

无人机辅助车联网边缘计算卸载和资源分配方案

岳文静¹, 李可¹, 陈志²

(1. 南京邮电大学通信与信息工程学院, 江苏 南京 210003; 2. 南京邮电大学计算机学院, 江苏 南京 210003)

摘要: 为解决地面边缘服务器灵活性差以及高密度交通要道服务器计算能力不足以提供可靠卸载服务的问题, 提出了一种无人机辅助的车联网边缘计算系统。该系统将多架无人机作为移动基站, 以辅助地面边缘服务器为车辆提供卸载服务。系统将计算卸载和资源分配的联合优化问题建模为一个混合整数非线性规划问题, 目的是优化车辆的卸载决策和边缘设备的资源分配, 使系统总成本最小。为求解该问题, 提出了融合遗传算法和深度确定性策略梯度 (GADDPG, genetic algorithm and deep deterministic policy gradient) 算法进行求解, 该算法通过引入遗传算法预训练演员模块和评论家模块的学习率, 使其在车联网场景中具有更好的寻优能力。将 GADDPG 算法应用于求解计算卸载和资源分配问题, 仿真结果表明, 在综合考虑卸载时延和能耗的情况下, 与 4 种计算卸载和资源分配问题算法进行相比, 可使系统成本分别降低 18.2%、18.8%、30.7% 和 78.3%, 因此该算法对降低系统成本具有更优的效果。

关键词: 车联网; 移动边缘计算; 深度强化学习; 遗传算法

中图分类号: TN929; TP311

文献标志码: A

doi: 10.11959/j.issn.2096-3750.WLW24050

UAV-assisted edge computing offloading and resource allocation scheme for Internet of vehicles

Yue Wenjing¹, Li Ke¹, Chen Zhi²

1. School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

2. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract: To solve the problems of poor flexibility of ground-based edge servers and insufficient computing power of servers in high-density traffic corridors to provide reliable offloading services, a UAV-assisted edge computing system for the Internet of vehicles (IoV) was proposed, which uses multiple UAVs as mobile base stations to assist ground-based edge servers in providing offloading services for vehicles. The system models the joint optimization problem of computational offloading and resource allocation as a mixed-integer nonlinear programming problem to optimize the offloading decision of vehicles and the resource allocation of edge devices to minimize the total system cost. To solve the problem, an algorithm incorporating a genetic algorithm and a deep deterministic policy gradient was proposed, which has a better optimization finding capability in the telematics scenario by introducing a genetic algorithm to pre-train the learning rate of the Actor module and the Critic module. Simulation results show that CORA_GADDPG reduces the system cost by 18.2%, 18.8%, 30.7%, and 78.3% compared to four algorithms for computing offloading and resource allocation problems, respectively, when offloading delay and energy consumption are considered. Therefore, CORA_GADDPG has a superior reduction effect on system cost.

Key words: Internet of vehicles, mobile edge computing, deep reinforcement learning, genetic algorithm

0 引言

随着车联网技术的不断进步,涌现出了一大批新型车载应用,如自动驾驶、安全驾驶、实时视频监控等,这些应用在给人们出行提供极大便利的同时也会产生大量数据,而这些数据中存在许多对计算能力要求较高的计算密集型任务。车载设备有限的计算能力显然无法处理这些密集型计算请求。为了满足这些海量数据的计算需求,基于云服务的车联网系统被视为潜在的解决方案,它可以将密集型数据处理请求迁移到远程云中心进行处理^[1]。但是,云中心服务器往往与车载设备距离很远,这可能会引起高时延、高负载等问题,大大降低了用户体验。而移动边缘计算(MEC, mobile edge computing)因其具有近距离部署、高计算能力等特点,可以大大减少数据上传和结果回传的时间,满足了车联网场景对于低时延、低能耗的性能要求^[2]。

MEC的能力因其服务器固定的特点受到限制,相比于陆地移动基站,以负载MEC服务器的无人机(UAV, unmanned aerial vehicle)作为移动基站可提升边缘服务器的计算能力。UAV依靠其部署灵活、适应能力强等特点已被广泛应用于计算卸载和资源分配。文献[3]针对多UAV的网络模型,提出了一种联合任务调度和计算资源分配方案,旨在最小化任务的处理时延。文献[4]提出了一种基于UAV的物联网协同边缘计算网络,其中UAV可为本地设备提供计算服务,也可为附近小范围基站提供计算服务。文献[5]旨在联合优化UAV轨迹和用户调度来最小化用户时延,并提出了一种惩罚凹凸过程的算法进行求解。文献[6]提出了一种UAV辅助边缘计算框架,该框架联合优化了UAV轨迹、CPU频率以及卸载调度以最小化UAV的能量消耗。文献[7]联合优化UAV轨迹、任务划分以及多UAV场景下的计算和通信资源来最小化总加权能耗。以上研究仅侧重于单一UAV辅助边缘计算的方法,没有考虑协同地面边缘服务器共同辅助边缘计算。

UAV辅助车联网边缘计算可以最大程度地为车辆提供计算服务^[8-17]。文献[18]提出了UAV辅助地面边缘设备车联网网络系统,在有限的网络带宽和功率约束下联合优化UAV的飞行轨迹和卸载策略。但是该系统并没有考虑车辆任务卸载过程中的能耗,也没有考虑边缘设备计算资源对卸载决策

的影响。基于此,本文提出了一种UAV辅助车联网边缘计算卸载和资源分配方案。该方案将车辆与边缘设备的距离优先模型考虑在内,旨在最小化系统卸载时延和能耗的加权和。

针对该系统的优化问题,研究者根据不同的训练框架和交互策略,提出了许多优秀的深度强化学习(DRL, deep reinforcement learning)算法,如深度Q网络(DQN, deep Q-network)算法^[19],异步优势演员-评论家(asynchronous advantage actor-critic, A3C)算法^[20],深度确定性策略梯度(DDPG, deep deterministic policy gradient)算法^[21],以及近端策略优化(PPO, proximal policy optimization)算法^[22]。这些算法被广泛应用于边缘计算卸载和资源分配领域^[23-27]。然而,这些算法的性能通常依赖于学习率的设定。如果学习率设置过大,模型参数更新的步长会变得很大,导致模型在参数空间中来回振荡,甚至可能导致参数发散而无法收敛到最优解。相反,如果学习率设置过小,模型参数更新的步长会变得很小,导致模型收敛速度缓慢,需要更多的迭代次数才能达到收敛。为解决这一问题,本文提出了一种融合遗传算法和深度确定性策略梯度(GADDPG, genetic algorithm and deep deterministic policy gradient)算法。该算法通过遗传算法预训练学习率,将预训练后的最佳学习率应用到DDPG算法中以提高算法的优化能力。通过仿真实验可知,GADDPG算法可以有效解决UAV辅助车联网边缘计算的系统能耗和时延最小化问题。

本文的主要贡献如下。

1) 提出了一种适用于UAV辅助的车联网边缘计算卸载和资源分配方案,该方案将车辆与边缘设备的距离优先模型考虑在内,并且车辆的计算任务可以卸载到配备有边缘服务器的路边单元(RSU, road side unit)执行,也可以卸载到UAV上执行。

