Dynamic Many-to-Many Task Offloading in Vehicular Fog Computing: A Multi-Agent DRL Approach

Zhiwei Wei[1], Bing Li[1], Rongqing Zhang[1], Xiang Cheng[2], Liuqing Yang[3,4]
[1]School of Software Engineering, Tongji University, Shanghai, China
[2]School of Electronics, Peking University, Beijing, China
[3]IoT Thrust & INTR Thrust, The Hong Kong University of Science & Technology (GZ), Guangzhou, China
[4]Department of Electronic & Computer Engineering, The Hong Kong University of Science & Technology, Hong Kong SAR
Email: {2031563, 1710056, rongqingz}@tongji.edu.cn, xiangcheng@pku.edu.cn, lqyang@ust.hk

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001342

***Abstract*—Confronted with the increasing computation-intensive requirements of vehicular applications, vehicular fog computing (VFC) has emerged as the promising solution to mitigate the load at the edge of vehicular network. In VFC, vehicles are employed as vehicular fog nodes to provide reliable services with applicability. However, considering the individual serving and offloading intentions of the privately-owned vehicles, the many-to-many task offloading in dynamic vehicular environment becomes a challenging problem. In this paper, we propose a distributed dynamic many-to-many task offloading framework based on vehicle-to-vehicle (V2V) trading paradigm to improve the fog resource utilization in VFC. In order to reach an effective and stable offload-and-serve cooperation between vehicles as service demanders and vehicles as computation providers in the proposed framework, we formulate the trading process as a partially observable Markov decision processes (POMDP) and design a Multi-Agent Gated actor Attention Critic (MA-GAC) approach, leading to an efficient offloading optimization process in a distributed manner. Theoretical analysis and experiments verify the feasibility and efficiency of the proposed framework, and simulation results demonstrate that the proposed MA-GAC approach outperforms other benchmarks in the dynamic environment.**

***Index Terms*—POMDP, task offloading, multi-agent deep reinforcement learning, many-to-many, vehicular fog computing.**

I. Introduction

Driven by the emerging vehicular applications, the increasing demands for the high-complexity but low-latency computation stimulate mobile edge/fog computing paradigm [1]. However, the limited resources of the stationary fog nodes (e.g., base stations, BSs and roadside units, RSUs) and the lack of service applicability led by vehicles' high mobility may hinder the improvement in vehicular applications [2], [3]. To further exploit edge computing capability and provide available resources with satisfactory quality-of-service (QoS), vehicular fog computing (VFC) has been proposed and greatly mitigated the edge computing burden.

In VFC paradigm, intelligent vehicles are employed as vehicular fog nodes to enable more flexible share of services at the edge of vehicular network [4], [5]. Nonetheless, the utilization of vehicular resources brings about a number of issues on computational offloading. In [6], Zhu *et al.* formulated the task allocation process to fog nodes as a bi-objective minimization problem and utilized traditional centralized mathematical tools to solve it. Considering the influence of task execution order on the QoS, reference [7] determined the execution order of tasks through a heuristic method among multiple users and servers. To motivate vehicles to share their computational resources, Shi *et al.* in [8] proposed a centralized task offloading scheme with pricing mechanism and designed a deep reinforcement learning (DRL) approach to maximize the expected reward of vehicular fog nodes, and the authors in [9] proposed a centralized multiattribute-based double auction mechanism in VFC for resource bargaining. However, the existing literatures do not provide an efficient and scalable solution to the many-to-many task offloading problem where multiple vehicles as service demanders and multiple vehicles as computation providers coexist, taking each vehicle's individual intention into consideration. Moreover, considering the dynamics in both computation and communication information of vehicular network, the centralized methods [7]–[9] may not be scalable to achieve the optimal offloading decision with robustness.

Motivated by the above-mentioned challenges, in this paper, we investigate the dynamic many-to-many task offloading problem with the consideration of vehicles' individual rationality. To efficiently model and solve the complicated many-to-many task offloading problem, we employ a vehicle-to-vehicle (V2V) computational resource trading paradigm and propose a distributed V2V-trading-based many-to-many task offloading framework. In the proposed framework, vehicles as service demanders and vehicles as computation providers are motivated to cooperate with each other for higher individual utilities in a more flexible and efficient manner subject to two-sided cooperation willingness. Coalitional game and mid-market-rate (MMR) pricing mechanism are applied to formulate the stable cooperation relationships among vehicles. In order to fully exploit the local computing potentials, we formulate the trading process as a partially observable Markov decision processes (POMDP) and design a Multi-Agent Gated actor Attention Critic (MA-GAC) approach to reach an effective and stable cooperation deal between service demanders and provides. Theoretical analysis and simulation results demonstrate the feasibility and effectiveness of the proposed V2V trading framework, and the proposed MA-GAC is verified to

978-1-6654-3540-6/22 © 2022 IEEE

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.

outperform other benchmarks.

