

车联网边缘计算环境下基于流量预测的 高效任务卸载策略研究

许小龙^{1,2,3}, 杨威¹, 杨辰翊⁴, 程勇¹, 齐连永^{5*}, 项昊龙¹, 窦万春⁶

(1. 南京信息工程大学软件学院, 江苏南京 210044; 2. 江苏省大气环境与装备技术协同创新中心, 江苏南京 210044; 3. 江苏省先进计算与智能服务工程研究中心, 江苏南京 210044; 4. 东南大学软件学院, 江苏南京 211189; 5. 中国石油大学(华东)计算机科学与技术学院, 山东青岛 266580; 6. 南京大学计算机软件新技术国家重点实验室, 江苏南京 210000)

摘要: 车联网(Internet of Vehicles, IoV)边缘计算通过将移动边缘计算和车联网相结合, 实现了车辆计算任务从云服务器向边缘服务器的下沉, 从而有效降低了车联网服务的响应时延。然而, 车联网中不规则的交通流时空分布会导致边缘服务器计算负载不均衡, 进而影响车联网服务的实时响应。为此, 本文提出了一种车联网边缘计算环境下基于流量预测的高效任务卸载策略。具体而言, 首先设计了能充分挖掘路段间连通性和距离信息的切比雪夫图加权网络(Chebyshev graph Weighted Network, ChevWN)进行交通流量预测。然后, 设计了一种基于深度强化学习的二元任务卸载方法(DRL-based Binary task Offloading Algorithm, DBOA), 该算法将二元任务卸载的决策过程分为两个阶段, 即首先通过深度强化学习得到卸载策略, 再通过一维双端查找算法确定最大化总计算速率的时间片分配方案, 降低了决策过程的复杂度。最后, 通过大量的对比实验验证了ChevWN在预测交通流量方面的准确性, 以及DBOA在提升车联网服务响应速度方面的优越性。

关键词: 移动边缘计算; 深度强化学习; 车联网; 图神经网络(GNN); 任务卸载

基金项目: 国家自然科学基金(No.62372242, No.92267104)

中图分类号: TN929.5

文献标识码: A

文章编号: 0372-2112(2025)02-0329-15

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20240609

Efficient Task Offloading Based on Traffic Prediction in IoV-Enabled Edge Computing

XU Xiao-long^{1,2,3}, YANG Wei¹, YANG Chen-yi⁴, CHENG Yong¹, QI Lian-yong^{5*},
XIANG Hao-long¹, DOU Wan-chun⁶

(1. School of Software, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China;

2. Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing, Jiangsu 210044, China;

3. Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services, Nanjing, Jiangsu 210044, China;

4. School of Software, Southeast University, Nanjing, Jiangsu 211189, China;

5. College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China;

6. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210000, China)

Abstract: Vehicle edge computing combines mobile edge computing and the internet of vehicles(IoV) to offload the vehicle computing tasks from the cloud servers to edge servers, which effectively reduces the response time of IoV services. However, the irregular spatiotemporal distributions of traffic flows in vehicle networking will lead to the imbalance of computing load on the edge servers, which impacts real-time responsiveness of vehicle networking services. To address this issue, this paper proposes an efficient task offloading strategy based on traffic prediction in the vehicle edge computing. Specifically, a chebyshev graph weighted network (ChevWN) is designed to forecast traffic flow by fully leveraging connectivity and distance information between road segments. Next, a deep reinforcement learning-based binary task offloading algorithm (DBOA) is designed, which divides the binary task offloading decision process into two stages. Initially, a deep reinforcement learning approach is employed to derive the offloading strategies. Subsequently, a one-dimensional bi-end search

algorithm is utilized to determine the time slot allocation scheme that maximizes the overall computation rate, thereby reducing the complexity of the decision-making process. Finally, a large number of comparative experiments demonstrate the accuracy of ChebWN in predicting traffic flow and the superiority of DBOA in improving the response speed of vehicle services.

Key words: mobile edge computing; deep reinforcement learning; internet of vehicles; graph neural network; task offloading

Foundation Item(s): National Natural Science Foundation of China (No.62372242, No.92267104)

1 引言

智能交通系统(Intelligent Traffic System, ITS)是一种融合了人工智能(Artificial Intelligence, AI)技术的交通运维管理系统,通过实时收集和分析交通数据,为交通管理部门提供科学的决策支持,同时也为用户提供安全、可靠的智能交通服务。作为ITS的核心模块,车联网(Internet of Vehicles, IoV)通过连接车辆和道路基础设施,保障系统各部分之间的协作,实现交通管理和服务智能化。基于IoV技术,服务提供商实时收集并分析车辆速度、定位、周围环境等车载设备的实时传感数据,用于实现车辆导航和自动驾驶等服务^[1-3]。然而,上述服务的实现需要对大量传感器数据进行实时融合与高效分析,以保证控制决策的准确性和时效性,这对车联网的数据处理能力提出了更高要求^[4,5]。因此,在满足车辆计算需求的前提下,如何快速且精准地处理海量实时数据,是当前IoV发展亟待解决的难题。

由于车辆的计算资源有限,无法在规定时间内完成本地数据处理,这对车辆导航和自动驾驶等时延敏感型服务而言是不可接受的^[6]。传统解决方案通常将车辆生成的数据上传到云计算中心,依托其强大的计算和存储资源,实现任务卸载,并将处理结果反馈给车辆^[7]。然而,大量数据从网络边缘传输到云数据中心会显著增加网络负载,并可能导致网络拥塞^[8]。此外,车辆与云计算中心的长距离传输进一步增加了数据传输时延^[9]。为解决上述挑战,移动边缘计算(Mobile Edge Computing, MEC)依靠其独特的地理位置和计算资源,提供了一种新的解决方案。边缘服务器(Edge Server, ES)在网络边缘提供计算和存储资源,使车辆仅需将数据卸载到邻近的ES,即可实现高效处理^[10,11]。因此,基于MEC的IoV不仅能显著降低数据传输时延,还能有效缓解网络带宽压力^[12,13]。

尽管MEC提高了IoV服务的响应效率,但基于MEC的IoV计算任务卸载仍然面临诸多挑战。由于ES计算和存储资源有限,难以在短时间内处理大量卸载任务。因此,如何合理选择任务卸载的目标ES是保证IoV高效运行的关键。IoV中庞大的车辆数量以及动态多变的车辆位置和网络环境使得任务卸载问题更加复杂。同时,不同任务规模所需计算资源不同,ES需要根