2) 为了最小化车联网卸载过程中能耗和时延的加权和,本文将卸载决策、计算资源和频谱资源的联合优化问题建模为一个混合整数非线性规划问题。

3) 将上述复杂的计算卸载和资源分配问题建模为一个马尔可夫决策过程,并基于DDPG框架设计了GADDPG算法。该算法通过引入遗传算法预训练演员(Actor)模块和评论家(Critic)模块的学习率,使其在车联网场景中具有更好的寻优能力。

1 系统模型

1.1 网络模型

图1展示了一种UAV辅助车联网边缘计算网络模型。

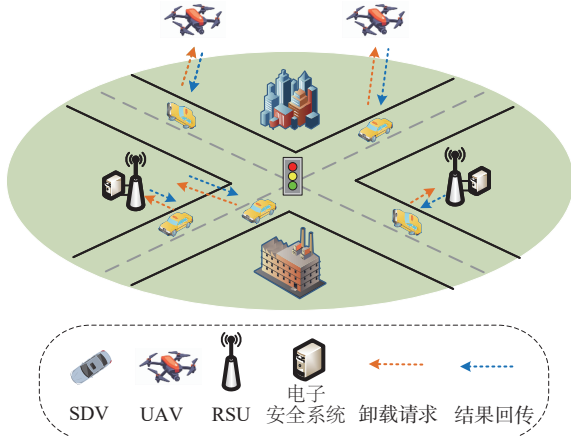


图1 UAV辅助车联网边缘计算网络模型

该模型主要包括3个实体，自动驾驶汽车 (SDV, self driving vehicle)、RSU和UAV。其中，RSU布置在道路两侧，每个RSU配备一个边缘服务器，为道路上的车辆提供计算或卸载服务。RSU位置固定，并且在高密度交通或城市区域易受信号干扰，导致通信质量下降。而UAV具有高机动性和灵活部署等特点，可以有效解决高密度场景中的信号强干扰问题。因此，本文在车联网系统中引入了配有高性能服务器的UAV，为SDV提供卸载服

$$d_u^e(t) = \|l_u(t) - l_e(t)\| =$$

$$\sqrt{(x_u(t) - x_e(t))^2 + (y_u(t) - y_e(t))^2 + (z_u(t) - z_e(t))^2}$$

其中， $z_u(t)$ 和 $z_e(t)$ 分别表示SDV和边缘设备在时隙 t 的高度。

考虑城市道路间存在障碍物遮挡，由SDV到RSU的无线传输速率 $R_{un}(t)$ 可以表示为

$$R_{un}(t) = b_{un}(t) \lg \left(1 + \frac{p_u h_{un}(t)}{f_k(t) P_{NLOS} + \sigma^2} \right) \quad (4)$$

其中， $b_{un}(t)$ 表示RSU分配给车辆 u 的带宽大小， p_u 表示SDV在上行链路的发射功率， σ^2 表示噪声功率， P_{NLOS} 表示传输损耗， $f_k(t)$ 表示在时隙 t 内，SDV和RSU之间是否存在阻塞标识符 (0表示没有阻塞，1表示阻塞)^[28]。

而UAV在空中飞行，与车辆间的障碍物较少，

务。SDV和RSU的集合分别为 $U = \{1, 2, \dots, U'\}$ 和 $N = \{1, 2, \dots, N'\}$ 。UAV的集合为 $V = \{1, 2, \dots, V'\}$ ，定义 $E = \{1, 2, \dots, e, \dots, E'\}$ 为边缘设备的集合，即 $E = \{1, 2, \dots, n, \dots, N', 1, 2, \dots, v, \dots, V'\}$ 表示 E 为RSU和UAV的集合总和，其中 $u \in U$ ， $n \in N$ ， $v \in V$ ， $e \in E$ 且集合 E 的最大值为 E' 。通常情况下，所搭建系统以时隙的形式工作，时隙被划分为离散时间帧 $t \in \{1, 2, \dots, T\}$ ，RSU和UAV可以处理来自SDV的计算请求，并且每个RSU或UAV可以为多个SDV提供卸载和资源分配服务。车辆在时隙 t 的位置为 $l_u(t) = (x_u(t), y_u(t), 0)$ ，RSU在时隙 t 的位置为 $l_n(t) = (x_n(t), y_n(t), 0)$ ，UAV在时隙 t 的位置为 $l_v(t) = (x_v(t), y_v(t), H)$ ， H 表示UAV的高度。

1.2 通信模型

在所构建的系统中，SDV经无线信道连接到RSU或UAV，并且可以在没有信道干扰的情况下卸载计算任务并下载所需要的结果。在每个计算任务的数据传输过程中，SDV与RSU和SDV与UAV的无线信道保持不变，SDV与RSU和SDV与UAV的无线链路信道增益 $h_{un}(t)$ 、 $h_{uv}(t)$ 可分别表示为

$$h_{un}(t) = \frac{h_0}{(d_u^n(t))^2} \quad (1)$$

$$h_{uv}(t) = \frac{h_0}{(d_u^v(t))^2} \quad (2)$$

其中， h_0 表示传输功率为1 W，传输距离为1 m时的信道功率增益； $d_u^n(t)$ 、 $d_u^v(t)$ 分别表示SDV与RSU和SDV与UAV的欧几里得距离。具体的计算公式为

$$(3)$$

因此不考虑障碍物对传输速率的影响，由SDV到UAV的无线传输速率可以表示为

$$R_{uv}(t) = b_{uv}(t) \lg \left(1 + \frac{p_u h_{uv}(t)}{\sigma^2} \right) \quad (5)$$

其中， $b_{uv}(t)$ 表示UAV分配给车辆 u 的带宽大小， p_u 表示SDV在上行链路的发射功率。

1.3 计算模型

假设每辆车 $u \in U$ 一次产生一个计算任务 $Q_u = \{c_u, d_u, r_u, T_u^b\}$ ， c_u 表示车辆任务的工作量，即完成任务所需要的CPU周期数， d_u 表示将程序执行从车辆传输到服务器所需要的输入数据量， r_u 表示由RSU或UAV返回给SDV的数据量， T_u^b 表示最大可

容忍的时延。在第 t 个时隙, 定义卸载策略 $x_u^e(t)$, $x_u^e(t)$ 为二进制变量, $x_u^e(t) = 1$ 表示车辆 u 的任务将被卸载到 UAV; $x_u^e(t) = 0$ 表示车辆 u 的任务被卸载到 RSU, 其中 $e \in E$ 。假设在一个时隙内, 每个 SDV 只能由一个 RSU 或 UAV 提供服务, 任务卸载策略变量可表示为

$$x_u^e(t) = \begin{cases} 1, & \text{第 } u \text{ 个车辆的任务卸载到第 } e - N \text{ 个 UAV} \\ 0, & \text{第 } u \text{ 个车辆的任务卸载到第 } e \text{ 个 RSU} \end{cases} \quad (6)$$

