time t is $q_{k,b,z}$ if its value in time $t-1$ is $q_{k,b,z'}$.

As for spatial dimension, each VUE has a rectangular region of interest where the closer sensory blocks are more expected by the VUEs. With VUE k 's and block b 's positions $l_k(t)$, l_b , the euclidean distance $d_{k,b}$ between VUE k and block b and the angle $\theta_{k,b}$ between VUE k 's moving direction and block b can be easily calculated, as illustrated in Fig.1. Thus, we define the spatial value of block b for VUE k as

$$w_{k,b}(t) = \frac{d^{\text{exp}} - |d_{k,b}(t) \cos \theta_{k,b}(t)|}{d^{\text{exp}}}. \quad (2)$$

Finally, the spatial-temporal value of VUE k 's block message b for VUE k can be calculated by $u_{k,k',b}(t) = q_{k',b}(t)w_{k,b}(t)$.

B. Communication and Computation Models

As illustrated above, \mathcal{T}_k includes block message uploading, joint processing, and output downloading process. Like [8], we ignore the download latency. Denote τ_k^{comm} and τ_k^{comp} as the uploading and computation latencies, we have \mathcal{T}_k 's sum latency $\tau_k = \tau_k^{\text{comm}} + \tau_k^{\text{comp}}$. Moreover, each task \mathcal{T}_k could be handled by following two possible modes: the cloud mode and the F-AP mode. Let $x_{k,n} \in \{0, 1\}$ denote the offloading mode selection of VUE k : if $x_{k,0} = 1$, VUE k choose the cloud mode; while $x_{k,n} = 1, n \in \{1, 2, \dots, N\}$, VUE k offloads the task to F-AP n . Next, we describe the communication and computation models considered in both modes in sequence.

1) *Communication model*: In FVNETs, the total bandwidth is divided into S resource blocks (RBs) with W bandwidth each and full frequency reuse is considered for all F-APs and RRHs. Define $a_{k,s} \in \{0, 1\}$ as the RB allocation indicator. $a_{k,s} = 1$ if RB s is allocated to VUE k and $a_{k,s} = 0$, otherwise. Without loss of generality, each VUE is allocated a single RB for uploading. For VUE k associating with F-AP $n \in \mathcal{N}$, its uploading rate can be expressed as

$$R_{k,n}(t) = \sum_{s \in \mathcal{S}} W \log_2 \left(1 + \frac{a_{k,s} p_k |h_{k,n,s}^{(F)}(t)|^2}{\sum_{j=1, j \neq k}^K a_{j,n,s} p_j |h_{j,n,s}^{(F)}(t)|^2 + \sigma^2} \right), \quad (3)$$

where p_k , σ^2 , and $h_{j,n,s}^{(F)}$ are the transmit power, noise power, and channel gain from VUE k to F-AP n on RB s at time t , respectively.

When VUE k selects the cloud mode, it associates with its $|\mathcal{M}_k|$ close RRHs, whose set is described as \mathcal{M}_k . In the meantime, optimal linear detection, i.e. minimum mean square error (MMSE) detection, is performed at the cloud server to mitigate inter-RRH interference, thus the uplink rate of VUE k in the cloud mode at time t is

$$R_{k,0}(t) = \sum_{s \in \mathcal{S}} W \log_2 \left(1 + \frac{a_{k,s} p_k |\mathbf{g}_{k,s}^H \mathbf{h}_{k,s}^{(C)}(t)|^2}{\text{Int}_{k,s} + \sigma^2 \mathbf{g}_{k,s}^H \mathbf{g}_{k,s}} \right), \quad (4)$$

where $\text{Int}_{k,s} = \sum_{j \neq k} a_{j,s} p_j |\mathbf{g}_{k,s}^H \mathbf{h}_{j,k,s}^{(C)}(t)|^2$ is the interference and $\mathbf{g}_{k,s}$ is the MMSE detection vector. $\mathbf{h}_{k,s}^{(C)}(t)$ is the channel gain from VUE k to its associated RRHs \mathcal{M}_k on RB s at time t , while $\mathbf{h}_{j,k,s}^{(C)}(t)$ is the channel from VUE j to the associated RRHs of VUE k .