The rest of this paper is organized as follows. Section II introduces the system models and formulates the task offloading problem as an optimization problem. Section III proposes a task offloading framework based on V2V trading paradigm. Section IV proposes the MA-GAC approach and Section V verifies the performance with other benchmarks. Section VI concludes this paper.

II. System Model and Problem Formulation

A. System Model

Vehicle set is denoted as $\mathcal{V} = \{V_1, V_2, \cdots, V_N\}$ and the *individual intention* is expressed as $\mathcal{L} = \{l_{1,t}, l_{2,t}, \cdots, l_{N,t}\}$ in each time slot t where $l_{i,t} > 0$ means that the vehicle requests computational resources and $l_{i,t} < 0$ presents the willingness to share resources.

1) Mobility Model: Each vehicle V_n is characterized by three attributes: velocity (v_n), direction ($dr_n \in \{-1, 1\}$), and location (loc_n). Similar to [8], we refer to a free flow traffic model and suppose that all the vehicles drive at a constant speed following Gaussian distribution. We explain the mean $\bar{v}$ and the variance σ_v of velocity as:

$$\bar{v} = v_{\max}\left(1 - \frac{N}{N_{\max}}\right), \qquad \sigma_v = \alpha_v \bar{v} \tag{1}$$

where α_v is the scaling parameter, N is the number of vehicles, and $N_{\max}$ is the maximum capacity of vehicles of the road.

2) Task Model: For each vehicle V_n, the generated task at time slot t is characterized by the tolerant service latency $\delta_{n,t}$, the calculated condition $\zeta_{t,t'}$, the required cpu frequency $cr_{n,t} \propto \zeta_{n,t,t'}$, the upload/download data size $up_{n,t}/dw_{n,t}$, the delay sensitive parameter ε_n, and the basic task utility $u_{n,t}$. We design the task utility reflecting QoS as:

$$U_n = \begin{cases} \dfrac{\delta_n}{T_n^{total}} \times u_n, & T_n^{total} \geq \delta_n \\ u_n + \varepsilon_n \log\left(1 + \delta_n - T_n^{total}\right), & T_n^{total} < \delta_n \end{cases} \tag{2}$$

where T_n^{total} is the service delay of the current task. If the task cannot be accomplished within the tolerant delay, the utility is obtained at a discount; otherwise, the utility increment enjoys a logarithmic relation to the reduced latency. The total service latency is $T_n^{total} = T_n^c + \max_{V_{n'} \in \mathcal{V}}\left(T_{n,n'}^c + T_{n,n'}^t\right)$ where T_n^c is the computing latency and $T_{n,n'}^t$ is the transmission latency.

3) Communicating and Computing Model: The V2V channel model is constructed with the assumption that each vehicle is allocated an orthogonal spectrum resource block. The transmission data rate is given by:

$$r_{n,n'} = B_{n,n'} \log\left(1 + \frac{P_{n,n'}^t d_{n,n'}^{-\alpha_p} |h_{n,n'}|^2}{N_0}\right) \tag{3}$$

where $B_{n,n'}$ is the bandwidth, $P_{n,n'}^t$ is the transmission power, $d_{n,n'}$ is the distance, α_p is the path-loss exponent, N_0 is the power noise, and $h_{n,n'}$ is the Rayleigh channel coefficient. The communication delay and computing delay for vehicle $V_{n'}$ to tackle the task from V_n is thus given by:

$$T_{n,n'}^t = \frac{(dw_n + up_n)}{r_{n,n'}}, \qquad T_{n,n'}^c = \frac{cr_n}{f_{n',n}} \tag{4}$$

where $f_{n',n}$ is the devoted CPU resources from $V_{n'}$ to V_n.

4) External Cost Model: Suppose that computational requirements are served with a negotiated price for the appointed resources. The total buy and sell pricing of resources for each vehicle is termed as *external cost*, and the external cost model for vehicle V_n to buy or sell from $V_{n'}$ is given by:

$$ext(l_{n,n'}) = p_n^b \left[l_{n,n'}\right]^+ + p_n^s \left[l_{n,n'}\right]^- \tag{5}$$

where $l_{n,n'}$ is the amount of computational resources between $V_{n'}$ and V_n as serving intention, p_n^b and p_n^s are the unit prices for buy and sell, and the operators $[\cdot]^{+/-}$ indicate to take the maximum or minimum value between $\cdot$ and zero.