由于每个 RSU 或 UAV 的计算资源是有限的, 并且每个 RSU 或 UAV 可以为不同的 SDV 提供计算资源, 因此很难将全部计算资源分配给卸载请求任务的 SDV。第 v 个 UAV 或第 n 个 RSU 计算第 u 个 SDV 任务的时间 $T_{uv}^{\text{exc}}(t)$ 、 $T_{un}^{\text{exc}}(t)$ 可分别表示为

$$T_{uv}^{\text{exc}}(t) = x_u(t) \frac{c_u}{f_u^v(t)} \quad (7)$$

$$T_{un}^{\text{exc}}(t) = (1 - x_u(t)) \frac{c_u}{f_u^n(t)} \quad (8)$$

其中, $f_u^v(t)$ 表示在时隙 t , 第 v 个 UAV 分配给第 u 个 SDV 的瞬时计算资源; $f_u^n(t)$ 表示在时隙 t , 第 n 个 RSU 分配给第 u 个 SDV 的瞬时计算资源。

第 u 个 SDV 将任务上传到 UAV 或 RSU 的时延 $T_{uv}^{\text{up}}(t)$ 、 $T_{un}^{\text{up}}(t)$ 可分别表示为

$$T_{uv}^{\text{up}}(t) = x_u(t) \frac{d_u}{R_{uv}(t)} \quad (9)$$

$$T_{un}^{\text{up}}(t) = (1 - x_u(t)) \frac{d_u}{R_{un}(t)} \quad (10)$$

其中, $R_{uv}(t)$ 表示 UAV 与 SDV 经无线信道传输的平均速率, $R_{un}(t)$ 表示 RSU 与 SDV 经无线信道传输的平均速率。

UAV 或 RSU 将结果返回给第 u 个 SDV 的回传时间 $T_{uv}^{\text{ret}}(t)$ 、 $T_{un}^{\text{ret}}(t)$ 可分别表示为

$$T_{uv}^{\text{ret}}(t) = x_u(t) \frac{r_u}{R_{uv}(t)} \quad (11)$$

$$T_{un}^{\text{ret}}(t) = (1 - x_u(t)) \frac{r_u}{R_{un}(t)} \quad (12)$$

其中, r_u 表示由 UAV 返回给 SDV 或 RSU 返回给 SDV 的数据量。

1.4 移动性模型

SDV 需要提供高质量的车载服务来满足用户的需求, 然而 SDV 的高移动性对车联网系统的可靠计算卸载提出了巨大的挑战^[29]。因此, 在构建车联网系统时, 必须考虑 SDV 的动态移动性。假设第 u 个

SDV 在时隙 t 的位置为 $l_u(t) = (x_u(t), y_u(t), 0)$, 则下一个时隙的位置可表示为

$$\begin{cases} x_u(t+1) = x_u(t) + d_u \cos \theta_u \\ y_u(t+1) = y_u(t) + d_u \sin \theta_u \end{cases} \quad (13)$$

其中, d_u 表示第 u 个 SDV 在一个时隙内移动的距离, θ_u 表示 SDV 转动的角度。

UAV 具有高机动性和灵活部署等特点, 可以有效解决高密度场景中边缘设备计算能力不足的问题, 因此本文将 UAV 引入车联网场景中辅助 RSU 进行计算卸载。假设 UAV 飞行高度固定为 H , 第 v 个 UAV 从 $l_v(t) = (x_v(t), y_v(t), H)$ 到新的飞行位置可表示为

$$l_v(t+1) = (x_v(t+1), y_v(t+1), H) = (x_v(t) + d_v \cos \theta_v, y_v(t) + d_v \sin \theta_v, H) \quad (14)$$

其中, d_v 表示第 v 个 UAV 在一个时隙内移动的距离, θ_v 表示 UAV 转动的角度。

1.5 距离优先模型

在车联网场景中, 较低的时延对车辆的可靠性至关重要。由于时延与距离成正比关系, 在较短的距离范围内数据传输和处理能够显著降低时延, 加快响应速度。因此, 本文将在首次卸载时, 优先考虑距离车辆最近的 RSU 或 UAV 进行卸载。

SDV 与 RSU、SDV 与 UAV 的欧几里得距离可表示为式(3), 因此, 在初始时隙, 第 u 个 SDV 将选择距离最近的 UAV 或 RSU 进行卸载, 此时 $x_u(t) = x_u^e(t)$, 其中 e 表示距离最近的边缘设备。随着时间的推移, 如果有多个 SDV 竞争同一个 UAV 或 RSU 导致资源紧张时, 后续的 SDV 将摒弃距离优先原则, 而选择其他计算资源和可分配带宽充足的设备。

2 问题建模

假设本文不考虑 UAV 和车辆行驶过程中的能耗, 每个 SDV 在任务卸载过程中, 时隙 t 的系统总能耗 $E_u(t)$ 和任务执行延迟 $T_u(t)$ 可分别表示为

$$\begin{aligned} E_u(t) &= E_u^{\text{up}}(t) + E_u^{\text{exc}}(t) + E_u^{\text{ret}}(t) = \\ &= x_u(t) \sum_{v=1}^{V'} \left(\frac{d_u}{R_{uv}(t)} p_u + \frac{c_u}{f_u^v(t)} p_v + \frac{r_u}{R_{uv}(t)} p_v \right) + \\ &+ (1 - x_u(t)) \sum_{n=1}^{N'} \left(\frac{d_u}{R_{un}(t)} p_u + \frac{c_u}{f_u^n(t)} p_n + \frac{r_u}{R_{un}(t)} p_n \right) \end{aligned} \quad (15)$$

$$T_u(t) = T_u^w(t) + T_u^{\text{exe}}(t) + T_u^{\text{ret}}(t) = x_u(t) \sum_{v=1}^{V'} \left(\frac{d_u}{R_{uv}(t)} + \frac{c_u}{f_u^v(t)} + \frac{r_u}{R_{uv}(t)} \right) + (1 - x_u(t)) \sum_{n=1}^{N'} \left(\frac{d_u}{R_{un}(t)} + \frac{c_u}{f_u^n(t)} + \frac{r_u}{R_{un}(t)} \right) \quad (16)$$

其中, p_u 、 p_v 和 p_n 分别表示SDV的传输功率、UAV的平均消耗功率和RSU的平均消耗功率。

在本文中, 系统的总成本包括所有的SDV能耗和时延的加权和, 可表示为

$$\text{Cost}(X, B, F, L) = \sum_{t=1}^T \sum_{u=1}^{U'} [\psi \cdot T_u(t) + (1 - \psi) \cdot E_u(t)] = \sum_{t=1}^T \sum_{u=1}^{U'} [\psi \cdot (T_u^w(t) + T_u^{\text{exe}}(t) + T_u^{\text{ret}}(t)) + (1 - \psi) (E_u^w(t) + E_u^{\text{exe}}(t) + E_u^{\text{ret}}(t))] \quad (17)$$

