



面向车联网的多智能体强化学习边云协同卸载

叶佩文¹, 贾向东², 杨小蓉¹, 牛春雨¹

(1. 西北师范大学 计算机科学与工程学院, 兰州 730070; 2. 南京邮电大学 江苏省无线通信重点实验室, 南京 214215)

摘要: 车联网边缘计算是实现车联网系统低时延和高可靠性的关键技术, 但现有方法普遍存在场景趋同和系统建模局限的问题, 同时包含复杂的训练过程并面临维灾风险。通过结合云计算技术, 提出一种基于多智能体强化学习的边云协同卸载方案。依据随机几何理论计算卸载节点覆盖概率, 对车辆节点与卸载对象进行预配对。利用线性Q函数分解方法反映每个智能体多效用因子与任务决策间的映射关系, 通过云端协同机制将智能体决策记录作为经验上传到云端, 并在云端将训练更完备的神经网络反馈到边缘节点。仿真结果表明, 该方案在功耗和延时方面性能优于单一固定边缘的计算策略, 且算法复杂度较低, 能够有效提升边云协同卸载能力, 实现低时延、高可靠的任务卸载。

关键词: 车联网; 多智能体强化学习; 随机几何理论; 边云协同计算; 任务卸载策略; 资源分配

开放科学(资源服务)标志码(OSID):



中文引用格式: 叶佩文, 贾向东, 杨小蓉, 等. 面向车联网的多智能体强化学习边云协同卸载[J]. 计算机工程, 2021, 47(4): 13-20.

英文引用格式: YE Peiwen, JIA Xiangdong, YANG Xiaorong, et al. Collaborative edge and cloud offloading for Internet of vehicles using multi-agent reinforcement learning[J]. Computer Engineering, 2021, 47(4): 13-20.

Collaborative Edge and Cloud Offloading for Internet of Vehicles Using Multi-Agent Reinforcement Learning

YE Peiwen¹, JIA Xiangdong², YANG Xiaorong¹, NIU Chunyu¹

(1. College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China;

2. Wireless Communication Key Lab of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 214215, China)

[Abstract] Edge computing for Internet of Vehicles (IoV) is key to realizing highly reliable and low-latency IoV systems. However, existing methods generally have the problems of scene convergence and system modeling limitations, and are faced with complex training processes and disaster maintenance risks. By combining the cloud computing technology, this paper proposes a collaborative edge and cloud offloading scheme based on multi-agent reinforcement learning. The strategy uses the stochastic geometry theory to calculate the coverage probability of the offloading nodes and pre-match the vehicular nodes to offloading objects. On this basis, the linear Q function decomposition method is used to reflect the mapping relationship between each agent's multi-utility factor and task decision. Then through the collaborative cloud and edge computing mechanism, each agent's decision records are uploaded to the cloud as experience, and the more comprehensively trained neural network is returned to the edge nodes. The results of simulation show that the proposed scheme outperforms the computing strategies using only fixed edge servers in terms of power consumption and latency. The method reduces the algorithm complexity, and can significantly improve the collaborative edge and cloud offloading ability to realize highly reliable and low-latency task offloading.

[Key words] Internet of Vehicles (IoV); multi-agent reinforcement learning; stochastic geometry theory; collaborative edge and cloud computing; task offloading strategy; resource allocation

DOI: 10.19678/j.issn.1000-3428.0058323

0 概述

目前, 车辆消费升级、道路容量日趋饱和等客观因素加剧了城市的交通困境。在5G商用落地同时

展望6G愿景的背景下, 构筑“车-人-路-云”泛在连接的车联网(Internet of Vehicles, IoV)成为必然趋势^[1-2]。与此同时, 在城市中部署智能交通系统(Intelligent Transportation System, ITS)缓解交通压力也已成为

基金项目: 国家自然科学基金(61861039, 61561043, 61261015); 甘肃省科技计划“无人机关键技术研究”(18YF1GA060)。

作者简介: 叶佩文(1993—), 男, 硕士研究生, 主研方向为车联网通信、移动边缘计算; 贾向东, 教授、博士; 杨小蓉、牛春雨, 硕士研究生。

收稿日期: 2020-05-14 修回日期: 2020-06-26 E-mail: jiaxd@nwnu.edu.cn

主流选择。而车联网作为ITS的核心部分,更需要在网络架构和使能技术上不断演进^[3]。

在网络架构方面,得益于云计算技术的迅猛发展,车联网的大量计算任务可以有效地迁移到分布式云端服务器上进行处理、统一调度和计算资源分配。然而,单一的车辆云架构不可避免地要求数据进行长距离、高时延传输,这无法满足需要敏捷响应的V2V安全类信息通信要求。通过分析车辆行为与预测模型可以发现车辆群体存在局部性特征^[4-5],即计算任务的卸载范围通常局限于相邻行驶车辆或车与周边路旁单元之间,而将移动边缘计算(Mobile Edge Computing, MEC)作为一种新的范式引入车联网,能够将计算能力下沉至网络边缘,从而减少服务时延^[6]。

在使能技术上,车联网边缘计算卸载可利用凸优化、图论以及博弈均衡等方法。但近年来人工智能特别是深度强化学习^[7-8]在计算机视觉、自然语言处理、语音识别等领域获得巨大成功,这吸引了国内外学者重新思考车联网边缘计算卸载方案的设计思路。

现有车联网边缘卸载策略存在场景同质化严重的问题,且在性能上仍有较大的提升空间。本文针对更泛在的城市街道场景,结合强化学习和随机几何理论,提出一种边云协同的车辆边缘卸载方案。结合随机几何理论和人工智能方法优化车联网边云卸载过程,将每个源车辆单元(Source Vehicle Unit, SVU)作为智能体来进行学习决策,并把由此产生的复杂训练过程转换到云端训练神经网络中,使SVU仅依靠局部决策即能把握全局特征。此外,还将资源队列模型作为神经网络输入前件,以降低维灾风险。