B. Many-to-Many Task Offloading Problem Formulation

The dynamic many-to-many task offloading problem is coupled with the offloading decisions, individual tradeoff between external cost and task utility, and resource scheduling strategies. The offloading decision for each vehicle V_n at time slot t is formulated as an optimization problem in a long run:

$$\begin{aligned} \max_{\mathbf{l},\mathbf{f}} \quad & \sum_{t=1}^{T} [U_{n,t} - \omega \sum_{V_{n'} \in \mathcal{V}} ext(l_{n,n',t})] \\ \text{s.t.} \quad & C1: \sum_{V_{n'} \in \mathcal{V}} \left[l_{n,n',t}\right]^- \geq \left[l_{n,t}\right]^-, \forall t \in [1, T] \\ & C2: T_{n,n'}^{total} \leq \delta_{n',t}, \forall V_{n'} \in \mathcal{V}, \forall t \in [1, T] \\ & C3: f_{n,n',t} \in [[l_{n,n',t}]^-, f_{n,t}], \forall V_{n'} \in \mathcal{V}, \forall t \in [1, T] \\ & C4: \sum_{V_{n'} \in \mathcal{V}} f_{n,n',t} \leq f_{n,t}, \forall t \in [1, T] \end{aligned} \tag{6}$$

where ω is the scaling factor to tradeoff the importance of external cost and task utility. $C1$ guarantees the shared resources to be within the computational capability; $C2$ is the tolerant latency constraint that guarantees the offloaded tasks being calculated within deadlines; $C3$ and $C4$ are the constraints on the ranging of computational resource allocation. The program (6) is modelled from the perspective of individual vehicle, and it is NP-hard to address an optimized solution for all vehicles jointly. Moreover, considering the temporal influences of the offloading decisions, the problem becomes even more complicated to solve.

III. A Distributed Dynamic Many-to-Many Task Offloading Framework

A. Distributed Task Offloading Based on V2V Trading

As shown in Fig. 1, we propose a distributed many-to-many task offloading framework based on V2V computational resource trading paradigm in VFC. In the VFC architecture, the cloudlet layer consists of multiple service zones and the BSs in different service zones are connected with the neighboring ones through wired links. In such context, the BS has the ability to

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.

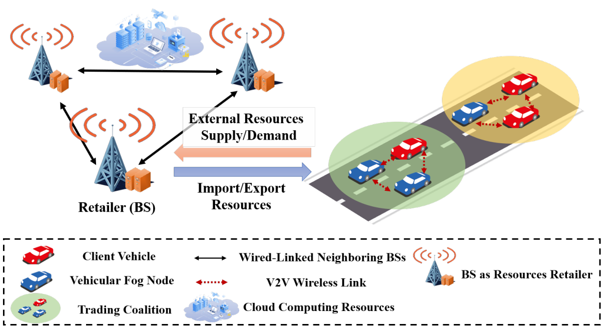


Fig. 1. The V2V computational resource trading paradigm in VFC.

achieve inter-region task offloading and takes the role as the resource retailer in the proposed V2V trading paradigm.

Nonetheless, there are two main factors that directly affect the framework feasibility and efficiency: the poor wireless communication links and the trading unit price determination agreement. To tackle those two issues, the cooperation among vehicles is adopted as a feasible way to operate the resources with stability and efficiency. Vehicles are motivated to form coalitions to operate the resources cooperatively and transmit data with reliability. To achieve the stable coalitional structure and avoid the dynamics resulted by the changing topology of vehicles, a cooperative coalitional game is proposed and MMR pricing [12] is utilized as the revenue allocation strategy in the coalitional core.

B. Coalitional Game

A coalitional game $\mathcal{G}$ in resource trading framework contains the vehicle set $\mathcal{V}$, the computational load $\mathcal{L}$, and the value function $v(\cdot)$ which projects coalition utility to a scalar. Coalitional game is introduced to allow vehicles to operate computational and communicating resources cooperatively. For coalition $C_i = \{V_{n,i}\}_N$, we design the external cost as:

$$ext(C_i) = p_t^b \left[\sum_{n=1}^{N} l_{n,t}\right]^+ + p_t^s \left[\sum_{n=1}^{N} l_{n,t}\right]^- \tag{7}$$

where p_t^b and p_t^s ($p_t^b > p_t^s$) are the buying and selling unit price provided by the BS. The external cost reflects the capability to cooperatively deal with the internal demands. Therefore, the value function $v(C_i)$ is given by:

$$v(C_i) = \sum_{s=1}^{S} ext\left(\{V_{s,i}\}\right) - ext(C_i). \tag{8}$$

The designed value function quantifies the cost saving for both buyers and sellers by deriving the difference between the cost trading with the BS and the one trading with each others. Based on the designed value function, vehicles are motivated to formulate a grand cooperative coalition [12]. We further assume that vehicles form stable coalitions for any two vehicles V_n and $V_{n'}$ during the computational resource trading period T^{trade} given distance D_0: $\left\|loc_n\left(t+T^{trade}\right) - loc_{n'}\left(t+T^{trade}\right)\right\|_2 \leq D_0$.