据不同任务进行动态分配资源,以保证服务质量(Quality of Service, QoS)。

传统方法如动态规划^[14]虽然能够在理论上获得全局最优解,但其计算复杂度呈指数级增长,难以满足实际应用的需求。为降低计算复杂度,研究者们提出了启发式局部搜索^[15]和凸松弛^[16]方法。然而,这两种方法都需要大量的迭代才能达到令人满意的局部最优解,在动态多变的IoV边缘计算环境中,难以实现实时任务卸载决策。为了克服上述方法的局限性,本研究采用深度强化学习(Deep Reinforcement Learning, DRL)技术来应对IoV边缘计算环境下的任务卸载挑战。DRL通过不断地训练和更新策略网络来适应实时变化的环境,以最大化资源利用效率和服务质量^[17,18]。相比启发式和凸松弛方法,DRL方法能够在IoV边缘计算环境变化时迅速调整策略,避免重新求解优化问题的高计算开销,从而更好地满足MEC网络中的实时决策需求。此外,道路交通流量预测也在IoV领域得到广泛应用,利用交通网络的图结构进行交通流量预测,从而实现边缘资源的提前配置^[19]。

虽然现有研究已经实现了车联网中的流量预测并通过深度强化学习实现了任务卸载,但随着用户车辆数量逐年增加以及车载应用日益丰富,车辆任务卸载需求也在持续增加。与此同时,由于车辆的高速移动性和车联网状态的动态多变,某个路段的任务卸载请求可能会突然激增,从而导致边缘服务器超负荷运行^[20]。此外,缺乏高效的任务卸载方案将会导致边缘资源的不合理利用,进而产生巨大的时延,严重影响QoS^[21,22]。因此,如何准确预测交通流量,并合理分配边缘服务器资源以最大化任务卸载速率,从而满足车辆实时任务需求,成为当前亟须解决的挑战^[23]。

为了解决上述问题,本文设计了一种车联网边缘计算中高响应速度的任务卸载方案。综上所述,本文的主要研究贡献包含以下四个方面:(1)针对车联网的交通流量时空分布不均和边缘服务器易过载问题,提出了一种车联网边缘计算系统模型,以最大化所有车辆的计算速率与卸载速率加权和为目标;(2)提出了一种交通流量预测方法(Chebyshev graph Weighted Network, ChebWN),该方法引入影响因子矩阵并结合切比雪夫多项式,能够充分挖掘路段间连通性和距离关系,实现

更加精准的交通流量预测;(3)提出了一种基于经验回放机制的深度强化学习来进行IoV任务卸载的算法框架(DRL-based Binary task Offloading Algorithm, DBOA),将二元任务卸载的决策过程分为两个阶段,即首先通过深度强化学习得到卸载策略,再通过一维双端查找算法确定最大化总计算速率的时间片分配方案;(4)通过大量对比实验,验证了ChebWN方法在交通流量预测上的准确性以及DBOA在提升车联网服务响应速度方面的优越性。

2 相关工作

2.1 车联网边缘计算

当前,车联网边缘计算的研究聚焦于如何通过优化网络架构和资源管理,实现对车辆数据的高效处理和智能交通服务的实时响应。文献[19]提出了一种基于时空强化学习的方案,专门用于学习并预测车联网中资源需求的时空变化,从而进行边缘服务器的动态预留。文献[24]提出了一种适应性比特率多媒体流架构,利用MEC在车联网中的边缘缓存和通信能力,实现高质量的流媒体服务。该架构将多媒体文件分割为多个不同比特率级别的块,并提出了联合资源优化问题,旨在优化这些块的放置和传输,确保播放的流畅性和服务质量。文献[25]提出了一种边缘计算辅助的车联网高效协作数据共享方案,通过优化车辆间的通信和数据交换,解决大规模数据传输的挑战。利用图论算法和计算资源优化,提升内容分布的均衡性和资源利用率,并通过模拟实验证实方案的有效性和扩展性。文献[26]提出了一种结合边缘计算和区块链的智能驾驶资源交易架构,通过动态分配算法优化车联网中的资源管理和任务卸载。利用双重拍卖机制和智能合约,实现了资源分配的自动化和去中心化,有效提升车联网中资源利用率和用户服务满意度。文献[27]针对具有多种通信技术和多种任务类型的异构车联网边缘计算环境,提出了一种有效的资源分配策略,旨在最小化系统效用,同时满足超可靠低延迟通信要求。尽管上述方案对城市化带来的交通挑战提供了创新思路,如基于时空强化学习的资源预留、适应性比特率的多媒体流架构,以及结合边缘计算和区块链的资源交易架构等,但在动态网络环境下的可扩展性、实时适应性、通信效率等方面仍需进一步考量和优化。

2.2 车联网流量预测

交通流量预测在于通过精准预测未来各路段的交通流量数据,提前合理规划车联网设施的建设以及资源的分配。近年来,车联网流量预测的研究迅速发展,涵盖了从数据驱动模型到基于深度学习的复杂算

法,以实现更高精度和实时性的预测。文献[28]设计了基于卷积神经网络(Convolutional Neural Networks, CNN)的交通流量预测模型,其训练过程通过路段分组算法进行优化。实验结果表明,该方法在交通流量预测上具有较高的准确率。然而,上述研究忽视了路段之间对彼此交通流量变化的影响。为了充分挖掘路段之间的连通性,有些研究基于图神经网络(Graph Neural Network, GNN)模型进行交通流量预测。文献[29]设计了图卷积网络(Graph Convolutional Network, GCN),GCN是一种针对图结构数据的深度学习算法,是图数据的特征提取器,可以有效地同时学习图数据的特征和结构信息。文献[30]设计了基于图注意力网络(Graph Attention Networks, GAT)的细粒度交通流量预测方案,利用大规模轨迹数据预测车辆速度、队列长度等交通状况。大量实验数据证明了该方法在短期交通信息预测上表现出很高的准确性和稳健性。文献[20]设计了图加权卷积网络(Graph Weighted Convolution Network, GWCN),GWCN本质上是一个图节点接收相邻节点的特征信息,并依据它与所有相邻节点的位置关系,将这些信息乘以一定的权重,将其与自身的特征相融合,得到新提取的特征并传播到神经网络的下一层。然而,这些研究未能充分考虑路段间道路节点的相互依赖性以及它们之间的连通性和距离关系。本文提出的ChebWN方法引入影响因子矩阵并结合切比雪夫多项式,能够充分挖掘路段间连通性和距离关系信息,实现更加精准的交通流量预测。

2.3 车联网任务卸载方法

在车联网场景下,鉴于带宽资源的有限性,合理的任务卸载策略对于保障用户服务质量和降低系统损耗起着至关重要的作用^[31]。因此,车联网任务卸载的研究集中于开发更高效、更智能的卸载策略,以适应不断增长的计算需求和动态变化的网络环境。为了避免资源浪费并提高车联网任务的完成率,文献[32]提出了一种新的任务卸载方案,解决了移动边缘计算框架下5G异构车联网中车载任务冗杂的问题。文献[33]提出了基于动态规划的半分布式算法进行任务卸载,通过实验,他们证明了该方法显著降低了系统能耗和延迟。考虑到边缘计算资源分布不均和移动车辆之间对资源的竞争激烈的情况,文献[34]提出了一种基于博弈论的双阶段任务卸载机制,旨在提高车载任务完成率并促进车辆间的合作。为了最小化车联网系统的整体能耗,文献[35]将任务卸载分为流量重定向和卸载决策两个阶段,并提出了基于DRL的任务卸载策略。实验结果表明,与基线方法相比,该方法的平均能耗降低了约60%。文献[36]在雾计算环境中考虑了车辆移动性、任务优先级和用户服务质量,采用基于软 Actor-Critic

框架的深度强化学习算法来最小化系统时延. 文献[17]提出了一种融合深度强化学习的边缘计算策略, 专门针对车联网中的动态计算卸载问题, 以优化边缘服务器资源的利用并减少延迟. 尽管上述研究在资源利用优化和服务质量提升方面取得了一定成果, 但这些研究未充分考虑到有限带宽资源对任务卸载时延的影响, 且难以适应复杂的大规模动态网络环境.