1 相关工作

文献[9]针对车辆边缘计算(Vehicular Edge Computing, VEC)网络提出了移动感知的任务卸载方法,以达到执行成本最小化的目的。文献[10]提出一种联合云计算、移动边缘计算和本地计算的多平台智能卸载方案,根据任务属性,利用强化学习算法选择卸载平台,旨在最小化时延并节省系统总成本,但网络模型中的控制面和数据面深度耦合,使得任务处理缺乏灵活性。文献[11]提出了基于软件定义^[12]的车载网络框架,其核心思想是将控制面和数据面分离,使运营商能够更灵活地控制和更快速地部署网络,但是车辆业务复杂和网络拓扑结构多变的因素导致这一框架对车辆特征的抽象还不够成熟,相应的车载网络虚拟化技术仍需要深入研究。文献[13]针对类似高速路口拥塞场景,利用车联网异构资源性能互补特性,即计算资源开销来供给通信资源需求,提出了基于雾计算^[14]的车联网边缘资源融合机制,从而弥补车联网资源时空分布不均的不足,但雾化机制^[15]涉及大量基础设施的改造和升

级,因此,该机制在构建部署阶段仍面临诸多挑战。

文献[16]针对车联网超可靠低延时通信(Ultra-Reliable Low-Latency Communication, URLLC)过程,将有异构性需求的车辆节点作为多智能体,利用强化学习进行数据卸载决策。文献[17]以相邻的车辆节点作为移动边缘服务器,以路边设施作为固定边缘服务器,利用半马尔科夫过程对时变信道进行建模,使移动用户根据Q学习算法确定卸载对象,同时针对增加训练过程动作状态空间可能引起维灾的问题,提出利用深度神经网络来逼近Q函数的深度强化学习算法,旨在使系统总效用最大。文献[18]基于值迭代和策略迭代两种思路提出动作-评价学习(Actor-Critic learning, AC)算法。得益于Actor执行动作然后Critic进行评估的优势,该算法在高维度空间仍具有良好的收敛属性,但存在评价策略偏差较大的问题,导致求解所得只是局部最优解。文献[19]将无线信道状态、缓存状态以及计算能力均纳入系统状态作为环境进行交互,由于通信、缓存、计算(Communication, Caching, Computing, 3C)资源在应用场景具有耦合互补的特性,因此综合权衡3C资源效用为任务调度卸载提供了一个广阔的思路。文献[20]在此基础上进一步考虑了时间尺度对协调优化的影响,提出大时间尺度采用粒子群优化理论而小时间尺度采用深度Q学习算法调优的细化方案。

然而,现有车联网边缘计算方法普遍存在以下不足:1)多数方法仅在单一的高速公路场景进行建模,而此类场景通常假定车辆服从空间泊松过程(Spatial Poisson Process, SPP),这明显限制了适用范围,且简化了车联网实际通信的真实时空分布;2)云计算平台大多采用集中式部署方案,计算任务从本地迁移到云端存在重构开销,且队列形式的任务传输易导致额外排队时延和无序争用,而目前缺乏边缘计算节点协同云平台的相关研究;3)在利用人工智能手段方面,现有研究的奖励机制设计单一,从而导致训练模型的泛化性较差。

本文结合强化学习和随机几何理论,提出一种边云协同的车辆边缘卸载方案,主要包括以下工作:

1)针对场景趋同、系统建模局限的问题,将城市街道建模为经典Manhattan模型^[21],并利用随机CoX过程^[22]对移动车辆进行细粒度建模,相应考虑视距(Line of Sight, LoS)和非视距(Non-Line of Sight, NLoS)两种情况的信道状态。进一步地,考虑到级联对象包含目标车辆单元(Target Vehicle Unit, TVU)和路边单元(Road Side Unit, RSU),对于SVU而言在时空上具备离散性和流动性,通过随机几何理论分析级联对象接收信干比(Signal to Interference Ratio, SIR)覆盖概率,从而划分出卸载节点的优先级,从根本上消除转化成组合优化问题的必要性,降低计算复杂度。

2)依据边云协同的思想,将SVU作为智能体进行决策,并将决策记录作为经验上传到云端,云端通过经验训练神经网络,每隔一段时间将训练更完备的神经网络反馈到边缘节点上。由此,只专注局部决策的SVU能够捕捉到云端存储的全局特征而无需承担复杂的训练过程。

3)由于强化学习的本质是环境交互和基于奖励,因此设计更贴近实际的多角色博弈奖励机制。同时,为使从全局观察缩小到局部观察具有实质性作用,将节点资源队列分析作为输入的预先工作,从而减少计算任务的排队时间,并在一定程度上降低维灾风险。

2 系统模型

本文研究的系统模型如图1所示,其中小区的网路架构由基站(Base Station, BS)和RSU共同组成。基站通过核心网络连接云端服务器,具有计算能力的RSU作为固定边缘服务器(Fixed Edge Server, FES),中央云服务器可以通过回程链路连接FES支持远程调度。将具有计算能力的TVU作为移动边缘服务器(Mobile Edge Server, VES),并从更广义的角度定义边缘节点性质,包括TVU和RSU两种类型。

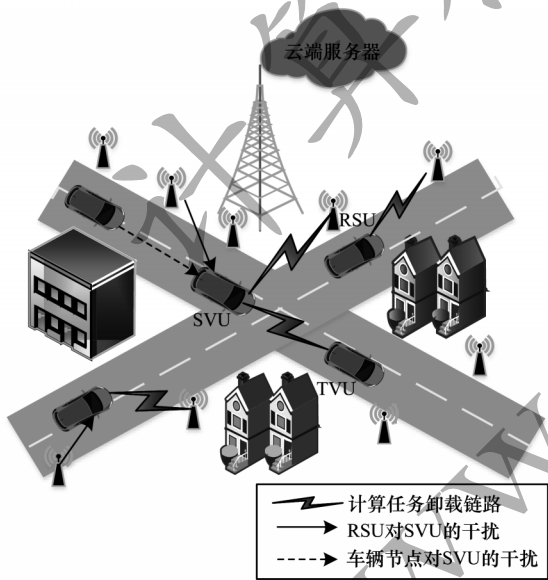


图1 车联网边云协同卸载系统模型

Fig.1 System model of collaborative edge and cloud offloading for IoV

假设本地计算容量已饱和,上述场景下的卸载途径可分为以下3种情况:1)在LoS范围内,SVU将计算任务卸载给相邻满足条件的TVU,由于车辆到BS上行链路的利用相对不充分,且BS端对干扰更具可控性,因此为提高频谱利用率,SVU可以复用V2B上行链路进行计算任务卸载;2)在NLoS范围内,SVU同样可以复用V2B上行链路进行计算任务卸载;3)在基础设施完备(即已部署RSU)的车辆稀