To find the core of the proposed coalitional game for the coalition structure stability, a proper pricing mechanism should be introduced as the revenue allocation strategy in each coalition [11]. We propose to use MMR pricing mechanism because of its simplicity and feasibility.

C. Mid-Market-Rate Pricing Mechanism

Denote the local demand and supply of the coalition as $D = \sum_{V_i \in C} [l_i]^+$ and $S = -\sum_{V_i \in C} [l_i]^-$. The MMR method set the local buying and selling prices under three specific conditions: *1)* If $D = S$, the local buy price and sell price are set to be the mid price of the BS $p_{L,t}^b = p_{L,t}^s = p_t^{mid} = \frac{p_t^b + p_t^s}{2}$. *2)* If $D > S$, the deficit computational resources are bought from the BS, and the extra payment is proportionally shared by buyers $p_{L,t}^b = \frac{p_t^{mid} S + p_t^b (D-S)}{D}$, $p_{L,t}^s = p_t^{mid}$. *3)* If $D < S$, the local unit price is defined as:$p_{L,t}^b = p_t^{mid}$, $p_{L,t}^s = \frac{p_t^{mid} D + p_t^s (S-D)}{S}$.

IV. Multi-Agent Deep Reinforcement Learning-Based Trading Strategy

A. POMDP Formulation

In order to fully exploit the local computing potentials, we apply a multi-agent DRL (MADRL) approach to solve the strategic trading.

1) Observation: The observation $o_{n,t}$ for each vehicle V_n contains the common features including time step t, local sell/buy price $p_{L,t}^s/p_{L,t}^b$ and supplying/demanding amount of resources $n_{L,:t}^s/n_{L,:t}^b$ history with length ι, current computing frequency $f_{i,t}$ of all the vehicles, current local load $\zeta_{i,t}$ of all the vehicles, current basic task utility $u_{n,t}$, and the vehicle delay sensitive parameter ε_n:

$$o_{n,t} = [t, p_{L,:t}^s, p_{L,:t}^b, n_{L,:t}^s, n_{L,:t}^b, \{f_{i,t}\}, \{\zeta_{i,t}\}, u_{n,t}, \varepsilon_n] \tag{9}$$

and the global state space is $O_t = \{o_{1,t}, o_{2,t}, \cdots, o_{N,t}\}$.

2) Action: We use $x_{n,t} \in \{0, 1\}$ to represent the participating willingness and $l_{n,t}$ as the individual intention to sell or buy. The action for V_n at time slot t is given by:

$$a_{n,t} = [x_{n,t}, l_{n,t}] \tag{10}$$

and the global action space is $A_t = \{a_{1,t}, a_{2,t}, \cdots, a_{N,t}\}$.

3) Reward: The reward is designed as the task utility minus the basic task utility $u_{n,t}$ and the weighted external cost:

$$R_{n,t} = U_{n,t} - u_{n,t} - \omega\left(p_{L,t}^b \left[l_{n,t}\right]^+ + p_{L,t}^s \left[l_{n,t}\right]^-\right). \tag{11}$$

B. MA-GAC: Multi-Agent Gated Actor Attention Critic

1) Policy and Value Function: The proposed MA-GAC approach is based on the design of soft actor critic (SAC) method. We denote the critic network as $Q^\theta(O_t, A_t)$ and actor network as $\pi^\psi(a_{n,t}|o_{n,t})$ where θ and ψ are the corresponding network parameters. To maximize not only the long-term reward but also the selection entropy, the update target is given by:

$$L(\theta) = \mathop{\mathbb{E}}_{(O_t, A_t, r_t, O_{t+1}) \sim \mathcal{D}} \left[(Q^\theta(O, A) - y)^2\right] \tag{12}$$

where $\mathcal{D}$ is the replay buffer, y is the next step estimated value and $Q^{\bar{\theta}}$ is the target Q function. y is given as:

$$y = r_t + \gamma \mathbb{E}\left[Q^{\bar{\theta}}(O_{t+1}, A_{t+1}) - \alpha \log(\pi^{\bar{\phi}}(A_{t+1}|O_{t+1}))\right] \tag{13}$$

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.