Note that the offloaded task \mathcal{T}_k starts computing only when the whole VUEs finish uploading. Therefore, the communication latency of task \mathcal{T}_k relies on the maximum one among cooperative VUEs, i.e.,

$$\tau_k^{\text{comm}}(t) = \max \left(\left\{ \frac{\sum_{b \in \mathcal{B}_{k'}} e_{k',b} I}{R_{k',n}(t)} + x_{k',0} \tau^{fh} \right\}_{\forall k' \in \mathcal{K}} \right), \quad (5)$$

where τ^{fh} is fronthaul delay. $e_{k,b} \in \{0, 1\}$ denotes the block message selection of VUE k : if $e_{k,b} = 1$, then block message b of VUE k is selected for uploading.

2) *Computation model*: For cooperative perception, the offloaded task \mathcal{T}_k has to process all uploaded block messages within its region of interest. Assuming that the computation capabilities of the cloud server and each F-AP n are characterized by the maximum CPU-cycle frequency f_0^{max} and f_n^{max} . With $f_{n,k}$ CPU-cycle frequency allocated to VUE k , the computation latency of task \mathcal{T}_k can be obtained by

$$\tau_k^{\text{comp}}(t) = \frac{\sum_{k' \in \mathcal{K}} \sum_{b \in \mathcal{T}_k} e_{k',b} \mu I}{f_{n,k}}. \quad (6)$$

C. Problem Formulation

In FVNETs, the cloud server and F-APs are interested in finishing VUEs' offloaded tasks with more valuable block messages as quickly as possible, therefore we define the satisfaction of VUE k at time t as

$$U_k(t) = \varepsilon_1 \sum_{b \in \mathcal{T}_k} \sum_{k' \neq k} e_{k',b} u_{k,k',b}(t) + \varepsilon_2 (\tau_k^{\text{max}} - \tau_k(t)), \quad (7)$$

where ε_1 and ε_2 are weight parameters. The first part shows the overall spatial-temporal values of block messages from cooperative VUEs, and the second part denotes the effect of task latency.

The key of this work is to maximize the long-term sum satisfaction for cooperative perception in FVNETs by optimizing the offloading mode selection scheme \mathbf{x}^* , the sensory block message selection scheme \mathbf{e}^* , the uplink RB allocation

scheme \mathbf{a}^* , and the frequency resource allocation scheme \mathbf{f}^* . Mathematically, it can be formulated as

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{e}, \mathbf{a}, \mathbf{f}} \quad & \sum_{t=0}^{T-1} \sum_{k \in \mathcal{K}} U_k(t) \\ \text{s.t.} \quad & \text{(a) } \tau_k(t) \leq \min(\tau_k^{\text{max}}, \tau_{k,n}^{\text{soj}}(t)) \quad \forall k \\ & \text{(b) } \sum_{k \in \mathcal{K}} e_{k,b} \leq 1, \quad e_{k,b} \in \{0, 1\}, \quad \forall b \\ & \text{(c) } \sum_{n=0}^N x_{k,n} \leq 1, \quad x_{k,n} \in \{0, 1\}, \quad \forall k \\ & \text{(d) } \sum_{s \in \mathcal{S}} a_{k,s} \leq 1, \quad a_{k,s} \in \{0, 1\}, \quad \forall k \\ & \text{(e) } \sum_{k \in \mathcal{K}} x_{k,n} f_{k,n} \leq f_n^{\text{max}}, \quad \forall n \in 0 \cup \mathcal{N} \end{aligned} \quad (8)$$

where constraints (8a) shows the overall latency of individual VUE should be less than the maximum tolerant latency of \mathcal{T}_k and its sojourn time $\tau_{k,n}^{\text{soj}}$, constraint (8b) regulates that the uploading block messages of different VUEs are non-overlapped, constraints (8c,d) means each VUE can be allocated to a single mode and RB, and constraint (8e) denotes that the allocated CPU-cycle frequencies are not allowed to exceed the frequency budgets.