综上所述, 当前大部分研究方案侧重于降低车联网系统的能耗或时延, 并没有充分考虑到系统的任务处理速率和用户服务质量. 此外, 当前的主流研究很少将车联网边缘资源预分配与任务卸载相结合, 以提高任务计算速率和优化用户服务质量.

3 系统模型

本节首先提出了一种车联网边缘计算系统模型. 基于该模型, 建立了系统通信和计算的数学模型. 最后, 本节将车联网边缘计算中的任务卸载问题抽象为一个整数规划问题. 系统模型中一些重要变量的符号表示及其含义如表1所示.

表1 系统模型中的重要符号及其含义

变量	含义
M	车辆数
W	时间片分配集合
H_i	从边缘服务器到 UV_i 的信道增益
λ	能量接收效率
P	能量传输功率
C	车辆的CPU计算速率
ρ	CPU处理单位比特数据所需转数
δ_i	计算能量效率系数
B	信道带宽
N_0	高斯噪声功率

3.1 车联网边缘计算系统模型

车联网边缘计算系统模型如图1所示, 该系统分为车联网用户层和边缘层两层. 车联网用户层包括 M 辆在道路上行驶的车辆, 表示为 $UV = \{UV_1, UV_2, \dots, UV_M\}$. 每辆车都装备有一根收发天线, 用于与边缘服务器通信和接收服务器传输的能量. 车联网用户层的主要任务是接收射频能量并将其存储在可充电电池中, 以用于本地计算或将任务卸载到边缘服务器. 车辆可以根据自身状态和任务需求选择最佳的计算方式. 边缘层由 N_e 个边缘服务器组成, 表示为 $ES = \{ES_1, ES_2, \dots, ES_{N_e}\}$, 这些服务器拥有充足的计算资源和通信资源. 每个边缘服务器都装备多根收发天线和稳定的能量源, 通过基于射频信号的无线能量传输(Wireless Power Transfer, WPT)技术为其覆盖范围内的

车辆传输能量. 边缘层的主要任务是接收和处理来自车联网用户层的任务请求, 并通过预测未来交通流量动态调整计算资源分配策略, 以确保服务的高效性和稳定性. 系统时间被分为 T_s 个连续且等长的时间片 T , 且每个时间片 T 被分为 $M+1$ 份, 每份在该时间片中所占比例的集合表示为 $W = \{w_0, w_1, w_2, \dots, w_M\}$. 其中, w_0 代表无线能量传输在时间片 T 中的时间占比, $w_i (1 \leq i \leq M)$ 表示车辆 UV_i 将任务卸载到边缘服务器在时间片 T 中的时间占比. 如果车辆 UV_i 选择进行本地计算, 则 w_i 的值为0. 所有车辆的卸载策略用集合 $\theta = \{\theta_1, \theta_2, \dots, \theta_M\}$ 表示, 其中每个元素的取值为0或1. $\theta_i = 1$ 表示车辆 UV_i 将计算任务卸载到边缘服务器, $\theta_i = 0$ 表示车辆 UV_i 在本地处理任务.

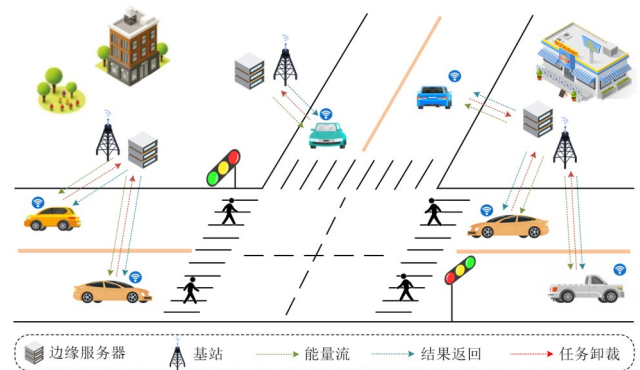


图1 车联网边缘计算系统模型图

3.2 系统通信模型

当多个车辆同时向边缘服务器卸载计算任务时, 可能会产生通信干扰. 为避免任务卸载时的通信干扰, 本研究采用了时分多路复用方法. 由于边缘服务器的计算能力比车辆的计算能力至少高出三个数量级, 并且返回给车辆的结果数据量远小于卸载的任务数据量, 因此, 边缘服务器处理数据及将结果数据下载到车辆所产生的时间延迟可以忽略不计.

H_i 表示边缘服务器与车辆 UV_i 之间的无线信道增益, 其取值与车辆 UV_i 接收到的能量以及边缘服务器与车辆 UV_i 之间的数据传输速率密切相关. 在同一时间片内, 无线信道增益 H_i 保持不变, 但是在不同时间片下无线信道增益 H_i 各不相同. 此外, 边缘服务器和车辆之间上、下行链路的无线信道增益是相同的. 为了合理分配边缘服务器的通信资源, 根据每个路段在某一时刻的交通流量占总流量的比例进行计算, 即第 i 个路段在时刻 t 下分配到的信道增益为

$$H_i^t = \left[\frac{f_i^t}{\sum_{n=1}^N f_n^t} \right] H_{\text{sum}} \quad (1)$$

其中, f_i^t 表示在时刻 t 下第 i 个路段的交通流量; N 表示路段总数; H_{sum} 表示信道增益总量.