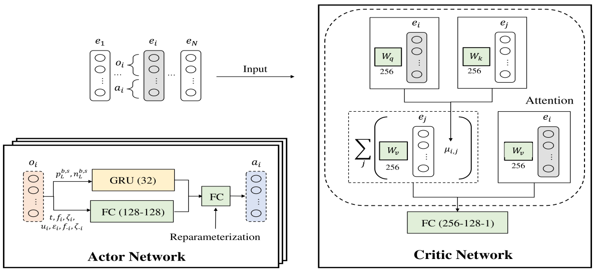


Fig. 2. Actor network and critic network in the proposed MA-GAC approach.

where α is the temperature parameter to weighing the importance of policy entropy and $\pi^{\bar{\phi}}$ is the target policy network. Target network is soft updated by the current network. The critic network can be trained by directly minimizing $L(\theta)$, while the policy parameters ϕ are updated by taking the performance gradient $\nabla_{\phi}J(\pi^{\phi})$ as:

$$J(\pi^{\phi}) = \mathop{\mathbb{E}}_{(O,A)\sim\mathcal{D}}\left[\alpha\log(\pi^{\phi}(A'|O) - Q^{\theta}(O,A') + Q^{\theta}(O,A))\right] \tag{14}$$

where A is the action taken in the replay buffer and A' is the estimated action according to π^{ϕ}.

2) Attention Mechanism in Critic Network: To promote the robustness of the proposed approach faced with uncertain number of agents, attention mechanism [2] is employed to the critic network Q^{θ} as shown in Fig. 2. The attention layer takes the action and observation as embeddings and trains three matrices (value W_v, query W_q, and key W_k) to obtain the attention weight of each embedding:

$$att_i = \sum_{j\neq i}\mu_{i,j}h(W_v e_j), \quad \mu_{i,j} = \frac{\exp^{(W_k e_j)^T W_q e_i}}{\sum_j \exp^{(W_k e_j)^T W_q e_i}} \tag{15}$$

where att_i is the attention to other embeddings, $\mu_{i,j}$ is the attention weight and $h(\cdot)$ is the activation function. The attention layer is shared among different agents.

3) GRU Network: We merge GRU into the actor network as the hidden layer to help make trading decisions more wisely as shown in Fig. 2. GRU is chosen in our approach because it performs well in sequence analyzing and enjoys a cheap computational cost.

In the proposed MA-GAC method, the centralized training with decentralized execution (CTDE) framework is utilized. The training process of the proposed MA-GAC approach is shown in Algorithm 1. In order to avoid overestimating Q-value in the training process, we apply twin critic network in the proposed MA-GAC approach and use the minimum of the network outputs as the certain Q-value.

Algorithm 1 Training of MA-GAC Approach

```
Initialize the environment with $N$ agents.
for each training episode do
  Reset environment.
  for time step $t = 1 : T$ do
    Get observation $O_t$ from the environment.
    Select an action from $a_{i,t} \sim \pi^{\theta}(o_{i,t})$.
    Each agent execute the action and get the changed observation $\tilde{o}_{i,t}$ as well as the reward $R_{i,t}$.
    Store the experiences $(O_t, A_t, R_t, \tilde{O}_t)$ into $\mathcal{D}$.
    if start to train then
      Retreive a minibatch of experiences from $\mathcal{D}$.
      Evaluate $Q_i^{\theta_{1,2}}(O_t, \pi^{\psi}(O_t))$, $Q_i^{\theta_{1,2}}(O_t, A_t)$, $Q_i^{\bar{\theta}_{1,2}}(\tilde{O}_t, \tilde{A}_t)$, $\tilde{a}_{i,t} \sim \pi_i^{\bar{\psi}}(\tilde{o}_{i,t})$ for each agent.
      Update current critic network with $\nabla_{\theta}L(\theta)$ in (12).
      Update current actor network with $\nabla_{\psi}J(\psi)$ in (14).
      Soft update target network with $\bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta$, $\bar{\psi} \leftarrow \tau\bar{\psi} + (1-\tau)\psi$.
    end if
  end for
end for
```