III. JOINT SENSING, COMMUNICATION, AND COMPUTATION RESOURCE ALLOCATION

Note that the formulated problem (8) is a mixed-integer non-linear programming problem which is in general intractable. In this section, we decouple the original problem into a joint mode selection and block selection sub-problem plus a joint communication and computation resource allocation sub-problem. Afterwards, multi-agent DRL and swap matching-based algorithms are developed.

A. Optimization of Offloading Mode and Block Selection

Under the circumstance that communication and computation resource allocation scheme has been determined, the original problem can be reformulated as

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{e}} \quad & \sum_{t=0}^{T-1} \sum_{k \in \mathcal{K}} U_k(t) \\ \text{s.t.} \quad & \text{(8b), (8c)}. \end{aligned} \quad (9)$$

Since the block message's value is dynamic and unknown, we cannot solve problem (9) using traditional optimization method. Inspired by [10], we resort to multi-agent DRL to resolve this uncertainty.

1) *Multi-Agent Markov Game*: To be concrete, we treat every VUE as an intelligent agent and convert problem (9) into a Markov game with K agents. At time t , each agent k observes local state $s_k(t)$ and selects its own action $a_k(t)$. Given the joint actions $\mathbf{a}(t) = (a_1(t), \dots, a_K(t))$ taken by K agents, the FVNET feeds back new state $\mathbf{s}(t+1) = (s_1(t+1), \dots, s_K(t+1))$ and immediate reward $r_k(t)$. The formal definitions of three elements are given in the following.

State: Each agent k 's local state includes the following parts: τ_k^{max} , the maximum tolerant latency of VUE k ; $l_k(t)$, the coordinate of its own; $q_k(t-1)$, the temporal value of VUE k 's sensory messages at time $t-1$; $\{O_n(t-1)\}_{n \in 0 \cup \mathcal{N}}$,

the number of serving VUEs of the cloud server and F-APs at time $t - 1$; and $r^{sat}(t - 1)$, the latency satisfied ratio.

Action: Consistent with (9), each agent selects the offloading mode and uploading messages $a_k(t) = \left(n, \{e_{k,b}\}_{b \in \mathcal{B}_k}\right)$.

Reward: We consider this game as a fully cooperative one and define the immediate reward function of each VUE as the sum satisfaction, i.e., $r_k(t) = r(t) = \sum_{k \in \mathcal{K}} U_k(t)$.

2) *Attention Multi-Agent Deep Deterministic Policy Gradient (DDPG)-based Algorithm:* Due to non-stationary local states and large action spaces, canonical DRL algorithms, like deep Q-network, are always ineffective. Inspired by [11], [12], multi-agent DDPG algorithm is tuned with an attention mechanism in this article to tackle this problem, namely attention multi-agent DDPG.

In attention multi-agent DDPG, each agent k has its own actor network θ_k and critic network δ_k , acting as the policy and the policy evaluator, respectively. Note that instead of the locals, the global states and actions are considered at the critic network to address the non-stationary and cooperative issues. Furthermore, two *target networks* (θ_k^- and δ_k^-) and *experience replay* are applied to stabilize training and to remove data correlation. We denote \mathcal{D} as the replay buffer with capacity $|\mathcal{D}|$ and express the stored experience of all agents as the tuple (s, a, r, s') . Here, we omit the time index t and denote mark \cdot' as time $t + 1$ for simplicity.