疏路况场景中,SVU可以将计算任务卸载到满足条件的RSU端。

本文将车联网的空间分布建模为泊松线性Cox点过程(Poisson Line Cox Point Process, PLCPP),对象包含车辆节点和路边节点。具体过程如下:将车辆节点空间分布建模为密度为 μ_v 的独立PLCPP,用 Φ_v 表示;考虑到RSU沿道路布放,将RSU空间分布建模为线密度为 μ_r 的独立泊松线过程(Poisson Line Process, PLP),用 Φ_r 表示。假设车辆节点中TVU占比为 β ,遵循PLCPP,则TVU服从密度为 $\mu_{TVU} = \mu_v \beta$ 的PLCPP,SVU服从密度为 $\mu_{SVU} = \mu_v (1 - \beta)$ 的PLCPP。

2.1 通信模型

不失一般性,本文假设SVU使用最近距离级联卸载准则^[23],并遵循广义边缘节点性质。定义计算任务集合 $T = \{T_1, T_2, \dots, T_j\}$,SVU集合用 K 表示,TVU集合用 N 表示,RVU集合用 γ 表示。考虑计算任务卸载到边缘节点存在视距(LoS)和非视距(NLoS)两种情况,在周期 t 内,SVU卸载计算任务 $T_j (j \in J)$ 到边缘节点的路径损耗可表示为:

$$h_j^{(t)} = \begin{cases} \rho \left(\sqrt{|x_j^{(1,t)} - x_j^{(2,t)}|^2 + |y_j^{(1,t)} - y_j^{(2,t)}|^2} \right)^{-\rho}, & \text{LoS范围} \\ \xi \left(|x_j^{(1,t)} - x_j^{(2,t)}| g |y_j^{(1,t)} - y_j^{(2,t)}| \right)^{-\xi}, & \text{NLoS范围} \end{cases} \quad (1)$$

其中, $x_j^{(t)} = (x_j^{(1,t)}, x_j^{(2,t)})$ 和 $y_j^{(t)} = (y_j^{(1,t)}, y_j^{(2,t)})$ 分别表示SVU和边缘节点的位置信息, $D = (\|x_j^{(t)} - y_j^{(t)}\|: \{x_j^{(t)} = (x_j^{(1,t)}, x_j^{(2,t)}), y_j^{(t)} = (y_j^{(1,t)}, y_j^{(2,t)})\})$ 为两者的距离, ρ 和 ξ 是路径损耗指数,且满足 $\xi < \rho(\varphi_0/2)^\circ$ 。

因此,在LoS范围内,第 n 个($n \in \mathbb{N}$)相邻TVU与第 k 个($k \in K$)SVU级联接收的SIR可表示为:

$$\text{SIR}_{k,n}^{\text{TVU, LoS}} = \frac{P_k g_{k,j}^{(t)} |D|^{-\alpha}}{I_{\text{RSU}}^{(k)} + I_{k'}^{(k)}} \quad (2)$$

其中, $g_{k,j}^{(t)} = \nu_k^{(t)} \cdot h_j^{(t)}$ 为信道增益, $\nu_k^{(t)}$ 为瑞利分布快速衰落因子, P_k 表示SVU的发射功率, $\|D\|$ 表示SVU到TVU的距离, $\alpha > 2$ 为频率参数, $I_{\text{RSU}}^{(k)} = \sum_{r \in \gamma} P_r g_r \|D\|^{-\alpha}$ 是来自RSU的干扰, P_r 是RSU的传输功率, g_r 是RSU对SVU的干扰。类似地, $I_{k'}^{(k)} = \sum_{(k' \in K) \vee (k' \neq k)} P_{k'} g_{k'} \|D_{k'}\|^{-\alpha}$ 表示其他车辆 k' ($k' \neq k, k' \in K$)对TVU的干扰, $P_{k'}$ 为 k' 的传输功率, $g_{k'}$ 为 k' 对SVU的干扰, $\|D_{k'}\|$ 表示第 k' 个SVU到第 k 个SVU的距离。

因此,LoS范围内卸载到第 k 个TVU的频谱效率可表示为:

$$e_{k,n}^{\text{TVU}} = \text{lb} \left(1 + \text{SIR}_{k,n}^{\text{TVU, LoS}} \right) \quad (3)$$

引理 1 假设 SVU 的卸载许可半径为 L_s , 与 SVU 级联的 TVU 接收的 SIR 覆盖概率可表示为式(4), 其中, $\Delta'(\alpha) = \pi \csc(\pi/\alpha)$, $\Delta = 2\pi \csc(2\pi/\alpha)/\alpha$ 。证明见文献[23]。