由于各个路段预测交通流量的时间间隔较短, 如果每次交通流量变化时都重新进行无线信道增益 H_i 的分配, 就会产生额外的时延和能耗. 为避免这种情况, 本文引入了一个触发条件来确定何时进行无线信道增益 H_i 再分配. 两个相邻预测时刻 $t-1$ 和 t 之间的交通流量差为

$$|\Delta f_i^t| = |f_i^t - f_i^{t-1}| \quad (2)$$

从时刻 $t-\phi$ 到时刻 t 这 ϕ 段时间内的最大交通流量 $f_i^{t-\max}$ 与最小交通流量 $f_i^{t-\min}$ 的差值 F_i^t 被用作判断是否进行通信资源分配的指标. 当 t 小于 ϕ 时, F_i^t 的值等于时刻 t 的交通流量 f_i^t . 系统仅在 $|\Delta f_i^t|$ 大于 F_i^t 时才会对通信资源重新进行分配, 否则将保持目前的资源分配策略. F_i^t 可表示为

$$F_i^t = \begin{cases} f_i^t, & 0 < t < \phi \\ f_i^{t-\max} - f_i^{t-\min}, & t \geq \phi \end{cases} \quad (3)$$

在时间片 T 内, 车辆 UV_i 接收的能量

$$E_i = \lambda H_i P w_0 T \quad (4)$$

其中, λ 表示车辆的能量接收效率, 并且 $0 < \lambda < 1$; P 表示从边缘服务器到车辆的能量传输功率, λ 和 P 都是固定值.

根据香农定理, 可以通过提高信道带宽和增大信噪比来提升极限信道传输速率. 如果车辆 UV_i 将接收的能量全部用来进行任务卸载, 则其产生的最大信号功率为

$$SP_i^{\max} = \frac{\lambda H_i P w_0}{w_i} \quad (5)$$

因此, 车辆 UV_i 在任务卸载时的数据传输速率为

$$R_i^t = \mu B \cdot \log_2 \left(1 + \frac{\lambda P H_i^2 w_0}{w_i N_0} \right) \quad (6)$$

其中, B 是信道带宽; N_0 表示高斯噪声功率; μ 是一个介于 0 和 1 之间的常数. 由于香农公式计算得出的信道传输速率是理想条件下的最大数据传输率, 在现实场景中无法达到这一极限. 为使实验结果更接近实际情况, 将信道极限速率乘以一个固定值 μ 进行调整.

3.3 系统计算模型

如果车辆选择本地计算, 则会将其接收到的能量全部用于处理任务. 在车联网系统中, 每个车辆可以在接收能量的同时进行本地任务处理. 时间片 T 内, 车辆在本地处理的任务数据量 Y 可表示为

$$Y = \frac{CT}{\rho} \quad (7)$$

其中, T 表示车辆本地计算所产生的时间延迟; C 表示车辆的 CPU 计算速度, 单位为每秒的转数; ρ 表示 CPU 每处理单位比特数据所需的转数.

车辆进行本地计算的速率为

$$R = \frac{C}{\rho} \quad (8)$$

车辆 UV_i 本地计算产生的能耗不能超过 E_i , 因此, 车辆 UV_i 的处理器计算速率 C_i 必须满足约束条件

$$\partial_i C_i^3 T \leq E_i \quad (9)$$

其中, ∂_i 为计算能量效率系数. 由此可得, 车辆 UV_i 处理器的最大计算速率为

$$C_i = \left(\frac{E_i}{\partial_i T} \right)^{\frac{1}{3}} \quad (10)$$

车辆 UV_i 的最大本地计算速率为

$$R_i^{\text{loc}} = \frac{1}{\rho} \cdot \left(\frac{\lambda H_i P w_0}{\partial_i} \right)^{\frac{1}{3}} \quad (11)$$

在本研究中, 边缘服务器进行任务处理以及将结果数据返回给车辆所产生的时延可以忽略不计. 假设车辆 UV_i 将任务完全卸载到边缘服务器进行计算, 则在任务卸载过程中, 车辆 UV_i 的计算速率 R_i^{off} 等于车辆卸载时的数据传输速率 R_i^t . 因此, 车辆 UV_i 在任务卸载时的计算速率 R_i^{off} 可表示为

$$R_i^{\text{off}} = \mu B \cdot \log_2 \left(1 + \frac{\lambda P H_i^2 w_0}{w_i N_0} \right) \quad (12)$$

3.4 问题定义

在本系统中, 最终的优化目标是最大化所有车辆的计算速率与任务卸载速率的加权和. 因此, 该优化问题可以被建模为

$$\max_{\theta, W} \sum_{i=1}^M \eta_i [\theta_i \cdot R_i^{\text{off}} + (1 - \theta_i) \cdot R_i^{\text{loc}}] \quad (13)$$

满足:

$$\sum_{i=0}^M w_i \leq 1 \quad (14)$$

$$\theta_i \in \{0, 1\}, i \in [1, M] \quad (15)$$

$$\eta_i \in \{\alpha, \beta\}, i \in [1, M] \quad (16)$$

其中, 式(14)表示在任意时间片内, 所有车辆将任务卸载到边缘服务器的时间占比之和不超过 1; 式(15)表示车辆在每个时间片内只能选择本地处理任务或将任务卸载到边缘服务器进行处理; 式(16)表示车辆 UV_i 的任务权重 η_i 只能取值为超参数 α 和 β , α 和 β 分别为车辆进行任务卸载和本地计算的计算速率所占权重的超参数, 通过设定固定权重, 可以明确地区分本地计算与任务卸载的优先级. 由于式(13)目标函数中包含了非凸的二值变量 θ_i 和离散取值的权重 η_i , 并且目标函数涉及这些变量的线性组合, 所以直接求出 θ 和 W 是一个结合了整数规划的非凸函数求解问题.

4 基于交通流量预测的车联网任务卸载

本节首先给出基于交通流量预测的车联网任务卸

载方案的总体架构,然后对 ChebWN 和 DBOA 方法进行详细说明.

4.1 总体架构

基于交通流量预测的车联网任务卸载方案包含两

个阶段,如图2所示.第一阶段是交通流量预测,根据历史交通流量预测未来交通流量.第二阶段是车联网边缘计算任务卸载,依据第一阶段的未来交通流量预测结果,优化车联网任务卸载决策.