其中, ψ 和 $(1 - \psi)$ 分别表示时延和能耗的加权因子, 可根据服务的需求和设备的性能来设置。

车辆卸载决策的集合可表示为 $X = \{x_1^1, x_1^2, \dots, x_1^{E'}, \dots, x_{U'}^1, x_{U'}^2, \dots, x_{U'}^{E'}\}$, 频谱资源分配决策的集合可表示为 $B = \{b_1^1, b_1^2, \dots, b_1^{E'}, \dots, b_{U'}^1, b_{U'}^2, \dots, b_{U'}^{E'}\}$, UAV或RSU的计算资源分配决策的集合可表示为 $F = \{f_1^1, f_1^2, \dots, f_1^{E'}, \dots, f_{U'}^1, f_{U'}^2, \dots, f_{U'}^{E'}\}$, 车辆的位置信息可表示为 $L = \{l_1, l_2, \dots, l_{U'}\}$ 。本文考虑了计算卸载、频谱分配和资源分配的联合优化, 因此, 最小化系统总成本的优化问题可以表示为

$$\begin{cases} \text{P: } Q^*(X, B, F, L) = \min_{A, B, F, L} \text{Cost}(X, B, F, L) \\ \text{s.t.} & \text{C1: } x_u^e \in \{0, 1\}, \forall u \in U, e \in E \\ & \text{C2: } \sum_{u=1}^{U'} x_u^e \cdot b_u^e \leq B_{\max}, \forall u \in U, e \in E \\ & \text{C3: } \sum_{u=1}^{U'} x_u^e \cdot f_u^e \leq F_{\max}, \forall u \in U, e \in E \\ & \text{C4: } 0 \leq l_u \leq l_{\max}, \forall u \in U, l \in L \end{cases} \quad (18)$$

其中, C1表示卸载决策为二进制变量, C2确保分配给SDV的带宽必须小于UAV或RSU的通信能力 $B_{\max}$, C3确保分配给SDV的计算资源小于边缘设备的最大处理能力 $F_{\max}$, C4表示SDV所属位置必须在所规定的范围内。

3 基于DRL的解决方案

在本文建立的混合整数非线性规划问题中, 卸载决策与频谱分配、资源分配是高度耦合的, 难以优化。为了解决这个问题, 本文提出了计算卸载和资源

分配算法GADDPG。其架构如图2所示, 模型将车载任务、网络状态和可用资源作为输入, 以计算卸载决策、频谱资源和计算资源分配方案作为输出, 使得算法能够获得最优的计算卸载和资源分配方案。

3.1 马尔可夫决策过程建模

为了便于使用GADDPG算法求解本文的联合优化问题, 需要使用强化学习来重新表述该问题。强化学习的标准模型是马尔可夫决策过程, 它可以用 (S, A, P, R) 来表示, 其中 S 表示状态空间, A 表示动作空间, P 表示状态转移函数, R 表示奖励函数。强化学习是一个时间序列模型, 因此需要将时间分帧处理。

3.1.1 状态空间

系统状态空间反映了从构建的系统环境中观察到的情况, 包括SDV的状态、边缘设备的可用资源。因此时隙 t 的状态空间 $s(t)$ 可表示为

$$s(t) = \{c_1(t), c_2(t), \dots, c_u(t), \dots, d_1(t), d_2(t), \dots, d_u(t), \dots, f_1(t), f_2(t), \dots, f_e(t), \dots, b_1(t), b_2(t), \dots, b_e(t), \dots, l_1(t), l_2(t), \dots, l_u(t)\} \quad (19)$$

其中, $c_u(t)$ 表示完成第 u 个SDV的请求任务所需要的CPU周期数, $d_u(t)$ 表示请求任务的输入数据量, $f_e(t)$ 表示边缘设备的可用计算资源, $b_e(t)$ 表示边缘设备可用信道带宽, $l_u(t)$ 表示SDV的位置。

3.1.2 动作空间

根据观察到的环境状态空间, 智能体决定车辆应该选择哪个边缘设备进行卸载, 以及应该向车辆分配多少计算资源和信道资源。因此, 智能体在时隙 t 的动作空间 $a(t)$ 可表示为

$$a(t) = \{x_1(t), x_2(t), \dots, x_u(t), \dots, f_1(t), f_2(t), \dots, f_e(t), \dots, b_1(t), b_2(t), \dots, b_e(t)\} \quad (20)$$

其中, $x_u(t)$ 表示车辆 u 在时隙 t 的卸载决策, $f_e(t)$ 表示边缘设备 e 在时隙 t 的可用计算资源, $b_e(t)$ 表示边缘设备 e 在时隙 t 的可用信道带宽。

3.1.3 奖励函数

本文的目标是最小化计算卸载和资源分配的系统总成本。因此, 奖励函数的设计应该能够引导智能体自主学习, 并做出最小化系统总成本的动作。其具体设计可表示为

$$R = -\text{Cost} = -\sum_{t=1}^T \sum_{u=1}^{U'} [\psi \cdot T_u(t) + (1 - \psi) \cdot E_u(t)] \quad (21)$$

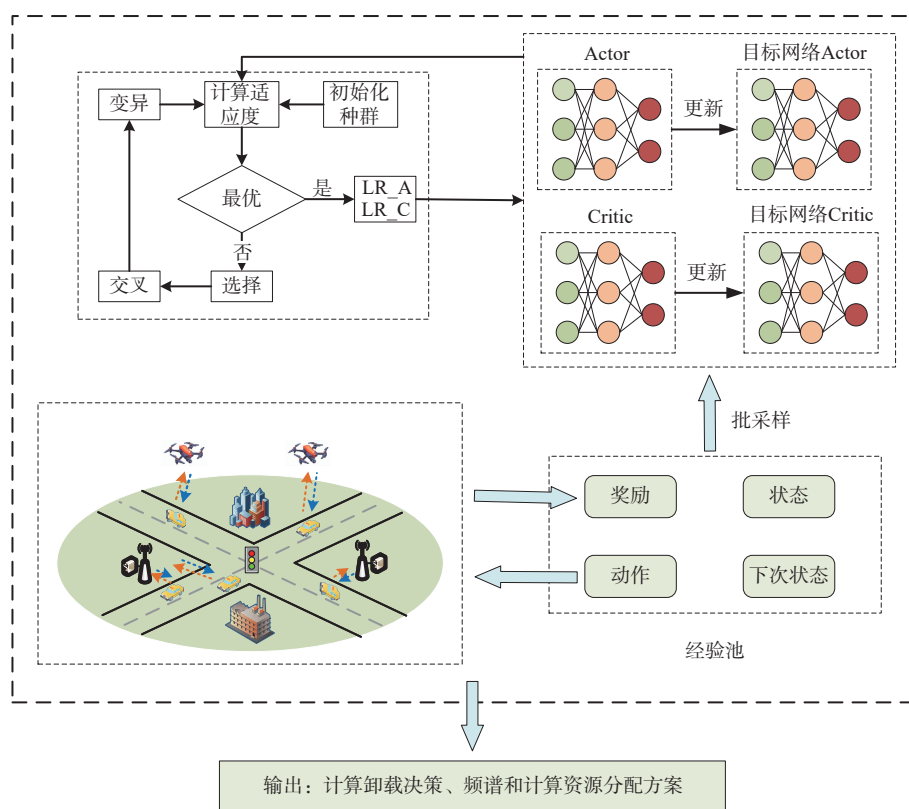


图2 基于GADDPG算法的计算卸载和资源分配策略架构

其中, $T_u(t)$ 和 $E_u(t)$ 分别表示车辆 u 在时隙 t 的卸载时间和能耗。

3.1.4 状态转移函数

本文的状态转移函数刻画计算资源与信道带宽资源的变化。该函数以当前时隙 t 的状态 $s(t)$ 和动作 $a(t)$ 作为输入, 以当前时隙的奖励 $R(t)$ 和下一个状态 $s(t+1)$ 作为输出。在此基础上, 状态转移函数可以表示为 $P(R(t), s(t+1) | s(t), a(t))$ 。