V. Simulation and Results

A. Simulation Setup and Implementation

TABLE I
Simulation Parameters

Vehicular one-hop distance D_0	100 m
Agent number $\lvert\mathcal{V}\rvert$	50
Maximum vehicular velocity $v_{\max}$	20 m/s
Maximum road capacity $N_{\max}$	100
Vehicular CPU frequency f	$[2, 6]$ GHz
Vehicular calculating condition ζ	$[3, 5]$ GHz
Delay sensitive parameter ε	$[0, 0.1]$
Basic task utility u	$[1, 2]$ $(\mathrm{GHz}\cdot\mathrm{s})^{-1}$
Task deadline δ	0.1 s
Task download data size dw	$[0.02, 0.2]$ MB
Task upload data size up	$[0.05dw, 0.1dw]$ MB
Task computational requirement cr	$[0.75, 1.25]\zeta\Delta t$ Giga-cycles
Trading time T^{trade}	0.5 s
Trading num for each buyer/seller	$[0.5, 3]$ GHz/s
Retailer buy unit price p_t^b	$[0.029, 0.053]$ GHz^{-1}
Retailer sell unit price p_t^s	$[0.008, 0.023]$ GHz^{-1}
Noise power N_0	-104 dBm
Total bandwidth B	20 MHz
Time slot length Δt	0.1 s

We construct the environment via PYTHON and suppose a two-lane road in the service zone with 500 meters coverage radius. The BS is located at $(500, 0)$. Vehicles are simulated according to the system models described in Subsection II-A. Tasks are generated in each time slot with the computational requirement positive related to the calculating condition ζ. From the perspective of reality, an acceptable fluctuation of computational requirement of about 75% ~ 125% $\zeta\Delta t$ is adopted in the stochastic computing environment. The pricing history length $\iota = 10$ and we further refer to the price design in [14] to set the sell and buy price of the BS as $p_t^s = \lambda_1(E_t^s)^2 + \lambda_2(E_t^s)^2$ and $p_t^b = \lambda_1(E_t^b)^2 + \lambda_2(E_t^b)^2$, where E_t^s and E_t^b are the quantified desires towards available resources and tasks in the global network, and λ_1 and λ_2 are the pricing parameter. For simplicity, a simple sinusoidal is utilized to simulate the pricing fluctuation within the range of $E_t^b \in [0.8, 1.2]$, $E_t^s \in [0.3, 0.7]$, and the pricing parameters are set as $\lambda_1 = \lambda_2 = 0.2$.

We choose 5 alternative methods as benchmarks.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.

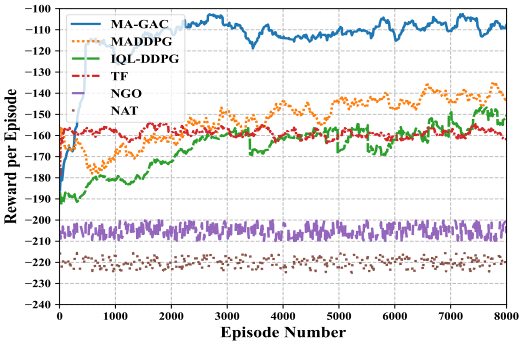


Fig. 3. The overall reward per epoch for all the approaches.

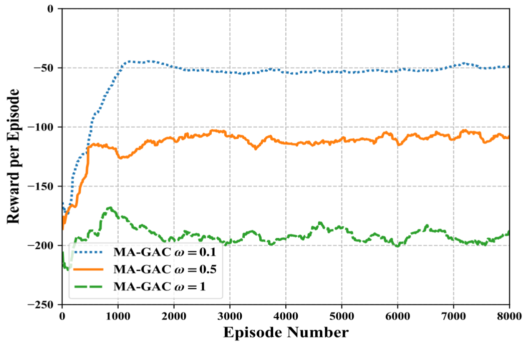


Fig. 4. The overall reward per epoch for MA-GAC with different ω.

(1) *MADDPG*: The MADDPG method [15] is a typical MADRL approach with deterministic policy.

(2) *Individual Q-Learning*: The individual Q-learning (IQL) learns the single-agent process and views the interactions of other agents as part of the environment.

(3) *Trading Forward*: Trading forward (TF) method denotes the circumstance in which the agents trade resources instantly regardless of price.

(4) *Naive Greedy Offloading*: In the naive greedy offloading (NGO) method, client vehicles are deemed to be served by the nearby vehicular fog nodes voluntarily. The client vehicles offload tasks greedily to the vehicular fog nodes that calculate tasks with the shortest service latency.

(5) *Not-Attend-Trading*: The agents in the not-attend-trading (NAT) method mean not to trade with each others and even not to offload tasks.

The implementation of the MA-GAC network is listed as follows. We set the episode length as 50, soft updating parameter $\tau = 0.01$, learning rate for actors as $lr_a = 1 \times 10^{-4}$, learning rate for the critic as $lr_c = 5 \times 10^{-4}$, discount factor $\gamma = 0.97$, batch size as 256, and memory size as 1×10^5. We use leakyReLUs as activation units and choose AdamW as the optimizers. The detailed network structure has been demonstrated in Fig. 2 and Table I shows other parameters.