$$\begin{aligned}
 P_{\text{Cov}}^{\text{TVU}}(\zeta_v, \alpha, L_s) &= P_{\text{Cov}}(\text{SIR}_k^{\text{TVU, LoS}} \geq \zeta_v) = \\
 P_{\text{Cov}}\left(\frac{P_k g_k^{(i)} \|L_s\|^{-\alpha}}{I_{\text{RSU}}^{(k)} + I_{k'}^{(k)}} \geq \zeta_v\right) &= \\
 P_{\text{Cov}}\left(g_k^{(i)} \geq \frac{T_v \|L_s\|^{-\alpha}}{P_k} (I_{\text{RSU}}^{(k)} + I_{k'}^{(k)})\right) &\approx \\
 E\left[\sum_{i=1}^N w_i \exp\left(-\frac{u_i \zeta_v \|L_s\|^{\alpha}}{P_k} (I_{\text{RSU}}^{(k)} + I_{k'}^{(k)})\right)\right] &= \\
 \sum_{i=1}^N w_i \mathcal{L}_{I_{\text{RSU}}^{(i)}}(\delta) \mathcal{L}_{I_{k'}^{(i)}}(\delta) &= \\
 \sum_{i=1}^N w_i \exp\left(-2\mu_{\text{TVU}} \|L_s\| u_i \zeta_v^{1/\alpha} \Delta' \alpha\right) &\cdot \\
 \exp\left(-\mu_{\text{TVU}} \mu_i \pi^2 \|L_s\|^2 u_i \zeta_v^{2/\alpha} \Delta \alpha\right) & \quad (4)
 \end{aligned}$$

结合式(2)~式(4)可知, 从第 k 个 SVU 卸载到第 n 个 TVU 的数据速率为:

$$R_{k,n}^{\text{TVU}} = P_{\text{Cov}}^{\text{TVU}}(\zeta_v, \alpha, L_s) e_{k,n}^{\text{TVU}} \quad (5)$$

类似地, 对于 SVU 处于车辆稀疏且 RSU 设施完善的区域, SVU 可以卸载计算任务到满足条件的 RSU, 与第 k 个 SVU 级联的第 r 个 TVU 接收的 SIR 为:

$$\text{SIR}_{k,r}^{\text{RSU}} = \frac{P_k g_k^{(i)} \|D_r\|^{-\alpha}}{I_{\text{TVU}}^{(k)} + I_{r'}^{(k)}} \quad (6)$$

其中, $I_{\text{TVU}}^{(k)}$ 是来自 TVU 的干扰, $I_{r'}^{(k)}$ 是来自其他 TVU 的干扰。

因此, 卸载到第 r 个 RSU 的频谱效率可表示为:

$$e_{k,r}^{\text{RSU}} = \text{lb}(1 + \text{SIR}_{k,r}^{\text{RSU}}) \quad (7)$$

引理 2 假设 SVU 的卸载许可半径为 L_s , 与 SVU 级联的 RSU 接收 SIR 覆盖概率可表示为:

$$\begin{aligned}
 P_{\text{Cov}}^{\text{RSU}}(\zeta_r, \alpha) &= P_{\text{Cov}}\left(\frac{P_r g_r^{(i)} \|D\|^{-\alpha}}{I_{\text{TVU}}^{(k)} + I_{r'}^{(k)}} \geq \zeta_r\right) = \\
 c \left\{ \frac{1}{2b} + \frac{a \sqrt{\pi} e^{-a^2/4b}}{4b^{3/2}} \left[\text{erf}\left(\frac{a}{2\sqrt{b}}\right) - 1 \right] \right\} & \quad (8)
 \end{aligned}$$

其中, $a = 2\mu_{\text{TVU}} \Delta'(\alpha)$, $b = \pi^2 \mu_{\text{TVU}} \mu_r (\zeta_r P_r)^{2/\alpha} \Delta(\alpha)$, $c = 2\pi \mu_r$ 。证明见文献[23]。

结合式(7)和式(8)可知, 从第 k 个 SVU 卸载到第 r 个 RSU 的数据速率为:

$$R_{k,r}^{\text{RSU}} = P_{\text{Cov}}^{\text{RSU}}(\zeta_r, \alpha) e_{k,r}^{\text{RSU}} \quad (9)$$

2.2 计算模型

在计算卸载过程中, 可定义 SVU 的卸载任务 $T_j \triangleq (H_j, Q_j, t_j^{\max})$, 其中, H_j 表示计算任务数据大小, Q_j 表示完成任务所需计算资源量, $t_j^{\max}$ 表示最大等待时间。

对于将计算任务卸载到 TVU 的场景, SVU 卸载计算任务 H_j 到 TVU 的时间开销包括通信时间和计算时间两部分。

卸载到 TVU 的通信时间取决于计算任务数据大小 H_j 和提供服务 TVU 的数据速率, 结合式(9), 通信时间可以表示为:

$$t_{j,\text{comm}}^{\text{TVU}} = \frac{H_j}{R_{k,n}^{\text{TVU}}} \quad (10)$$

对于卸载到 TVU 的计算任务, 依照队列形式保存到 TVU 缓存中, 并更新资源队列状态, 实行任务迁移, 保证排队延时远小于任务计算时间。因此, 计算时间仅依赖于任务所需计算资源 Q_j 和 TVU 的计算能力 f_j^{TVU} (即单位时间内 CPU 周期数), 可表示为:

$$t_{j,\text{comp}}^{\text{TVU}} = \frac{Q_j}{f_j^{\text{TVU}}} \quad (11)$$

结合式(10)和式(11)可知, TVU 的总执行时间为:

$$t_j^{\text{TVU}} = t_{j,\text{comm}}^{\text{TVU}} + t_{j,\text{comp}}^{\text{TVU}} \quad (12)$$

类似地, 将计算任务卸载到 RSU 场景的执行时间同样包括通信时间和计算时间两部分。

卸载到 RSU 的通信时间可表示为:

$$t_{j,\text{comm}}^{\text{RSU}} = \frac{H_j}{R_{k,r}^{\text{RSU}}} \quad (13)$$

卸载到 RSU 的计算时间可表示为:

$$t_{j,\text{comp}}^{\text{RSU}} = \frac{Q_j}{f_j^{\text{RSU}}} \quad (14)$$

结合式(13)和式(14)可知, RSU 的总执行时间为:

$$t_j^{\text{RSU}} = t_{j,\text{comm}}^{\text{RSU}} + t_{j,\text{comp}}^{\text{RSU}} \quad (15)$$