3.2 基于GADDPG算法的车联网计算卸载和资源分配

在车联网场景中, 车辆的瞬时移动性导致信道快速变化, 并且根据智能化驾驶对于多样化娱乐和低时延的需求, 传统方法难以进行建模求解。随着DRL的不断发展, 研究人员发现DRL更适合应用于具有不确定性和非稳定的车联网场景^[30]。一般来说, 有几种DRL算法可供选择, 如DQN算法和DDPG算法。DQN是一种传统方法, 可以基于Q值找到最优策略。通过实现深度神经网络来评估Q值以探索动作, 与Q学习算法相比, DQN算法在高维状态上学习策略, 但只能处理离散动作。然而, 本文考虑了混合连续和离散空间的高维状态和动

作, 所以选择了DDPG算法。DDPG算法是一种无模型、基于Actor-Critic架构的方法, 其具有两种网络架构, 可以通过策略梯度方法处理高维和连续的状态和动作。

但是, DDPG算法没有提供一个用来自主调整学习率的网络模型, 导致研究人员需要手动调整学习率来优化算法性能。然而, 手动调整学习率需要大量的人力和时间投入, 特别是在算法的不同阶段或者不同任务中, 需要反复尝试不同的学习率来找到最佳的性能。这不仅增加了调试和优化的复杂性, 还可能导致不稳定的结果。因此, 本文提出了GADDPG算法, 通过将遗传算法与DDPG算法相结合, 利用遗传算法预训练算法的学习率, GADDPG算法在求解过程中能够表现出较好的寻优能力。算法的具体设计思路如下。

1) 首先, 将遗传算法的适应度函数设置为GADDPG算法的奖励函数; 其次, 初始化染色体, 并利用适应度函数进行评估; 再次, 通过选择、交叉和变异操作更新染色体, 保留适应度较高的染色体并持续迭代训练; 最后, 获得具有最高适应度值的学习率。

2) 将GADDPG算法的学习率设定为遗传算法预训练后的数值, 并初始化车联网场景、经验重放缓冲区、网络参数、Actor和Critic模型等。

3) 智能体通过不断获得车联网环境信息来学习策略, 其中环境信息包括车辆的状态和边缘设备的可用计算资源和带宽资源, 然后根据所学到的策略将当前状态映射为动作, 制定资源分配方案和卸载决策。在迭代初期, 智能体通过添加高斯噪声来选择动作 $a(t)$, 提高探索不同动作的能力。动作 $a(t)$ 可表示为

$$a(t) = g(s(t)|\theta^\mu) + n(t) \quad (22)$$

其中, $g(s(t)|\theta^\mu)$ 表示 Actor 模块探索到的策略, $n(t)$ 表示高斯噪声。

4) 选择动作 $a(t)$ 后, 智能体可以计算当前时隙的奖励, 并返回下一个状态 $s(t+1)$ 。在每个训练周期, 智能体从经验池中拿到四元组数据, 即 $\{s(t), a(t), s(t+1), R(t+1)\}$, 作为 Actor 模块和 Critic 模块的训练数据集。为了保证网络的实时性, Critic 模块通过降低成本函数来不断更新参数 θ^Q , 成本函数可表示为

$$L(\theta^Q) = E[(y(t) - Q(s(t), a(t)|\theta^Q))^2] \quad (23)$$

其中, $Q(s(t), a(t)|\theta^Q)$ 表示动作值函数, 主要用来评估策略的价值。每一个动作都需要通过 $s(t)$ 和 $s(t+1)$ 来计算动作值函数, $y(t)$ 表示当前动作的目标值, 通过 Critic 模块的参数来获得

$$y(t) = R(s(t), a(t)) + \lambda Q'(s(t+1), \mu'(s(t+1)|\theta^\mu)|\theta^Q) \quad (24)$$

根据 $Q(s(t), a(t)|\theta^Q)$ 和采样经验池, Actor 模块通过梯度策略来更新 θ^μ , 其中梯度策略表示为

$$\begin{aligned} \nabla_{\theta^\mu} J &\approx E[\nabla_a Q(s, a|\theta^Q)|_{s=s(t), a=\mu(s=s(t))}] \\ &\nabla_{\theta^\mu} \mu(s|\theta^\mu|_{s=s(t)}) \end{aligned} \quad (25)$$

5) 上述更新仅涉及 Actor 模块和 Critic 模块, 其对应的目标网络尚未被更新。因此, 网络参数 θ^Q 和 θ^μ 使用软更新策略来更新 Actor 模块和 Critic 模块对应的目标网络, 其更新公式分别为

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (26)$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^\mu \quad (27)$$

其中, τ 是一个更新系数, 取值范围为 $\tau \in [0, 1]$ 。

3.3 网络训练算法设计

基于3.1节和3.2节的分析, 本文将所提出的GADDPG算法应用于求解计算卸载和资源分配问

题 (简称 CORA_GADDPG 算法)。首先, 初始化相关参数和染色体种群, 并设定适应度函数。然后, 在每个迭代周期根据适应度函数评判染色体种群的适应度值, 记录全局最优的适应度值和所对应的学习率。其次, 通过交叉、变异和选择操作来生成新一代的个体, 并通过不断迭代更新找到使得适应度函数最优的学习率 LR_A 和 LR_C。再次, 将这两个学习率输入到 GADDPG 算法的模型中, 并初始化 GADDPG 算法系统参数和车联网网络参数。在后续的训练过程中, 智能体不断与环境交互, 并根据动作 $a(t)$ 和状态 $s(t)$ 获得当前状态的奖励值 $R(t+1)$ 和下一个状态 $s(t+1)$, 将其存入经验重放缓冲区中。当缓冲区已满时, 从中选择64个样本进行训练, 并更新系统参数, 在每个时间步长结束后计算系统总成本。最后, 通过不断迭代获得系统的最优卸载策略和资源分配方案, 并输出系统最小总成本。具体算法设计如算法1和算法2所示。

算法1 学习率预训练算法

初始化染色体的个数、交叉概率、变异概率、迭代次数 $iter_{max}$ 等;

初始化环境、经验重放缓冲区、系统参数、最大训练次数 ep_{max} 、车联网网络数据等;

设定适应度函数 $fitness = Cost$;

根据染色体个数和编码长度初始化种群;

for $iter=1:iter_{max}$ **do**

根据适应度函数评估种群适应度值, 并选择适应度较高的策略;

记录全局最优解和对应的学习率 LR_A、LR_C;