Specifically, the actor network θ_k is responsible for finding a deterministic policy $\mu_k(s_k(t); \theta_k)$ to maximize the cumulative discounted reward $J(\mu_k) = \mathbb{E} \left[\sum_{t'=t}^T \gamma^{t'-t} r(t') \right]$, where γ is the discounted factor. Then, $a_k(t)$ can be obtained by

$$a_k(t) = \mu_k(s_k(t); \theta_k) + \mathcal{N}(t), \quad (10)$$

where $\mathcal{N}(t)$ is the stochastic noise to encourage exploration. For its updating, the parameter θ_k is directly adjusted in the direction of $\nabla_{\theta_k} J(\mu_k)$, which is given by

$$\nabla_{\theta_k} J(\mu_k) = \mathbb{E}_{s, a \sim \mathcal{D}} [\nabla_{\theta_k} \mu_k(a_k | s_k) \nabla_{a_k} Q_k(s, a)]. \quad (11)$$

Here, $Q_k(s, a)$ is the action-value function that is established by the critic network δ_k . It takes as the input of the states s and actions a of all agents and outputs the Q-value for agent k . With $(s, a, r, s') \sim \mathcal{D}$, the weights of agent k 's critic network δ_k can be updated by minimizing the MSE-based loss function $L_k(\delta_k)$, i.e.,

$$L(\delta_k) = \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}} \left[(y_k - Q_k(s, a))^2 \right], \quad (12)$$

where $y_k = r + \gamma \max_{a_k} Q^-(s', a')|_{a'_k = \mu_k^-(s'_k)}$. Here, $Q^-(s, a)$ and $\mu_k^-(s_k)$ are agent k 's target critic and actor networks with weights δ_k^- and θ_k^- , respectively.

Furthermore, we adjust the critic network Q_k with an attention mechanism which facilitates fine-grained and discriminatory treatment of different VUEs. To this end, we firstly customize a multi-layer perception (MLP) network $g_k : (s_k, a_k) \rightarrow e_k$ to reduce the input dimension s, a and extract the higher features $e = (e_1, \dots, e_K)$. Then, agent k 's attention value for the other agents can be calculated by

Algorithm 1 Attention multi-agent DDPG-based algorithm

```

1: Initialize actor, critic, target actor, target critic networks
    $\theta, \theta^-, \delta, \delta^-$ , and the replay memory  $\mathcal{D}$  with capacity  $|\mathcal{D}|$ .
2: for epoch  $e = 1, \dots, E$  do
3:   Initialize the state  $s$ .
4:   for step  $t = 1, \dots, T$  do
5:     Select action  $a_k$  with (10),  $\forall k \in \mathcal{K}$ .
6:     Obtain current reward  $r$  and next state  $s'$ .
7:     Store tuple  $(s, a, r, s')$  into  $\mathcal{D}$ .
8:     for agent  $k = 1, \dots, K$  do
9:       Randomly sample a mini-batch of tuples from  $\mathcal{D}$ .
10:      Update the critic network by minimizing (12).
11:      Update the actor network with (11).
12:      Update the target critic and actor networks
          $\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-, \delta^- \leftarrow \tau\delta + (1 - \tau)\delta^-$ .
13:    end for
14:  end for
15: end for

```

$v_k = \sum_{j \neq k} \alpha_{k,j} v_{k,j}$. Here, $\alpha_{k,j}$ and $v_{k,j}$ denote agent k 's attention weight and value for agent j , which is given by

$$\alpha_{k,j} = \frac{\exp(e_k^T W_q^T W_k e_j)}{\sum_{j \neq k} \exp(e_k^T W_q^T W_k e_j)}, \quad (13)$$

$$v_{k,j} = h(V_k e_j). \quad (14)$$

Wherein, the parameter matrix W_q and W_k constitute a bilinear mapping for e_k and e_j . In addition, $h(\cdot)$ is a ReLU function and V is a transform matrix. With the derived high feature e_k and the attention value v_k , we could establish a new MLP network $Q(e_k, v_k)$ to replace the original critic network $Q(s, a)$.

Finally, we conclude the attention multi-agent DDPG procedures to solving problem (9) in **Algorithm 1**.

B. Optimization of RB and CPU-frequency Allocation

Once the offloading modes and selected blocks have been determined, the original problem is reduced into

$$\begin{aligned} \min_{a, f} \quad & \sum_{k \in \mathcal{K}} \tau_k \\ \text{s.t.} \quad & (8a), (8d), (8e). \end{aligned} \quad (15)$$

Hereafter, the RBs and frequencies are successively allocated.