2.3 资源队列分析模型

SVU 在卸载计算任务时需要考虑卸载节点计算队列大小。不失一般性, 假设队列节点初始资源量为 q_{initial} , 平均计算任务到达率 $E[T_j] = \lambda$, 在许可半径 L_s 内满足资源量的 TVU 和 RSU 概率分别可表示为:

$$P_q^{\text{TVU}} = \sum_{i=1}^{q_{\text{initial}}} Z_i(\lambda, L_s) \quad (16)$$

$$P_q^{\text{RSU}} = \sum_{i=1}^{q_{\text{initial}}} Z_i(\lambda, L_s) \quad (17)$$

其中, $Z_i(\lambda, L_s)$ 表示 Zipf 分布。

假设所考虑的车联网中 SVU 在周期 t 内生成任务的概率为 p_j , SVU 级联卸载节点可提供的资源量表示为 $\Theta = P(D^2)(1-p_j)\mu_{TVU}\mu_{SVU}E[L]$, 其中, $P(D^2)$ 表示平面周长, $E[L]$ 表示道路平均长度。因此, TVU 所需保证的队列长度为 $\Theta^{TVU} = \Theta(1-P_q^{TVU})$, SVU 所需保证的队列长度为 $\Theta^{SVU} = \Theta P_q^{RSU}$ 。相应地, 对于周期 t 内生成任务 T_j , 卸载节点资源队列长度 $q_t^{(j)} = \max\{\Theta(1-P_q^{TVU}), \Theta P_q^{RSU}\}$ 。

3 问题描述

本节分别从执行时延、能耗约束和费用开销维度分析车联网任务卸载问题, 并量化统一成系统效用评价卸载性能, 将 3 个维度评价性能的累加作为奖励机制来反馈训练神经网络。

1) 执行时延 d_j^F 。定义 F 为指示符, 用于区分 TVU 和 RSU, 则计算任务 T_j 执行时延 d_j^F 可表示为:

$$d_j^F = \begin{cases} t_{j, \text{comm}}^{TVU} + t_{j, \text{comp}}^{TVU}, & F \propto \text{TVU} \\ t_{j, \text{comm}}^{RSU} + t_{j, \text{comp}}^{RSU}, & F \propto \text{RSU} \end{cases} \quad (18)$$

2) 能耗约束 ψ_j^F 。设置计算任务卸载满足能耗约束条件为正, 否则作为惩罚进行负反馈, TVU 和 RSU 完成计算任务 T_j 的能耗分别可表示为 $E_j^{TVU} = P_n Z_j / R_{k,n}^{TVU}$ 和 $E_j^{RSU} = P_r Z_j / R_{k,r}^{RSU}$, 相应地, 设定能耗阈值分别为 δ^{TVU} 和 δ^{RSU} , 同时设置指示函数 $I_{\{\Omega\}}$, 若 Ω 成立则 $I_{\{\Omega\}} = 1$, 否则 $I_{\{\Omega\}} = -1$, 则能耗约束可表示为:

$$\psi_j^F = I_{\{\{E_j^{TVU} > \delta^{TVU}\} \vee \{E_j^{RSU} > \delta^{RSU}\}\}} \quad (19)$$

3) 费用开销 ϕ_j^F 。考虑实际网络架构包括车联网运营商 (Vehicle Network Operator, VNO)、基础设施供应商 (Infrastructure Provider, InP) 和业务供应商 (Service Provider, SP) 三类角色。假设 VNO 需向 InP 支付的频谱租赁费用为 ϵ , VNO 需向 SP 支付计算费用 ϕ , 则计算任务 T_j 的费用开销包含通信开销和计算开销两部分, 可以表示为:

$$\phi_j^F = \begin{cases} (\epsilon^{TVU} + \phi^{TVU}) \cdot R_{k,n}^{TVU}, & F \propto \text{TVU} \\ (\epsilon^{RSU} + \phi^{RSU}) \cdot R_{k,r}^{RSU}, & F \propto \text{RSU} \end{cases} \quad (20)$$

综上所述, 基于多角色博弈的奖励机制, 卸载任务的系统效用可由执行时延、能耗约束和费用开销三者的子效用累计和来表示, 即:

$$u_j = \omega_1 d_j^F + \omega_2 \psi_j^F + \omega_3 \phi_j^F \quad (21)$$

其中, $\omega = [\omega_1, \omega_2, \omega_3]$ 表示子效用系数, 在训练过程中通过调整该参数来确定子效用的倾向性, 例如在稀疏场景下更关注费用开销, 而在密集场景下更关注执行时延。

4 多智能体强化学习边云卸载机制

车联网是典型的高速移动实时传输场景, 在其中进行单一云端集中式优化存在参数冗余、更新滞后和耗费通信开销等问题。分布式边云协同机制利用云端将训练更完备的神经网络反馈到边缘节点, 使得边缘节点仅需要相邻节点信息来更新参数执行操作。本节首先介绍状态空间、动作空间及系统效用, 然后描述多智能体强化学习流程及改进方案。

1) 状态空间。用 S 表示状态集合, 在时刻 t 的状态可表示为 $s_j^{(t)} = \{d_j^{(F,t)}, \psi_j^{(F,t)}, \phi_j^{(F,t)}\} \in S$, 该状态表征了计算任务 T_j 的时延、能耗和开销状态。

2) 动作空间。定义动作集为 A , 计算任务采取的动作 $\alpha_j \in A$, $\alpha_j = 1$ 代表计算任务 T_j 卸载到 TVU 上执行, $\alpha_j = 0$ 代表计算任务 T_j 卸载到 RSU 上执行, 否则在该周期内不采取任何动作。

3) 系统效用。累计任务时延、能耗约束、费用开销三者的子效用作为奖励函数评价动作空间与状态空间的映射关系, 结合式 (21), 在时刻 t 系统立即效用可表示为:

$$U(s_j^{(t)}, \alpha_j) = \omega_1^t d_j^F + \omega_2^t \psi_j^F + \omega_3^t \phi_j^F \quad (22)$$