通过交叉变异和选择操作生成新一代个体;

end for

输出 最优学习率 LR_A 和 LR_C

算法2 CORA_GADDPG 算法

使用 LR_A 和 LR_C 初始化算法学习率;

初始化车联网环境、经验重放缓冲区、系统参数、Actor 和 Critic 模型;

for $ep=1:ep_{max}$ **do**

根据距离优先模型为每个车辆选择距离最近的边缘设备;

根据环境状态初始化系统状态 $s(t)$;

for $t=1:T$ **do**

根据 $\mu(s(t)|\theta^\mu)$ 和 $s(t)$ 选择动作 $a(t)$;

对动作空间 $a(t)$ 进行标准化处理;

根据 $a(t)$ 和 $s(t)$ 获得 $R(t)$ 和 $s(t+1)$;
 将 $a(t)$ 、 $s(t)$ 、 $R(t)$ 和 $s(t+1)$ 存储到经验缓冲区;
if 缓冲区保存的记录超过缓冲区容量
 从经验缓冲区采样得到 64 条数据进行训练, 并根据式(24)计算 $y(t)$;
 通过式(23)更新网络参数 θ^o 以最小化损失函数;
 更新 $\mu(s(t)|\theta^\mu)$;
 根据软更新策略更新 θ^o 和 θ^μ ;
end if
 根据式(17)计算系统总成本;
end for
end for
 输出 最小的系统总成本。

4 仿真实验

本文对所提出的 CORA_GADDPG 算法进行了仿真实验, 并针对学习率和缓冲区大小对算法性能的影响进行了分析, 将 CORA_GADDPG 算法与随机进行计算卸载和资源分配算法 CORA_RANDOM、基于 DDPG 的计算卸载和资源分配算法 CORA_DDPG、基于 PPO 的计算卸载和资源分配算法 CORA_PPO、基于任务预处理深度确定性策略梯度 (TPDDPG, task preprocessing deep deterministic policy gradient)^[31] 的计算卸载和资源分配算法 CORA_TPDDPG 进行了比较, 以验证 CORA_GADDPG 算法的有效性。

4.1 仿真环境

在 UAV 辅助车联网边缘计算网络中, RSU 和 SDV 随机分布在 $L \times W = 1\,000 \times 1\,000$ (m^2) 的范围内, UAV 以固定高度 $H = 100$ m 飞行, 整个时间段被分为 900 个时隙。不考虑边缘设备的性能差异, 结合参考文献[32], 本文的仿真实验参数见表 1。

4.2 仿真结果

学习率和经验池大小作为强化学习中两个关键的超参数, 与网络的收敛速度和累计奖励密切相关。然而, 学习率和经验池大小与收敛速度并非线性相关, 因此, 选择合适的学习率和经验池大小至关重要。

DDPG 算法不能预训练学习率, 人工寻找学习率耗时且易出错, 因此本文使用遗传算法预训练学

表 1 仿真实验参数

参数	设定值
车辆数目 U'	{10,13,16,19,22}
RSU 服务器数量 N'	6
RSU 信道带宽 B_u/MHz	20
RSU 的计算能力 C_u/GHz	1
UAV 的数目 V'	2
UAV 信道带宽 B_v/MHz	40
UAV 的计算能力 C_v/GHz	1.5
车辆的最大传输功率 p_u/W	2
RSU 的平均功率消耗 p_u/W	2
任务所需 CPU 周期数 $c_u/\text{Mcycles}$	{10,30,50,70,90}
UAV 的平均功率消耗 p_v/W	4
任务的数据量 d_u/MB	{40,50,60,70,80}
任务回传数据量 r_u/MB	{4,5,6,7,8}
权重分配系数 ψ	0.5
采样数量 N_s	64
折扣因子 λ	0.95
软更新系数 τ	0.1

习率, 并将最终的结果作为 GADDPG 算法的学习率进行求解。学习率随迭代轮次变化情况见表 2, 学习率随迭代次数变化曲线如图 3 所示。通过表 2 和图 3 可知, 遗传算法预训练后的 LR_A 和 LR_C 分别为 0.000 267 和 0.000 988。不同学习率下的系统成本如图 4 所示。为验证预训练得到的学习率的正确性和有效性, 图 4 分别对比了 DDPG_LR 分别为 1×10^{-5} 和 1×10^{-3} (假设 DDPG 算法的 Actor 模块和 Critic 模块具有相同的学习率, 表示为 DDPG_LR) 以及 LR_A 和 LR_C 分别为 0.000 267 和 0.000 988 的

表 2 学习率随迭代轮次变化

轮次	LR_A	LR_C
10	0.000 314	0.000 004 4
20	0.003 423	0.003 653
30	0.000 123	0.000 224
40	0.000 176	0.000 687
50	0.000 313	0.000 723
60	0.000 267	0.000 988
70	0.000 216	0.000 987
80	0.000 268	0.000 988
90	0.000 677	0.000 988
100	0.000 267	0.000 988

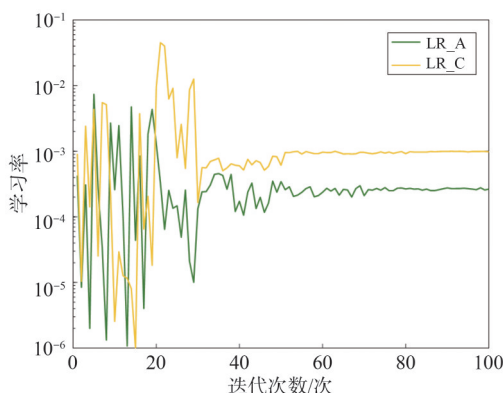


图3 学习率随迭代次数变化曲线

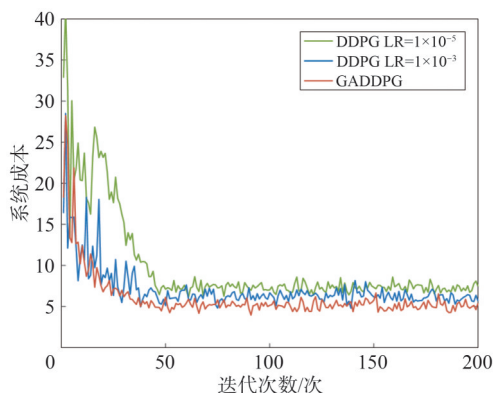


图4 不同学习率下的系统成本

实验结果，可以观察到，本文使用遗传算法预训练学习率对原始算法的收敛速度以及寻优结果都有着明显的改善。

不同经验池大小下的系统成本如图5所示。图5展示了本文所提出的算法在经验池大小分别为3 000、2 000和1 000条数据下，系统总成本随迭代次数的变化。当经验池大小为2 000条记录时，系统总成本收敛速度更快、更稳定，寻优效果更佳。因此设置网络的经验池大小均为2 000条记录。