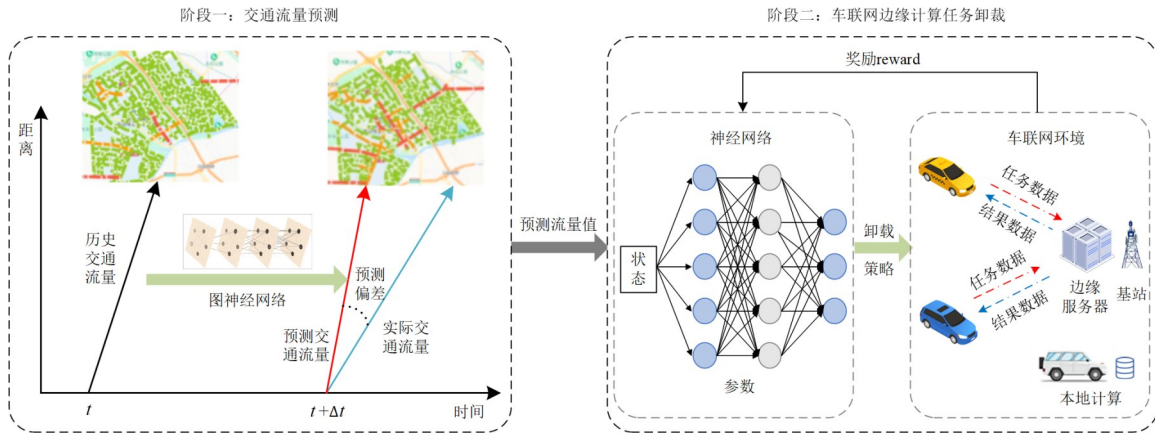


图2 基于交通流量预测的车联网任务卸载方案的总体架构

具体来说,第一阶段是用于交通流量预测的 ChebWN 模块,主要功能是根据历史交通数据,预测未来各个路段的交通流量,确保边缘服务器可以提前分配服务器资源,这样既有效降低边缘服务器过载的风险,又保证了边缘服务器的计算资源被充分利用.第二阶段是基于 DRL 的二元任务卸载算法 DBOA 模块,主要功能是根据第一模块的交通流量预测结果,优化车联网任务卸载决策来最大化整个车联网系统的计算速率.基于当前的车联网状态,DBOA 模块会选择一个动作,即决定将某计算任务卸载到边缘服务器上,或者在车辆本地计算.每次动作执行后,系统会反馈一个奖励值 reward,用于评估该卸载动作的效益. DBOA 模块的奖励值基于本地计算速率、任务卸载速率和任务卸载决策计算. DBOA 模块的目标是通过不断调整和优化卸载决策,找到最优的二元任务卸载方案 θ 和时间片分配方案 W ,来最大化整个车联网系统的总计算速率,从而提升车联网系统服务响应速度.

4.2 交通流量预测方法 ChebWN

通过对历史交通流量数据进行预测分析,可以提前了解未来某段时间内不同节点的计算需求,从而合理预分配边缘服务器资源,有效避免某些节点过载或边缘资源闲置.在车联网场景中,道路节点是路段或十字路口,道路节点之间的距离为彼此到达的最短距离.路段的交通流量会受到相邻节点的很大影响,特定交通节点的交通流量与历史交通流量密切相关,也与其他节点的连通性密切相关.只有路段的历史交通流数据,不足以准确预测未来的交通流,提取交通拓拓扑网络的空间信息和节点间的依赖关系对于准确预测道路节点的交通流量至关重要.用图数据结构来表示车联网交

通流量数据,可以很好地描述道路节点之间复杂的邻接关系和距离关系^[37]. GNN 可以挖掘道路节点之间的距离关系和连通性关系,并且在非欧几里得空间中进行数据分析和基于实体之间关系的特征提取是 GNN 的强项.因此, GNN 模型适用于预测道路节点的交通流量.

由于道路节点之间的关系是无向的,并且所有道路节点都是连通的,因此整个区域被构造为无向连通图.本文用 $\text{Graph} = (V_S, E_S)$ 表示由道路节点组成的图结构.其中, V_S 表示道路节点的集合, E_S 表示边的集合,节点的数量为 N .每个道路节点的特征值是从路段的历史交通流量数据中提取出的.道路节点之间的连通关系可以被邻接矩阵 $A (A \in \mathbb{R}^{N \times N})$ 表示,其元素 $a_{ij} (1 \leq i, j \leq N)$.如果节点 i 与节点 j 相邻,则 a_{ij} 等于道路节点 i 和道路节点 j 之间距离的倒数,否则 a_{ij} 等于 0.此外,定义一个对角矩阵 $D (D \in \mathbb{R}^{N \times N})$ 为度矩阵,度矩阵 D 的第 i 个对角元素 d_{ii} 表示节点 i 的度数(即与该节点相连的边数量).

定义一个影响因子矩阵 $S (S \in \mathbb{R}^{N \times N})$,用于衡量相邻节点的交通流量信息对道路节点流量的影响.道路节点的未来交通流量值与其邻接节点的距离密切相关,两个节点之间的距离越近,它们对彼此交通流量的影响也越显著.影响因子矩阵 S 的元素 s_{ij} 表示道路节点 j 对道路节点 i 的重要性系数, s_{ij} 可表示为

$$s_{ij} = \hbar \cdot \frac{a_{ij}}{\sum_{j=1}^N a_{ij}}, \quad \forall i, j \in \{1, 2, \dots, N\} \quad (17)$$

其中, \hbar 表示一个预设的超参数; s_{ij} 值越大,说明道路节点 j 对道路节点 i 的交通流量影响越显著.但在通常情

况 $s_{ji} \neq s_{ij}$, 因为节点 i 和节点 j 相互影响程度不同. 影响因子矩阵 \mathbf{S} 描述了道路节点间的距离关系对交通流量的影响.

但影响因子矩阵 \mathbf{S} 的主对角线上的元素 $s_{ii} (1 \leq i \leq N)$ 全为 0, 这会导致 GNN 在向前传播过程中忽略节点自身特征对流量预测的影响. 为了解决这个问题, 定义一个影响因子矩阵 $\mathbf{S}^+ (\mathbf{S}^+ \in \mathbb{R}^{N \times N})$ 为

$$\mathbf{S}^+ = \mathbf{S} + \mathbf{I} \quad (18)$$

其中, 矩阵 $\mathbf{I} (\mathbf{I} \in \mathbb{R}^{N \times N})$ 是主对角线元素全为 1, 其他元素全为 0 的矩阵. $\mathbf{S} + \mathbf{I}$ 相当于为 Graph 中的每个节点增加了一条自环边, 从而考虑节点自身对流量预测的影响.

然而影响因子矩阵 \mathbf{S}^+ 仅通过道路节点之间的距离关系来衡量节点之间的相互影响程度, 未考虑如何整合和聚合邻居节点的信息, 以更新节点的代表或特征向量. 在 ChebWN 网络中, 信息聚合的核心思想是利用图的拉普拉斯矩阵进行近似卷积操作. 具体来说, ChebWN 算法通过对图的拉普拉斯矩阵进行谱分解来获得相应的特征值和特征向量, 随后利用这些特征向量构造一组切比雪夫多项式. 这种基于谱方法的卷积操作可以有效地捕捉图结构中节点之间的局部和全局关系, 从而提升 GNN 在复杂交通网络等领域的表现和预测能力.