1) *Swap Matching-based RB Allocation:* With fixed CPU-frequencies, we firstly model the RB allocation problem as a one-to-one matching game and solve it in a distributed and low-complexity manner.

Formally, we define $\Phi : \mathcal{K}_n \leftrightarrow \mathcal{S}$ as the matching function for the associated VUE set $\mathcal{K}_n, n \in 0 \cup \mathcal{N}$ and the bandwidth set \mathcal{S} , which has the following properties

$$\begin{aligned} 1) \quad & k = \Phi(s) \leftrightarrow s = \Phi(k), \quad \forall k \in \mathcal{K}_n, \forall s \in \mathcal{S}, \\ 2) \quad & |\Phi(s)| \leq 1, |\Phi(k)| \leq 1, \quad \forall k \in \mathcal{K}_n, \forall s \in \mathcal{S}, \end{aligned} \quad (16)$$

where condition 1) implies that if RB s matches with VUE k , VUE k also matches with RB s ; condition 2) gives that each

Algorithm 2 Swap matching-based RB allocation algorithm

```

1: Initialize the preference lists  $\mathcal{L}_k, \mathcal{L}_s, \forall k, \forall s$  and the set
   of unmatched VUEs  $\mathcal{K}^{unmatch} = \mathcal{K}$ .
2: while  $\mathcal{K}^{unmatch} \neq \emptyset$  and  $\exists \mathcal{L}_k \neq \emptyset$  do
3:   for  $\forall k \in \mathcal{K}^{unmatch}$  do
4:     VUE  $k$  proposes to its most preferred RB in  $\mathcal{L}_k$  that
     has not rejected it before.
5:   end for
6:   for  $\forall s \in \mathcal{S}$  do
7:     if  $\sum_{k \in \mathcal{K}_n} a_{k,s} = 1$ , RB  $s$  holds the matching and
     rejects all proposals; otherwise, RB  $s$  accepts its
     favorite VUE and rejects the others.
8:   end for
9: end while
10: while there exists a swap-pair  $(k, k')$  which can reduce
    the maximum latency do
11:   Update  $\Phi_k^{k'}$ .
12: end while

```

RB can be matched with one VUE, and each VUE can only be matched with one RB in turn.

The utility of VUE k is defined as the uploading rate on RB s , i.e.,

$$\phi_k = R_{k,n}^s, \quad (17)$$

which indicates that every VUE selfishly favours the RBs with higher offloading rate. As for RB s , it aims to minimize the overall communication latency which equals to the maximum one, thus we define the utility of RB s as

$$\phi_s = \frac{\sum_{b \in B_{k'}} e_{k',b} I}{R_{k,n}^s} + x_{k',0} \tau^{fh}. \quad (18)$$

Based on above analysis, a swap matching-based RB allocation algorithm is developed in **Algorithm 2**. To start the matching process, each VUE $k \in \mathcal{K}_n$ and RB s establish their own preference lists \mathcal{L}_k and \mathcal{L}_s in the descending order. A deferred acceptance algorithm is then adopted for initial matching. To overcome the dynamics introduced by inter-cell interference, two matched VUEs could exchange their matched RBs $\Phi_k^{k'} = \{\Phi\{(k, \Phi(k)), (k', \Phi(k'))\}\} \cup \{(k, \Phi(k')), (k', \Phi(k))\}$ when the maximum communication latency could be reduced.