B. Performance Evaluation

Fig. 3 shows the convergence of the overall reward (i.e., social welfare) among all the training approaches. It is observed that the deterministic DRL approaches MADDPG and IQL-DDPG is not suitable in the stochastic V2V task offloading environment and thereby converges slowly. Specifically, the IQL method deems the other agents as part of the environment, which causes the instability in learning the environment and performs worse than the other MADRL-based approaches. The proposed MA-GAC converges after 2000 episodes and achieves about −100 overall reward for each episode. For comparison, the MADDPG method converges in about 5000 episodes with the overall reward −140, and the IQL-DDPG method falls into the premature optimal point early in 3000 episodes with −160 reward, which performs similarly to the TF approach. The NGO and NAT policies perform the worst among all the methods with the consideration of external costs.

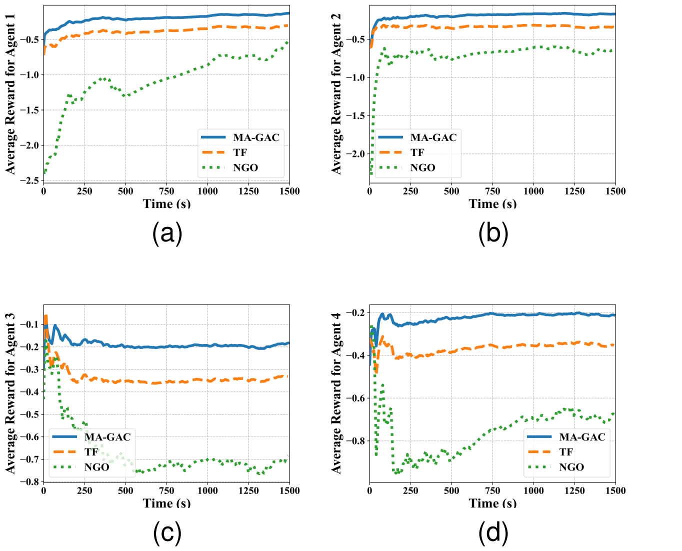
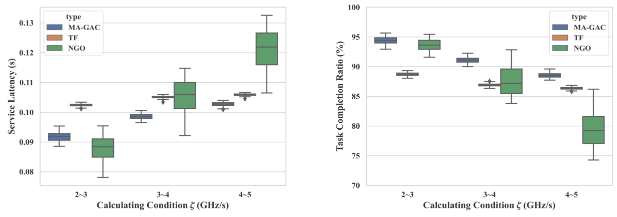


Fig. 5. Personal reward including external cost gained by arbitrary four agents during 1.5×10^3 seconds. (a)~(d) Agent 1~Agent 4.

In Fig. 4, we compare the performance of the proposed MA-GAC under different scaling factor ω. The factor ω is set to 0.1, 0.5, and 1 respectively to show the vehicles' intention to monetary revenues and external costs. Obviously, when ω is set to be small, the vehicles are more indifferent to the gain and of monetary currency and thereby tend to possess computational resources instead of share them for the sake of local task utility. Meanwhile, when ω is large, the vehicles despise their QoS of tasks and make a difference to sell the computational resources for more revenues. Statistics in Fig. 4 display that the proposed MA-GAC enables to learn suitable policy under different conditions with the changing ω.

Furthermore, the personally gained reward comparison among the MA-GAC, TF, and NGO methods are presented in Fig. 5 with the same random seeds. Four agents are chosen randomly from the simulation process and we compare the long term average reward during 1.5×10^3 seconds. As a result, the NGO method neglects the external cost to offload the tasks and

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 6. The box plot of the average service latency and task completion ratio under different calculating conditions.

thereby demonstrates an unsteady performance. On the other side of the coin, the agents are prompted to choose the trading-based approach for higher revenues, especially to enjoy the high profits earned by using the coordinated MA-GAC method. MA-GAC is verified to benefit best for any single agent in the term of computational resource trading.

As shown in Fig. 6, we modify the range of calculating conditions to simulate the cases where vehicles are running on the lower load, the medium load, and the higher load. It can be inferred from Fig. 6a that NGO achieves the lowest service latency with the mean of 89 milliseconds. The reason comes out to be that the vehicles using NGO approach devote their computational resources wholely to the task when the computational burden is low. Nonetheless, that the idle computational resources can be sold to the retailer BS for more benefits. The proposed MA-GAC method makes a tradeoff to maintain the service latency at about 90 milliseconds taking both the monetary benefits and the QoS of users into account and MA-GAC guarantees to finish the tasks with lower variance. A similar conclusion can also be deduced from Fig. 6b, in which the tasks are completed at a prominent 95% completion ratio with the calculating burden $\zeta \in [2,3]$ by MA-GAC and 89% completion ratio when the calculating condition falls in the range of $\zeta \in [4,5]$, which is nearly 10% higher than NGO.