拉普拉斯矩阵是描述图结构的重要工具, 常用于图信号处理以捕捉节点之间的连通性, 在 GCN 和 GNN 等任务中具有重要意义. 归一化的拉普拉斯矩阵 \mathbf{L} 有助于在图信号处理中捕捉节点之间的相对影响力, 同时避免

度数较大的节点对结果产生过大影响, 从而更好地保持图的结构信息^[38]. 归一化的拉普拉斯矩阵 \mathbf{L} 可表示为

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (19)$$

其中, \mathbf{I}_n 是与矩阵 \mathbf{A} 同大小的单位矩阵. ChebWN 网络的卷积操作是通过切比雪夫多项式 $T(x)$ 对输入的图信号进行变换, 这相当于在谱域上对图信号进行滤波操作, 同时保持卷积的局部性. 通过堆叠多个切比雪夫卷积层, ChebWN 网络在图中对节点进行多层次的信息聚合和特征提取. Q 阶切比雪夫多项式:

$$T_Q(x) = 2xT_{Q-1}(x) - T_{Q-2}(x) \quad (20)$$

其中, $T_{Q-1}(x)$ 和 $T_{Q-2}(x)$ 分别是 $Q-1$ 阶和 $Q-2$ 阶的切比雪夫多项式. 当阶数 Q 等于 0 时, $T_0(x) = 1$. 当 Q 等于 1 时, $T_1(x) = x$.

图 3 展示了 ChebWN 网络前向传播的流程. ChebWN 网络利用两层 ChebConv 来提取交通节点的特征, 并使用两层 ReLU 激活函数对这些特征进行非线性变换. 第 l 层 ChebConv 的前向传播过程可表示为

$$\mathbf{X}^{l+1} = \text{ReLU} [T(\mathbf{L})\mathbf{S}^+ \mathbf{X}^l \mathbf{W}^l + \mathbf{b}^l] \quad (21)$$

其中, \mathbf{X}^l 和 \mathbf{X}^{l+1} 分别表示第 l 层和第 $l+1$ 层 ChebConv 提取的特征值; \mathbf{W}^l 表示从第 l 层 ChebConv 前向传播的参数矩阵; $T(\mathbf{L})$ 指的是切比雪夫多项式函数; \mathbf{b}^l 表示偏置项. 该算法使用的损失函数是均方误差 (Mean Squared Error, MSE), 优化器为自适应矩估计 (Adaptive moment estimation, Adam).

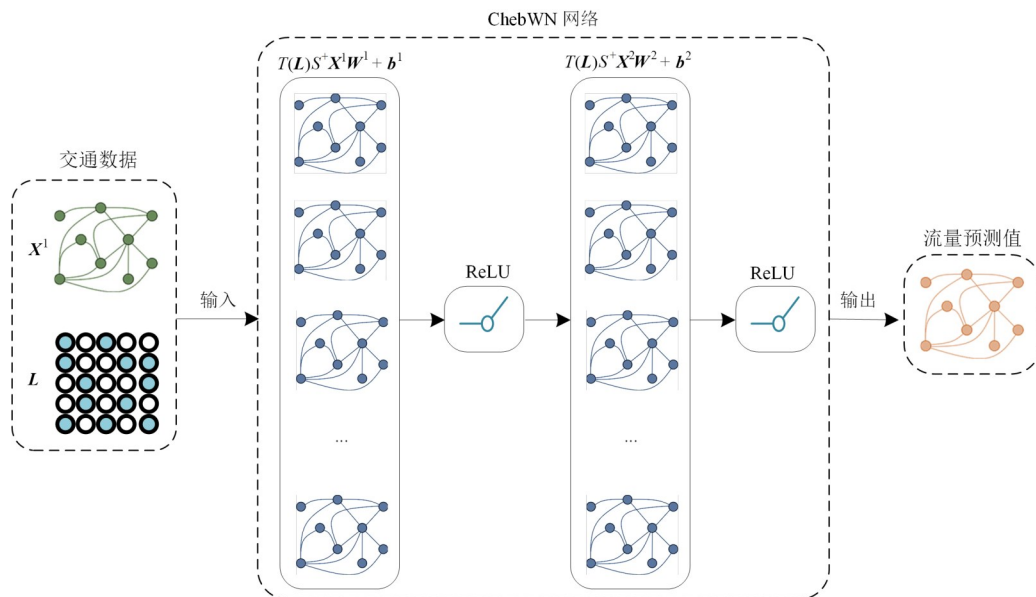


图 3 交通流量预测方法 ChebWN 网络图

算法 1 详细描述了车联网交通流量预测方法 ChebWN. 在算法 1 中, ChebWN 的每一层先对特征数据进行信息聚合, 然后将结果乘以神经网络的权重完成

进一步的特征提取. 整个训练过程的时间复杂度取决于邻接矩阵的度数和 ChebWN 每层神经网络的输入神经元和输出神经元的个数. C_{in}^i 和 C_{out}^i 分别是第 i 层的输

入神经元和输出神经元的个数, ChebWN 的总层数为 J . 因此 ChebWN 的时间复杂度为 $O\left(\sum_{i=1}^J N^4 C_{in}^i C_{out}^i\right)$. ChebWN 的空间复杂度主要由神经网络的参数总数决定, 表示为 $O\left(\sum_{i=1}^J C_{in}^i C_{out}^i\right)$.

算法 1 车联网交通流量预测方法 ChebWN

输入: 交通数据 X^t , 邻接矩阵 A , 度矩阵 D , 切比雪夫阶数 Q

输出: 预测的交通流量值 predict_value

1. 随机初始化 ChebWN 的参数 ω, b ;
2. FOR $i = 0$ to N
3. FOR $j = 0$ to N
4. 根据式(17)计算 s_{ij} ;
5. END FOR
6. END FOR
7. 根据式(18)计算 S^+ ;
8. 根据式(19)计算 L ;
9. FOR $q = 0$ to Q
10. 根据式(20)计算 $T_q(L)$;
11. END FOR
12. FOR epoch = 1 to epochs
13. 将 x^1, A, D 和 $T(L)$ 输入到 ChebWN;
14. FOR $l = 1$ to 2
15. 使用式(21)计算 X^{l+1} ;
16. END FOR
17. 计算 loss = MSE(predict_value, true_value);
18. 进行反向传播, 并使用 Adam 优化器更新网络参数;
19. END FOR
20. 输出 predict_value;

4.3 基于 DRL 的二元任务卸载方法 DBOA

本文的目标是同时找到一个最优的二元任务卸载方案 θ 和时间片分配方案 W , 使整个车联网边缘计算系统能得到最大的总计算速率. 由于直接求出 θ 和 W 是一个结合了整数规划的非凸函数求解问题, 以有限的算力很难求解出来. 因此, DBOA 算法把原问题分解为两个子问题, 即先确定任务卸载方案 θ , 并在已知 θ 的基础上通过一个一维双端查找算法求解出能获得最大化总计算速率的时间片分配方案 W . 由于所有可能的策略数量极其庞大, 而深度强化学习凭借其强大的感知能力和决策能力, 非常适合用来解决寻找合适的任务卸载策略. 图 4 是 DBOA 二元任务卸载算法框架图. 本小节首先介绍车联网边缘计算任务卸载的 DBOA 算法设计, 其次描述该算法训练流程.