进一步地, 由于当前的动作仅受前一时刻状态影响, 当前状态通过执行某一动作转换到下一状态, 因此可利用马尔科夫决策过程来表述。假设状态空间到动作空间的映射为 Φ , 即 $\Phi(s_j^{(t)}) = \alpha_j$, 则系统状态转移概率可表示为:

$$\begin{aligned} \Pr\{s_j^{(t+1)} | s_j^{(t)}, \Phi(s_j^{(t)})\} &= \Pr\{d_j^{(t+1)} | d_j^{(t)}, \Phi(s_j^{(t)})\} \cdot \\ &\Pr\{\psi_j^{(t+1)} | \psi_j^{(t)}, \Phi(s_j^{(t)})\} \cdot \\ &\Pr\{\phi_j^{(t+1)} | \phi_j^{(t)}, \Phi(s_j^{(t)})\} \end{aligned} \quad (23)$$

状态值函数 $V_\pi(s)$ 和状态动作函数 $Q_\pi(s, \alpha)$ 可以表示为:

$$V_\pi(s) = E_\pi \left[(1-\gamma) \sum_{t=0}^{\infty} \gamma^t U(s_j^{(t)}, \alpha_j) \mid s_j^{(0)} = s \right] \quad (24)$$

$$\begin{aligned} Q_\pi(s, \alpha) &= (1-\gamma) U(s_j^{(0)}, \alpha_j) + \\ &\gamma \cdot \sum_{s_j^{(t+1)}} \Pr\{s_j^{(t+1)} | s_j^{(t)}, \Phi(s_j^{(t)})\} \cdot V(s_j^{(t+1)}) \end{aligned} \quad (25)$$

其中, π 表示 SVU 当前采取的策略, γ 表示折扣因子。根据式 (25) 和贝尔曼公式的定义, 进一步可得到式 (26):

$$\begin{aligned} Q_\pi(s, \alpha) &= (1-\gamma) U(s, \alpha) + \\ &\gamma \sum_{s_j^{(t+1)}} \Pr\{s_j^{(t+1)} | s_j^{(t)}, \Phi(s_j^{(t)})\} \cdot \\ &\max_{\alpha_j^{(t+1)}} Q(s_j^{(t+1)}, \alpha_j^{(t+1)}) \end{aligned} \quad (26)$$

基于值迭代或策略迭代的传统方法需要智能体获得全局信息,不适用于信息特征变化频繁的车联网场景。由于Q学习在与环境交互的过程中通过局部信息不断地试错来找到最优行为,因此SVU可以通过Q学习最大化长期效用获得最佳的控制决策。但结合式(22)和式(26)可知此方法存在以下两点不足:1)状态空间的大小会随着训练的轮数大幅增加,影响收敛性;2)出现梯度消失或梯度爆炸现象,导致模型退化。本文对此做以下改进:

1)采用经验重放策略。

将智能体在环境探索过程中获得的经验数据存放在经验池中,在后续训练深度神经网络的过程中随机采样更新网络参数。经验池表示为 $M^{(j)} = \{m^{(j-M+1)}, m^{(j-M)}, \dots, m^{(j)}\}$,存放的经验数据元组表示为 $m^{(j)} = (s_j^{(j)}, a_j, U(s_j^{(j)}, a_j), s_j^{(j+1)})$ 。SVU随机采样 $\hat{M}^{(j)} \subseteq M^{(j)}$ 输入神经网络进行训练,而非直接利用连续样本进行训练。更新规则可表示为:

$$Q^{(t+1)}(s, a) = Q^{(t)}(s, a) + \alpha^{(t)} \left((1-\gamma) \cdot U(s, a) + \gamma \cdot \max_{a_j} Q^{(t)}(s_j, a_j) - Q^{(t)}(s_j, a_j) \right) \quad (27)$$

2)采用边云协同思想及线性Q函数分解理论。

由式(22)可知,系统效用由任务时延、能耗约束和费用开销累加形成,需要训练大量的参数,不可避免地需要更多的计算资源和存储资源,并且会增加训练时间。因此,本文利用边云协同思想,将神经网络的训练过程放置到云端,利用经验回放池的数据进行训练。此外,采用线性Q函数分解理论对式(22)做进一步改进。设智能体指示符 $K = \{1, 2, 3\}$ 分别对应3个效用分量,式(22)可表示为:

$$U(s, a) = \sum_{k \in K} \omega_k u_k \quad (28)$$

进一步地,式(25)可以表示为:

$$Q(s, a) = \sum_{k \in K} Q_k(s, a, \theta) \quad (29)$$

因此,式(27)所示的更新规则改进为:

$$Q_k^{(t+1)}(s, a) = Q_k^{(t)}(s, a) + \alpha^{(t)} \left((1-\gamma) \cdot U_k(s, a) + \gamma \cdot Q_k^{(t)}(s_j, a_j) - Q_k^{(t)}(s_j, a_j) \right) \quad (30)$$

5 仿真结果与分析

5.1 仿真设置

利用MATLAB仿真平台对所提边云协同卸载方案进行仿真评估。仿真遵循Manhattan模型描述的参数设置并按照MEC白皮书^[6]相关规定构建系统

模型,使用SNIA云服务器记录所有数据集,并提供云端的计算服务支持边缘节点参数更新。具体仿真参数见表1。

表1 仿真参数

Table 1 Parameters of simulation

参数	数值
车辆节点空间分布密度	$\mu_v = 50/\pi$
TVU占比/MHz	$\beta = 0.2$
计算任务数量	$J = 5\ 000$
视距/非视距路径损耗指数/dB	$\rho = -68.5, \xi = -54.5$
SVU/RSU/TVU数量	$K = 5, \gamma = 20, N = 10$
频率参数	$\alpha = 4$
SVU/RSU/TVU功率/dBm	23, 54, 23
SVU许可半径/m	$L_s = 50$
计算任务大小/KB	$H_j = 400$
资源量/GHz	$Z_j = 0.2$
最大时延/ms	$t_j^{\max} = 20$
RSU/TVU计算能力/(GHz·周期 ⁻¹)	4/2
计算任务到达率	0.1~0.9
子效用因子向量	$\omega = [0.6, 0.2, 0.2]$
经验重放池容量	$M = 3\ 000$
Mini-batch大小	500
更新步长	80
神经网络层数	4
神经元总数	100