VI. Conclusion

In this paper, we formulated the dynamic task offloading problem among vehicles in VFC with the consideration of each vehicle's individual rationality. We proposed a distributed many-to-many task offloading framework based on V2V computational resource trading. Coalitional game and MMR pricing mechanism were applied to allow vehicles to organize the steady and well-defined coalitions for resource orchestration. Then, a POMDP was formulated to model the multi-agent decision-making process in the proposed framework. We designed a MADRL approach termed MA-GAC to reach an effective and stable cooperation deal between service demanders and provides in a coordinated manner. Numerical results verified that the proposed MA-GAC outperforms other benchmarks in dynamic environment.

VII. Acknowledgment

This work is supported in part by the National Key R&D Program of China (Grant No. 2021ZD0112700), in part by the National Natural Science Foundation of China under Grant 61901302 and 62125101, in part by the Natural Science Foundation of Shanghai under Grant 20ZR1462400 and 22ZR1463600, in part by Chenguang Program supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission, in part by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities, and is also funded by State Key Laboratory of Advanced Optical Communication Systems Networks, China. The corresponding author is Rongqing Zhang.

References

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[2] J. Wu, X. Cheng, X. Ma, W. Li, and Y. Zhou, "A Time-Efficient and Attention-Aware Deployment Strategy for UAV Networks Driven by Deep Reinforcement Learning," in *Proc. IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 01-05.

[3] X. Cheng, R. Zhang, and L. Yang, "Wireless Toward the Era of Intelligent Vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 188-202, Feb. 2019.

[4] Z. Wei, B. Li, R. Zhang, and X. Cheng, "Contract-Based Charging Protocol for Electric Vehicles with Vehicular Fog Computing: An Integrated Charging and Computing Perspective," *IEEE Internet of Things Journal*, 2022.

[5] R. Zhang, R. Lu, X. Cheng, N. Wang, and L. Yang, "A UAV-Enabled Data Dissemination Protocol with Proactive Caching and File Sharing in V2X Networks," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3930-3942, Jun. 2021.

[6] C. Zhu *et al.*, "Folo: Latency and Quality Optimized Task Allocation in Vehicular Fog Computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4150-4161, Jun. 2019.

[7] J. Sun *et al.*, "Joint Optimization of Computation Offloading and Task Scheduling in Vehicular Edge Computing Networks," *IEEE Access*, vol. 8, pp. 10466-10477, 2020.

[8] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, "Priority-Aware Task Offloading in Vehicular Fog Computing Based on Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16067-16081, Dec. 2020.

[9] X. Peng, K. Ota, and M. Dong, "Multiattribute-Based Double Auction Toward Resource Allocation in Vehicular Fog Computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094-3103, Apr. 2020.

[10] L. Zou, M. S. Munir, Y. K. Tun, S. Kang, and C. S. Hong, "Intelligent EV Charging for Urban Prosumer Communities: An Auction and Multi-Agent Deep Reinforcement Learning Approach," *IEEE Transactions on Network and Service Management*, doi: 10.1109/TNSM.2022.3160210.

[11] W. Tushar *et al.*, "Peer-to-Peer Energy Trading With Sustainable User Participation: A Game Theoretic Approach," *IEEE Access*, vol. 6, pp. 62932-62943, 2018.

[12] J. Li, Y. Ye, D. Papadaskalopoulos, and G. Strbac, "Computationally Efficient Pricing and Benefit Distribution Mechanisms for Incentivizing Stable Peer-to-Peer Energy Trading," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 734-749, 15 Jan.15, 2021.

[13] Y. Ye, Y. Tang, H. Wang, X. -P. Zhang, and G. Strbac, "A Scalable Privacy-Preserving Multi-Agent Deep Reinforcement Learning Approach for Large-Scale Peer-to-Peer Transactive Energy Trading," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5185-5200, Nov. 2021.

[14] C. P. Mediwaththe and D. B. Smith, "Game-Theoretic Electric Vehicle Charging Management Resilient to Non-Ideal User Behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3486-3495, Nov. 2018.

[15] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent Deep-Reinforcement-Learning-Based Resource Allocation for Heterogeneous QoS Guarantees for Vehicular Networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1683-1695, 1 Feb.1, 2022.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on February 11,2023 at 07:54:06 UTC from IEEE Xplore. Restrictions apply.