4.3.1 DBOA 算法设计

(1) 状态空间和动作空间

本文车联网系统的状态主要由各个信道的信道增

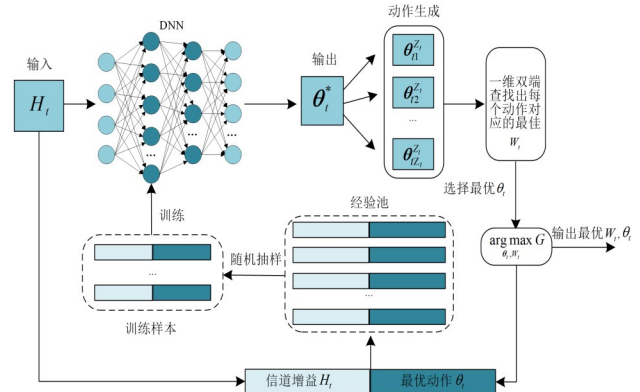


图 4 DBOA 二元任务卸载算法框架图

益决定, 即在时间片 t 下的系统状态 s_t 由边缘服务器及其覆盖范围内所有车辆之间的信道增益值集合 H_t 决定. 信道增益在同一个时间片中不发生变化, 但在不同的时间片下, 其值各不相同. 在时间片 t 下的车联网系统状态 s_t 可表示为

$$s_t = H_t \in \{H_i | i = 1, 2, \dots, M\} \quad (22)$$

每个车辆都采用二元卸载策略, 即车辆要么将任务全部卸载到边缘服务器进行处理, 要么全部在本地处理. 因此, 在时间片 t 下, 系统的动作空间 a_t 是一个 M 维向量, 其元素取值为 0 或 1. 取值为 0 表示车辆在本地处理车载任务, 取值为 1 表示车辆将任务卸载到边缘服务器进行处理. 由于每个车辆都有两种可能的卸载策略, 整个动作空间的大小为 2^M . 动作空间 a_t 可表示为

$$a_t = \{0, 1\}^M \quad (23)$$

(2) 动作生成和奖励函数

在车联网系统中, 所有可能的任务卸载方案总数为 2^M . 因此随着车辆数量的增加, 卸载方案总数呈指数级增长. 如果将所有的方案都进行计算, 会对有限的计算资源造成巨大的负担, 导致产生巨大的时间损耗和能量损耗. 此外, 只有部分动作有利于使 DBOA 达到全局最优. 因此, 设计一个动作生成算法 AG 来生成 Z 个动作. Z 是一个远小于 2^M 的整数, 其取值范围为 $[1, M]$.

本文设计了一种动态调整 Z 的方法, 其中 Z_t 表示在第 t 个时间片动作生成算法 AG 生成的动作数量. 初始时, 将 Z_t 设置为 M . 随后, 每隔 V 个时间片, 对 Z_t 进行更新. 如果在过去 V 个时间片的平均奖励值 $\overline{\text{reward}}$ 超过了预定的阈值 δ_1 , 则将 Z_t 更新为 $Z_t - 1$, 但不小于 1; 如果在过去 V 个时间片的平均奖励值 $\overline{\text{reward}}$ 小于预定的阈值 δ_2 , 则将 Z_t 更新为 $Z_t + 1$, 但不超过 M ; 否则将 Z_t 保持为 Z_{t-1} . 这种动态调整方法在提升系统执行效率的同时避免了频繁的调整. Z_t 的更新方式为

$$Z_t = \begin{cases} \max(Z_{t-1} - 1, 1), & \text{if } t > 1 \text{ and } t \bmod V = 0 \text{ and } \overline{\text{reward}} > \delta_1 \\ \min(Z_{t-1} + 1, M), & \text{if } t > 1 \text{ and } t \bmod V = 0 \text{ and } \overline{\text{reward}} < \delta_2 \\ Z_{t-1}, & \text{otherwise} \end{cases} \quad t = 2, 3, \dots, T_s \quad (24)$$

DBOA 算法的隐藏层使用 ReLU 作为激活函数,输出层使用 Sigmoid 函数,以保证神经网络的输出值在 0~1,其输出结果由 θ_i^* 表示,其元素 $\theta_{ii}^* \in \{\theta_{ii}^* | 0 \leq \theta_{ii}^* \leq 1, i = 1, 2, \dots, M\}$. 将 θ_i^* 输入动作生成算法 AG,生成 Z_t 个任务卸载方案 $\theta_{ii}^{Z_t}$,即

$$\theta_{ii}^{Z_t} = \left\{ \theta_{ii}^{Z_t} | \theta_{ii}^{Z_t} \in \{0, 1\}^M, i = 1, 2, \dots, Z_t \right\} \quad (25)$$

动作生成算法 AG 分为两个步骤. 首先设定一个阈值 $\tau \in (0, 1)$,将 θ_i^* 中所有大于 τ 的值转换为 1,所有不大于 τ 的值转换为 0,这样依据 θ_i^* 获得了第一个任务卸载方案 $\theta_{ii}^{Z_t}$

$$\theta_{ii}^{Z_t}(i) = \begin{cases} 1, & \text{if } \theta_{ii}^* > \tau \\ 0, & \text{if } \theta_{ii}^* \leq \tau \end{cases} \quad i = 1, 2, \dots, M \quad (26)$$

第 z ($2 \leq z \leq Z_t$) 个卸载方案 $\theta_{ii}^{Z_t}$ 的每一个元素 $\theta_{ii}^{Z_t}(i)$ 是根据 θ_{ii}^* 与 θ_{ii}^* 的第 $z-1$ 个元素 $\theta_{ii}^{Z_{t-1}}$ 以及 τ 的大小关系获得的. 如果 θ_{ii}^* 大于 $\theta_{ii}^{Z_{t-1}}$,则 $\theta_{ii}^{Z_t}(i)$ 的值就取 1; 如果 θ_{ii}^* 小于 $\theta_{ii}^{Z_{t-1}}$,则 $\theta_{ii}^{Z_t}(i)$ 的值就取 0. 如果 θ_{ii}^* 等于 $\theta_{ii}^{Z_{t-1}}$,则要分以下两种情况,若 θ_{ii}^* 大于 τ ,则 $\theta_{ii}^{Z_t}(i)$ 的值取 0; 若 θ_{ii}^* 小于 τ ,则 $\theta_{ii}^{Z_t}(i)$ 的值取 1. 通过减少需要考虑的卸载数量,降低 DBOA 算法计算复杂度,节省系统计算资源. 以上过程表示为

$$\theta_{ii}^{Z_t}(i) = \begin{cases} 1, & \text{if } \theta_{ii}^* > \theta_{ii}^{Z_{t-1}} \text{ or } \theta_{ii}^* = \theta_{ii}^{Z_{t-1}} \text{ and } \theta_{ii}^* < \tau \\ 0, & \text{if } \theta_{ii}^* < \theta_{ii}^{Z_{t-1}} \text{ or } \theta_{ii}^* = \theta_{ii}^{Z_{t-1}} \text{ and } \theta_{ii}^* > \tau \end{cases} \quad (27)$$

$$i = 1, 2, \dots, M$$

通过上述方法,在 t 时间片下,DBOA 的输出 θ_i^* 生成了 Z_t 个不同的卸载方案. DBOA 的奖励值则是这 Z_t 个卸载方案中能获得的最大加权总计算速率值. 在 t 时间片下的奖励值由以下公式计算得到:

$$\text{reward}_t = \max_{z=1,2,\dots,Z_t} \eta_i \cdot \left\{ \theta_{ii}^{Z_t}(i) \cdot R_i^{\text{off}} + (1 - \theta_{ii}^{Z_t}(i)) \cdot R_i^{\text{loc}} \right\} \quad (28)$$

其中, η_i 表示第 i 个车辆的权重系数; $\theta_{ii}^{Z_t}(i)$ 表示第 z 个卸载方案中第 i 个车辆的卸载决策; R_i^{off} 和 R_i^{loc} 分别表示车辆 i 在进行任务卸载和本地处理情况下的计算速率.