5.2 结果分析

为验证所提方案的收敛性并比较不同学习率对其收敛性的影响,将时延作为参照结果。由图2可以看出:当学习率为0.01时,收敛到一个局部最优解时延较大;缩小学习率至0.005可以得到较大的性能提升,但收敛速度变缓;学习率为0.001时,在收敛结果上仍有较大提升。考虑到更小的学习率会导致长时间无法收敛,本文采用0.001的学习率作为后续实验参数。

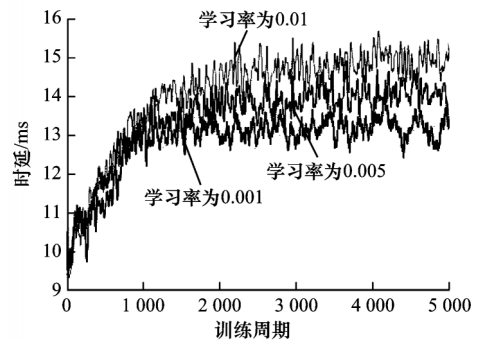


图2 不同学习率下训练周期与时延的关系

Fig.2 The relationship of training period and time delay under different learning rates

不同方案的累计能耗随训练周期的变化趋势如图3所示,其中累计能耗的大小代表了计算任务迁移量。可以看出:对照组没有利用云端技术,边缘节点累计能耗在短时间内快速上升,随着任务量逐步均衡迁移而达到稳定状态,因为路边单元的计算能力强于车辆节点,所以仅V2I卸载方案略优于仅V2V卸载方案;本文方案利用边云协同优势,在训练过程中参数更新及时,局部参数的快速迭代能拟合得到全局最优解,因此在较短的时间内就达到了系统功耗均衡。

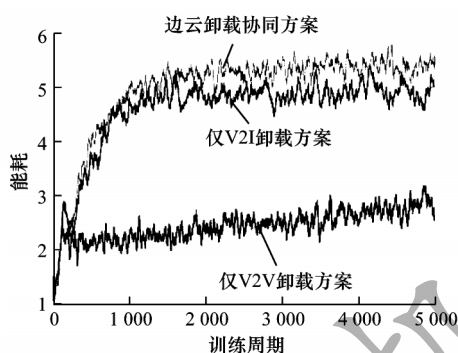


图3 不同方案训练周期与能耗的关系

Fig.3 The relationship of training period and energy consumption under different schemes

不同方案系统效用随计算任务到达率及频谱分配因子的变化趋势如图4所示。可以看出:一方面,系统效用随计算任务到达率先增后减,在计算任务到达率为0.6时达到最佳状态,此时资源队列较优,使得执行子效用对系统效用倾向性较大,通过调整子效用系数来应对不同场景需求,本文对于执行时间有强约束,故将子效用系数设置为 $\omega=[0.6,0.2,0.2]$;另一方面,计算任务生成率越高也能提升系统效用,任务数据传输不易丢包。相较于仅V2V卸载和仅V2I卸载方案,本文方案具有明显优势。在对照组实验中,当计算任务到达率仅为0.5时资源队列就达到饱和状态,计算任务已处于排队状态。

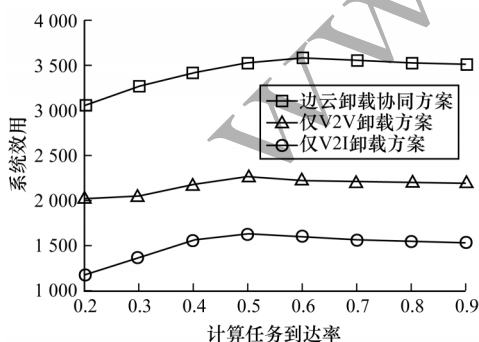


图4 不同方案计算任务到达率与系统效用的关系

Fig.4 The relationship of system utility and computation tasks arrival rate under different schemes

不同频谱分配因子 m 下系统效用与计算任务到达率的关系如图5所示。可以看出,尽管本文方案计算任务到达率为0.6时系统效用已达到最大值,但随着到达率的增加仍保持接近最佳值,体现了本文方案的有效性。

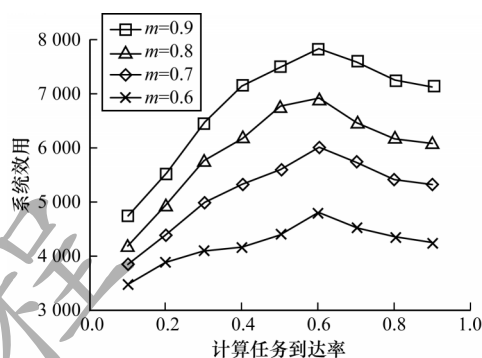


图5 不同频谱分配因子下计算任务到达率与系统效用的关系

Fig.5 The relationship of computation tasks arrival rate and system utility under different spectrum allocation factors

不同训练周期和子效用系数设置下时延与能耗的关系如图6所示。可以看出:一方面,随着训练周期增加,即训练的迭代次数的增加能反馈给智能体更完备的神经网络结构,使得计算任务的卸载能耗有所下降,但值得注意的是,云端训练的开销能否得到有效供给是一个开放性问题,在本文中训练周期为4000时达到了收敛状态;另一方面,随着子效用系数 ω_2 的增大,系统的能耗也逐渐增大。费用开销子效用系数 ω_3 对于能耗的影响较大, ω_3 值增大导致能耗快速增长。