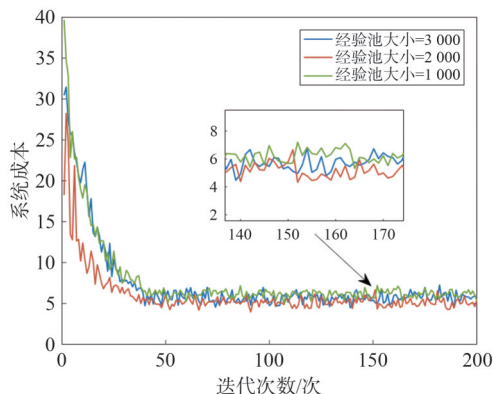


图5 不同经验池大小下的系统成本

图6展示了不同算法下的系统成本。与其他算法相比，通过CORA_RANDOM算法进行计算卸载和资源分配的成本更高，这是因为车辆卸载决策和资源分配方案是完全随机的，并没有考虑任务需求和可用资源的关系。此外本文所提出的CORA_GADDPG算法收敛速度更快，当迭代次数超过50次后，趋于稳定，具有更低的系统成本，说明本文算法是收敛且有效的。CORA_TPDDPG算法的优势在于可处理各种计算任务类型，并将其转化为标准化格式，以便进行有效的资源调度和任务卸载，而本文所提出的卸载策略为0-1卸载，仅为不可分割的任务类型，并未涉及其他任务类型。因此，CORA_TPDDPG算法在本文所提场景中效果欠佳。

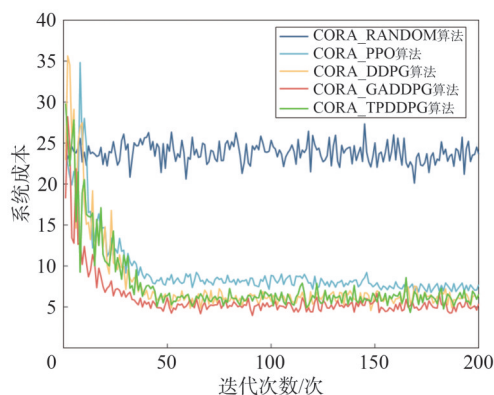


图6 不同算法下的系统成本

图7展示了不同车辆数下的系统成本，当边缘设备服务于更多的车辆时，这5种方法的系统成本都呈现增加的趋势。这是因为当有更多的车辆需要将任务卸载到边缘设备时，所产生的时延和能耗会变多，总成本自然也会增加。从图7可以看出，本文所提出的CORA_GADDPG算法相比其他4种基准算法具有最佳的效果，其次是CORA_DDPG算

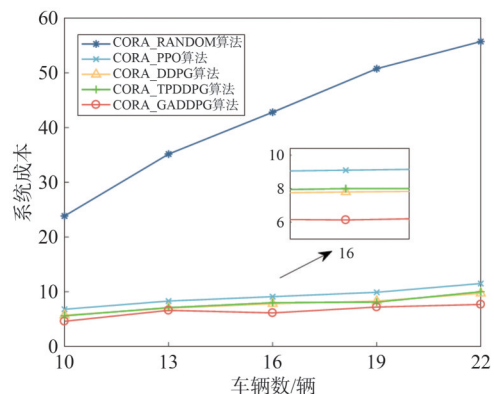


图7 不同车辆数下的系统成本

法和 CORA_TPDDPG 算法, 而 CORA_RANDOM 算法是通过随机方法进行卸载决策和资源分配, 所以其系统成本更高。

由图6和图7可知, CORA_RANDOM算法在进行计算卸载和资源分配时, 与其他4种算法差距较大, 因此在分析数据量大小和CPU周期对系统成本的影响时将不再进行对比。图8展示了不同数据量大小下的系统成本对比。请求任务需要卸载到边缘设备处理, 这会在传输过程中消耗时间, 数据量越大, 卸载延迟也就越大, 能量消耗也就越多。当数据量增大时, 与 CORA_TPDDPG 算法、CORA_DDPG 算法和 CORA_PPO 算法相比, CORA_GADDPG 算法可以获得更好的系统成本改善。这是因为 CORA_GADDPG 算法通过遗传算法预训练学习率, 使得 GADDPG 算法在动作空间能够获得更好的探索性能, 从而获得了具有更低成本的动作。此外 CORA_GADDPG 算法相比其他3种基线算法, 5种数据量大小的系统平均成本分别降低了8.1%、8.3%和15.9%。

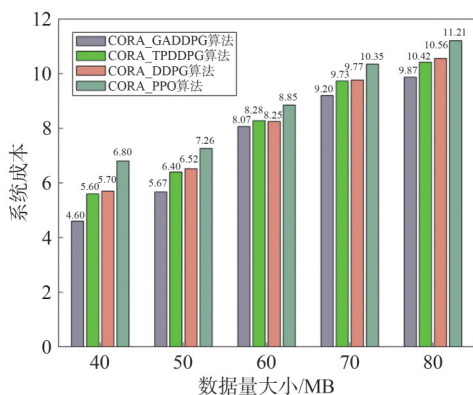


图8 不同数据量大小下的系统成本对比

不同CPU周期下的系统成本对比如图9所示。图9详细比较了 CORA_GADDPG 算法、CORA_TPDDPG 算法、CORA_DDPG 算法和 CORA_PPO 算法在不同CPU周期数的系统成本。可以看出, 当需要更多的CPU周期数时, 其他4种方案的系统成本均会增加。主要原因是车辆在处理计算密集型任务时, 需要更多的计算资源来执行, 所以能耗和时延也会随之增加。当CPU周期数为10 Mcycles时, 与 CORA_TPDDPG 算法、CORA_DDPG 算法和 CORA_PPO 算法相比, CORA_GADDPG 算法的系统成本分别降低了14.0%、15.38%和23.9%。随着所需要的CPU周期数增加, CORA_GADDPG 算法依然优于其他3种基线方案。

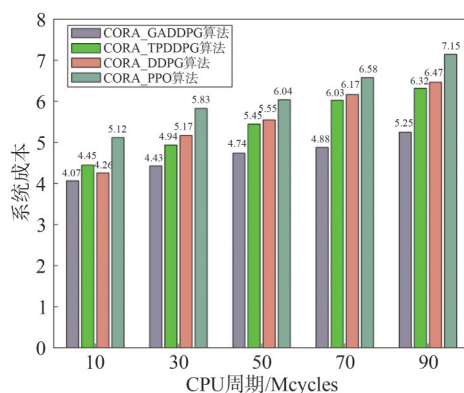


图9 不同CPU周期下的系统成本对比

5 结束语

本文针对车联网边缘计算这一热点领域, 提出了一种 UAV 辅助的车联网边缘计算卸载和资源分配方案, 该方案构建了车辆、UAV、搭载服务器的 RSU 之间的3层计算卸载和资源分配系统架构。考虑 UAV 和 RSU 的计算资源和频谱资源的有限性, 本文联合优化计算卸载策略、带宽资源和计算资源, 旨在最小化系统时延和能耗的加权和。上述复杂的计算卸载和资源分配问题被建模为一个马尔可夫决策过程, 并基于 DDPG 框架设计了一种 GADDPG 算法进行求解。仿真结果表明, 本文所提出的算法具有较好的收敛性并能有效降低系统成本。UAV 辅助车联网系统中的 UAV 轨迹优化问题以及多边缘节点之间的资源共享问题将是未来的研究方向。

参考文献:

- [1] Boukerche A, De Grande R E. Vehicular cloud computing: Architectures, applications, and mobility[J]. Computer Networks, 2018, 135: 171-189.
- [2] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing: a key technology towards 5G[R]. Sophia Antipolis: ETSI (European Telecommunications Standards Institute), 2015.
- [3] Guo H Z, Wang Y T, Liu J J, et al. Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond[J]. IEEE Transactions on Wireless Communications, 2024, 23(1): 347-359.
- [4] Liu Y, Xie S L, Zhang Y. Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system[J]. IEEE Transactions on Vehicular Technology, 2020, 69(10): 12229-12239.
- [5] 嵇介曲, 朱琨, 易畅言, 等. 多无人机辅助移动边缘计算中的任

- 务卸载和轨迹优化[J]. 物联网学报, 2021, 5(1): 27-35.
- Ji J Q, Zhu K, Yi C Y, et al. Joint task offloading and trajectory optimization for multi-UAV assisted mobile edge computing[J]. Chinese Journal on Internet of Things, 2021, 5(1): 27-35.
- [6] Sun C, Ni W, Wang X. Joint computation offloading and trajectory planning for UAV-assisted edge computing[J]. IEEE Transactions on Wireless Communications, 2021, 20(8): 5343-5358.
- [7] Li B, Yang R R, Liu L, et al. Robust computation offloading and trajectory optimization for multi-UAV-assisted MEC: a multiagent DRL approach[J]. IEEE Internet of Things Journal, 2024, 11(3): 4775-4786.
- [8] Yu X Y, Dong X, Yang X Q, et al. Air-ground integrated deployment for UAV-enabled mobile edge computing: a hierarchical game approach[J]. IET Communications, 2020, 14(15): 2491-2499.
- [9] Wang J Z, Zhang X L, He X S, et al. Bandwidth allocation and trajectory control in UAV-assisted IoV edge computing using multiagent reinforcement learning[J]. IEEE Transactions on Reliability, 2023, 72(2): 599-608.
- [10] Dai X X, Xiao Z, Jiang H B, et al. UAV-assisted task offloading in vehicular edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(4): 2520-2534.
- [11] 胡峰, 谷海洋, 林军. 无人机协同车载边缘网络中任务卸载策略[J]. 系统仿真学报, 2023, 35(11): 2373-2384.
- Hu F, Gu H Y, Lin J. UAV-enabled task offloading strategy for vehicular edge computing networks[J]. Journal of System Simulation, 2023, 35(11): 2373-2384.
- [12] 谭国平, 易文雄, 周思源, 等. 无人机辅助 MEC 车辆任务卸载与功率控制近端策略优化算法[J]. 电子与信息学报, 2024, 46(6): 2361-2371.
- Tan G P, Yi W X, Zhou S Y, et al. Proximal policy optimization algorithm for UAV-assisted MEC vehicle task offloading and power control[J]. Journal of Electronics & Information Technology, 2024, 46(6): 2361-2371.
- [13] Hu J N, Chen C, Cai L, et al. UAV-assisted vehicular edge computing for the 6G Internet of vehicles: architecture, intelligence, and challenges[J]. IEEE Communications Standards Magazine, 2021, 5(2): 12-18.
- [14] Zhang L, Ansari N. Optimizing the operation cost for UAV-aided mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2021, 70(6): 6085-6093.
- [15] Yang C, Liu B C, Li H Y, et al. Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2022, 71(9): 9884-9895.
- [16] Chu L J, Li X J, Xu J, et al. A holistic service provision strategy for drone-as-a-service in MEC-based UAV delivery[C]//Proceedings of the 2021 IEEE International Conference on Web Services (ICWS). Piscataway: IEEE Press, 2021: 669-674.
- [17] Hemmati A, Zarei M, Souri A. UAV-based Internet of Vehicles: a systematic literature review[J]. Intelligent Systems with Applications, 2023, 18: 200226.
- [18] Yan M, Xiong R, Wang Y, et al. Edge computing task offloading optimization for a UAV-assisted Internet of vehicles via deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2024, 73(4): 5647-5658.
- [19] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[PP/OL]. V1. arXiv (2013-12-19)[2025-08-19]. arXiv: 1312.5602.
- [20] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//Proceedings of the 33rd International Conference on Machine Learning (ICML 2016. New York: JMLR. org, 2016: 1928-1937.
- [21] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32. New York: ACM Press, 2014: I-387-I-395.
- [22] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[PP]. V2. arXiv (2017-08-28)[2025-08-19]. arXiv: 1707.06347.
- [23] 曾锋, 张政, 陈志刚. 基于深度强化学习的计算卸载与资源分配策略[J]. 通信学报, 2023, 44(7): 124-135.
- Zeng F, Zhang Z, Chen Z G. Computation offloading and resource allocation strategy based on deep reinforcement learning[J]. Journal on Communications, 2023, 44(7): 124-135.
- [24] Nath S, Wu J X. Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems[J]. Intelligent and Converged Networks, 2020, 1(2): 181-198.
- [25] Sadiki A, Bentahar J, Dssouli R, et al. Deep reinforcement learning for the computation offloading in MIMO-based Edge Computing[J]. Ad Hoc Networks, 2023, 141: 103080.
- [26] Liu X, Chai Z Y, Li Y L, et al. Multi-objective deep reinforcement learning for computation offloading in UAV-assisted multi-access edge computing[J]. Information Sciences, 2023, 642: 119154.
- [27] Gao H H, Wang X J, Wei W, et al. Com-DDPG: task offloading based on multiagent reinforcement learning for information-communication-enhanced mobile edge computing in the Internet of vehicles[J]. IEEE Transactions on Vehicular Technology, 2024, 73(1): 348-361.
- [28] Ge L H, Dong P H, Zhang H, et al. Joint beamforming and trajectory optimization for intelligent reflecting surfaces-assisted UAV communications[J]. IEEE Access, 2020, 8: 78702-78712.
- [29] Liu S S, Liu L K, Tang J, et al. Edge computing for autonomous driving: opportunities and challenges[J]. Proceedings of the IEEE, 2019, 107(8): 1697-1716.
- [30] Althamary I, Huang C W, Lin P. A survey on multi-agent reinforcement learning methods for vehicular networks[C]//Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). Piscataway: IEEE Press, 2019: 1154-1159.
- [31] Zhao J H, He L, Zhang D Y, et al. A TP-DDPG algorithm based on

cache assistance for task offloading in urban rail transit[J]. IEEE Transactions on Vehicular Technology, 2023, 72(8): 10671-10681.

- [32] Tian H, Xu X L, Qi L Y, et al. CoPace: edge computation offloading and caching for self-driving with deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13281-13293.

[作者简介]



岳文静(1982-), 女, 南京邮电大学通信与信息工程学院副教授, 主要研究方向为智能化无线网络、无线传感网、认知无线电网络。



李可(1999-), 男, 南京邮电大学通信与信息工程学院硕士生, 主要研究方向为车联网、移动边缘计算。



陈志(1978-), 男, 南京邮电大学计算机学院教授, 主要研究方向为移动物联网、无线传感网、计算机视觉。