2) *CPU-Cycle Frequency Allocation*: Given \mathbf{a} , the problem (15) is apparently a convex optimization problem with respect to \mathbf{f} and is easy to be solved by the interior-point method.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, simulation results are provided to verify the sum satisfaction of cooperative perception in FVNETs. We deploy 2 RRHs, 1 F-AP, and 6 ~ 12 VUEs on a 1000-meter-long road. The whole road is equally divided into 100 blocks and the size for each block message is 6.4-kbits with 3-level region quadtree. For cooperative perception, each VUE requests a task to expand its sensory range by 150 ~ 500 m with $\mu = 130$ and $\tau_k^{\max} = 100 \sim 150$ ms. In addition, The

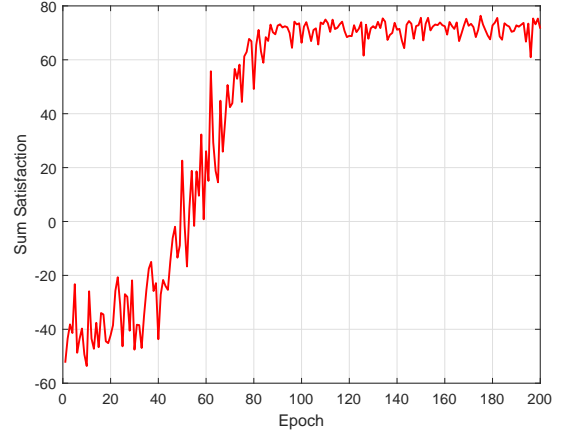


Fig. 2. The convergence of algorithm 1.

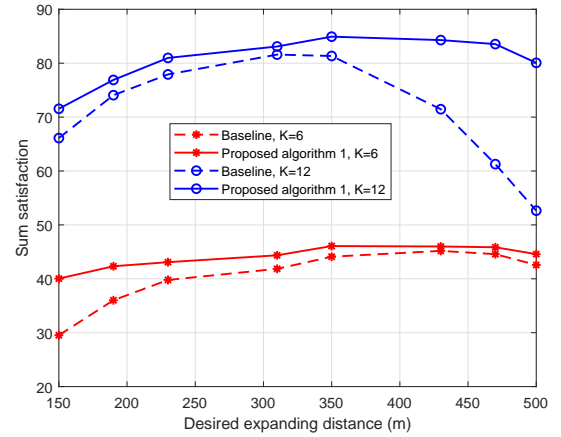


Fig. 3. Sum satisfaction versus desired expanding distance.

total bandwidth is 15 MHz, while the cloud server and the F-AP has a maximum computation resource of 30 GHz and 10 GHz, respectively.

Fig. 2 shows the convergence performance of the proposed attention multi-agent DDPG-based algorithm, when the number of VUEs is 12. It is seen that our proposed algorithm could effectively converge to stable sum satisfaction.

In Fig. 3, the sum satisfaction performances are provided with different desired expanding distances. For comparison, a distance-based mode selection and full message uploading scheme is adopted as the baseline. It can be observed that our proposed attention multi-agent DDPG algorithm outperforms the baseline, because it could select more valuable block messages and avoid undesired message uploading, meanwhile balancing the load among F-APs and the cloud server. Especially when the network load is large ($K = 12$), our proposed algorithm can effectively balance task load and constrained resource by controlling the selection of block messages.

Fig. 4 verifies the convergence of algorithm 2 with different numbers of VUEs, where the optimal latency which is derived

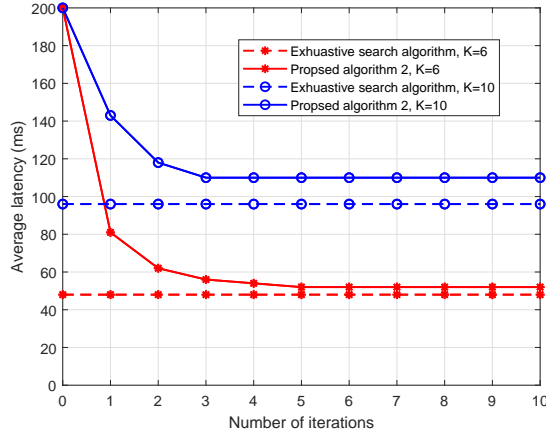


Fig. 4. The convergence of Algorithm 2.