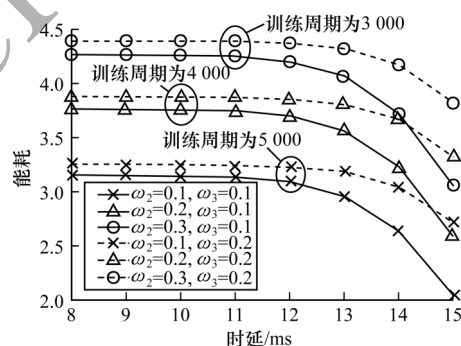


图6 不同训练周期和子效用系数设置下时延与能耗的关系

Fig.6 The relationship of delay and energy consumption under different settings of training period and sub-utility coefficient

6 结束语

本文提出一种基于多智能体强化学习的车联网任务卸载方案。采用随机几何理论对资源队列进行控制,从而降低任务排队时延,同时分离云端训练神经网络和节点决策神经网络,使云端能够更精准地提取环境特征,节点端则定时根据云端反馈的优化

参数进行在线决策。仿真结果表明,与单一固定边缘的计算策略相比,该方案能够有效减小时延和能耗并且降低计算复杂度。下一步将结合节点缓存技术设计更高效的车联网计算任务卸载方案。

参考文献

- [1] ZHANG L, LIANG Y C, NIYATO D. 6G visions: mobile ultra-broadband, super Internet-of-things, and artificial intelligence[J]. China Communications, 2019, 16(8):1-14.
- [2] YANG Xiaodong, PEI Xizhen, AN Faying, et al. Message authentication scheme for vehicular ad hoc network using identity-based aggregate signature[J]. Computer Engineering, 2020, 46(2):170-174, 182. (in Chinese)
杨小东,裴喜祯,安发英,等. 基于身份聚合签名的车载自组网消息认证方案[J]. 计算机工程, 2020, 46(2):170-174, 182.
- [3] CHANG Xiang, ZHANG Rongqing, YANG Liuqing. Wireless toward the era of intelligent vehicles[J]. IEEE Internet of Things Journal, 2019, 6(1):188-202.
- [4] ZHOU Huan, WANG Hui, CHEN Xi. Data offloading techniques through vehicular ad hoc networks: a survey[J]. IEEE Access, 2018, 6:65250-65259.
- [5] YE Peiwen, JIA Xiangdong, YANG Xiaorong, et al. End-to-end physical layer optimization scheme using auto-encoder based on deep learning[J]. Computer Engineering, 2019, 45(12):86-90, 97. (in Chinese)
叶佩文,贾向东,杨小蓉,等. 基于深度学习的自编码器端到端物理层优化方案[J]. 计算机工程, 2019, 45(12):86-90, 97.
- [6] ZHANG Ke, MAO Yuming, LENG Supeng, et al. Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading[J]. IEEE Vehicular Technology Magazine, 2017, 12(2):36-44.
- [7] NING Zhaolong, DONG Peiran, WANG Xiaojie, et al. Deep reinforcement learning for intelligent Internet of vehicles: an energy-efficient computational offloading scheme[J]. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(4):1060-1072.
- [8] XIE Shuai. A review of deep learning and intensive learning[J]. Information Technology & Informatization, 2020(5):225-227. (in Chinese)
解帅. 深度学习和深度强化学习综述[J]. 信息技术与信息化, 2020(5):225-227.
- [9] YANG Chao, LIU Yi, CHEN Xin. Efficient mobility aware task offloading for vehicular edge computing networks[J]. IEEE Access, 2019, 7:26652-26664.
- [10] CUI Yaping, LIANG Yingjie, WANG Ruyan. Resource allocation algorithm with multi-platform intelligent offloading in D2D-enabled vehicular networks[J]. IEEE Access, 2019, 7:21246-21253.
- [11] LI Baozhu, ZHAO Xuhui. New SDN-based architecture for integrated vehicular cloud computing network[C]// Proceedings of 2018 IEEE International Conference on Selected Topics in Mobile and Wireless Networking. Washington D. C., USA: IEEE Press, 2018:1-4.
- [12] KING D, ROTSOS C, AGUADO A, et al. The software defined transport network: fundamentals, findings and futures[C]//Proceedings of International Conference on Transparent Optical Networks. Washington D. C., USA: IEEE Press, 2016:1-4.
- [13] DAI Y Y, XU D, MAHARJAN S, et al. Artificial intelligence empowered edge computing and caching for Internet of vehicles[J]. IEEE Wireless Communications, 2019, 26(3):12-18.
- [14] REJIBA Z, MASIP-BRUIN X. Computation task assignment in vehicular fog computing: a learning approach via neighbor advice [C]//Proceedings of the 18th International Symposium on Network Computing and Applications. Washington D. C., USA: IEEE Press, 2019:1-5.
- [15] HOU Xueshi, LI Yong, CHEN Min, et al. Vehicular fog computing: a viewpoint of vehicles as the infrastructures[J]. IEEE Transactions on Vehicular Technology, 2016, 65(6):3860-3873.
- [16] ALTHAMARY I, HUANG C W, LIN P. A survey on multi-agent reinforcement learning methods for vehicular networks[C]//Proceedings of the 15th International Wireless Communications and Mobile Computing Conference. Washington D. C., USA: IEEE Press, 2019:1154-1159.
- [17] LIU Yi, YU Huimin, XIE Shengli. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(11):11158-11168.
- [18] YANG H L, XIE X Z, KADOCH M. Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5):4157-4169.
- [19] HE Yin, ZHAO Nan, YIN Hongxi. Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2018, 67(1):44-55.
- [20] TAN L T, HU R Q, HANZO L. Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(4):3086-3099.
- [21] CHEN X F, WU C, ZHANG H G, et al. Decentralized deep reinforcement learning for delay-power tradeoff in vehicular communications[C]//Proceedings of International Conference on Communications. Washington D. C., USA: IEEE Press, 2019:1-6.
- [22] CHOI C S, BACCELLI F. Poisson cox point processes for vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2018, 67(10):10160-10165.
- [23] CHOI C, BACCELLI F. An analytical framework for coverage in cellular networks leveraging vehicles[J]. IEEE Transactions on Communications, 2018, 66(10):4950-4964.