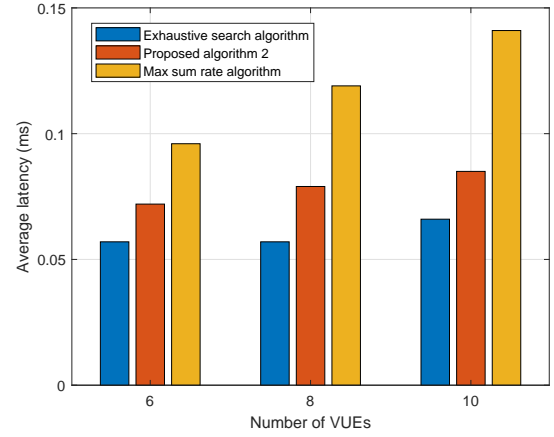


Fig. 5. Average latency versus the number of VUEs.

by exhaustive search is adopted for comparison. It can be seen that, our proposed swap matching algorithm can converge to a stationary point within 3 ~ 5 iterations, while its latency performance is close to exhaustive search algorithm.

Fig. 5 evaluates the average latency performances with different communication RB allocation schemes, when the numbers of VUEs are set as 6 ~ 10. For comparison, the exhaustive search and the matching-based sum rate maximization algorithms are selected as two baselines. Wherein, the former one offers the optimal average latency. It can be observed that our proposed algorithm outperforms the max sum rate algorithm and their latency gap becomes larger when the involved number of VUEs increases. That is because, on the one hand, the uploading latency of cooperative perception in (5) depends on the maximum one among all cooperative VUEs rather than overall latencies of all VUEs; on the other hand, the inter-cell interference has a larger effect on the VUE with lower channel gain. In addition, it is also observed that our proposed algorithm achieves considerable latency performance compared with exhaustive search algorithm.

V. CONCLUSION

In this article, we focus on a vehicular cooperative perception scenario with ultra-low latency requirement, and propose a joint sensing, communication, and computation resource allocation scheme with multi-agent DRL and swap matching for FVENTs, in which multiple VUEs constitute cooperative clusters to offload their computation tasks to either the cloud server or F-APs. Simulation results have verified the effectiveness and superiority of our proposed algorithms on the sum satisfaction and latency performances. In the future, it is interesting to incorporate radio sensing and investigate corresponding resource allocation for cooperative perception.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under No. 61921003 and

62001053, and the Fundamental Research Funds for the Central Universities under No. 248202020RC11 and 2020RC03.

REFERENCES

- [1] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020.
- [2] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 3, pp. 96–101, May 2021.
- [3] Y. Wang, G. de Veciana, T. Shimizu, and H. Lu, "Deployment and performance of infrastructure to assist vehicular collaborative sensing," in *Proc. IEEE VTC Spring*, Porto, Portugal, Jun. 2018, pp. 1–5.
- [4] E. E. Marvasti *et al.*, "Cooperative LIDAR object detection via feature sharing in deep networks," *ArXiv200208440 Cs*, Feb. 2020.
- [5] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Connected roads of the future: Use cases, requirements, and design considerations for vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 13, no. 3, pp. 110–123, Sep. 2018.
- [6] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, "Value-anticipating V2V communications for cooperative perception," in *Proc. IEEE IV*, Paris, France, Jun. 2019, pp. 1947–1952.
- [7] M. K. Abdel-Aziz, C. Perfecto, S. Samarakoon, M. Bennis, and W. Saad, "Vehicular cooperative perception through action branching and federated reinforcement learning," *ArXiv201203414 Cs*, Dec. 2020.
- [8] H. Du, S. Leng, K. Zhang, and L. Zhou, "Cooperative sensing and task offloading for autonomous platoons," in *Proc. IEEE GLOBECOM*, Taipei, Taiwan, Dec. 2020, pp. 1–6.
- [9] K. Zhang, M. Peng, and Y. Sun, "Delay-optimized resource allocation in fog-based vehicular networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1347–1357, Feb. 2021.
- [10] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6380–6391, Jul. 2020.
- [11] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proc. ICML*, California, USA, May 2019, pp. 2961–2970.
- [12] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3072–3108, 2019.

This figure "fig1.png" is available in "png" format from:

<http://arxiv.org/ps/2112.06224v1>