4.3.2 DBOA 算法训练过程

不同于其他主流的深度强化学习模型,DBOA 算法以复杂多变的环境状态数据作为输入并更新参数,通过经验回放机制来更新网络参数 ω ,主要的训练样本是由 t 时间片下的信道增益 H_t 和 DBOA 基于目前的神经网络参数作出的动作决策 $\Pi_\omega(H_t)$. 在开始训练时,首先随机初始化 DBOA 网络参数 ω ,接着将所有时间片下的信道增益数据输入到网络中,生成相应的最佳卸载方案 $\Pi_\omega(H_t)$,并将二元组 $\langle H_t, \Pi_\omega(H_t) \rangle$ 存入经验池中,作

为训练样本. 每过 X 个时间片,就从经验池中抽取 BS 个样本来进行网络参数的更新,直到 DBOA 的决策达到局部最优. 当经验池中的样本数达到上限时,就把最老的样本淘汰,用最新样本代替. 本算法用交叉熵损失当作损失函数,Adam 作为更新神经网络参数的优化器. 采用这样的经验回放机制来训练神经网络,既大大降低计算复杂度,又能打破训练样本之间的关联性,使得 DBOA 作出的决策更加接近全局最优解. 算法 2 详细描述了 DBOA 算法.

算法 2 DBOA 算法

输入: t 时间片下的信道增益值 $H_t, t = 1, 2, \dots, T$

输出: 每个时间片的最佳任务卸载方案和时间片分配方案

1. 随机初始化 DBOA 神经网络的参数 ω
2. FOR $t = 1$ to T
3. IF $t \bmod X = 0$ then
4. 从经验池中抽取 BS 个样本;
5. 通过交叉熵损失计算损失值并利用 Adam 更新网络;
6. END IF
7. 将 H_t 输入神经网络中得到 θ_i^* ;
8. 根据式(24)计算 Z_t ;
9. 根据式(26)计算 θ_{ii}^* ;
9. FOR $z = 2$ to Z
10. FOR $i = 1$ to M
11. 使用式(27)计算 $\theta_{ii}^{Z_t}(i)$;
12. END FOR
13. END FOR
14. 利用一维双端查找得到每个 $\theta_{ii}^{Z_t}$ 对应的最佳时间片分配方案;
15. 根据公式(28)从 $\theta_{ii}^{Z_t}$ 中选取最优的卸载方案;
16. 将 H_t 与当前最优卸载方案作为一个二元组放入经验池中;
17. IF 经验池已满 then
18. 淘汰最老的样本,用最新样本代替;
19. END IF
20. END FOR
21. 将未来所有时间片的信道增益数据输入到 DBOA;
22. 得到每个时间片的最佳任务卸载方案和时间片分配方案;

5 实验

本节首先对实验参数设置进行简单说明. 接着,通过和其他主流 GNN 模型的对比实验验证了 ChebWN 算法预测交通流量数据的准确性. 最后,通过大量实验证明了 DBOA 在提高车联网系统的加权总计算速率方面的有效性.

5.1 实验参数设置

本文在 Python3.8 环境下使用 Pytorch 框架进行

ChebWN 和 DBOA 的仿真实验. 本实验的关键参数设置如表 2 所示.

表 2 参数设置表

参数描述	值
路段节点数 N	120
车辆数量 M	{5, 10, 20, 30}
能量传输功率 P/W	3
能量接收效率 λ	0.7
计算能量效率系数 ∂_i	2×10^{-26}
信道带宽 B/MHz	2×10^{10}
权重参数 α	1.5
权重参数 β	1
信道速率损失率 μ	0.9
高斯噪声功率 N_0	10^{10}

5.2 ChebWN 准确度分析

本实验使用的数据集涵盖了两周内 307 个道路节点的交通流量数据, 数据来自加州交通局性能测量系统 (<https://pems.dot.ca.gov>). 本研究将提出的 ChebWN 的预测准确度与 GCN^[29]、GAT^[30] 和 GWCN^[20] 进行比较.

ChebWN、GCN、GAT 和 GWCN 用于预测特定路段节点在 24 h 内每 5 min 的交通流量. 随后, 使用平均绝对误差 (Mean Absolute Error, MAE)、均方根误差 (Root Mean Squared Error, RMSE) 和平均绝对百分比误差 (Mean Absolute Percentage Error, MAPE) 作为衡量模型预测准确度的指标, 这三项指标越小, 模型的预测准确度越高. 这三项指标的计算式为

$$\text{MAE} = \frac{1}{T_s} \sum_{t=1}^{T_s} |f^t - f_{\text{true}}^t| \quad (29)$$

$$\text{RMSE} = \left[\frac{1}{T_s} \sum_{t=1}^{T_s} (f^t - f_{\text{true}}^t)^2 \right]^{\frac{1}{2}} \quad (30)$$

$$\text{MAPE} = \frac{1}{T_s} \sum_{t=1}^{T_s} \left| \frac{f^t - f_{\text{true}}^t}{f_{\text{true}}^t} \right| \quad (31)$$

其中, T_s 表示所有预测时刻的数量; f^t 表示 t 时刻下交通流量的预测值; f_{true}^t 表示 t 时刻下交通流量的真实值.

图 5 描述了 ChebWN、GCN、GAT 和 GWCN 在训练过程中的损失值的变化趋势, 对这四种模型的收敛性能进行了对比. 四条曲线均在前五个训练周期中快速下降, 在第五个周期后, 损失值达到拐点并趋于平稳, 这意味着四个模型均已收敛. 在损失稳定后, ChebWN 的损失值相比 GCN、GAT 和 GWCN 都是最小的. 这表明 ChebWN 在此次训练任务中表现出了更优的收敛性能和更低的误差率.

图 6 描述了第 110 号路段节点被用来进行未来 24 h 内每隔 5 min 的交通流量预测, 对四种实验模型的流量

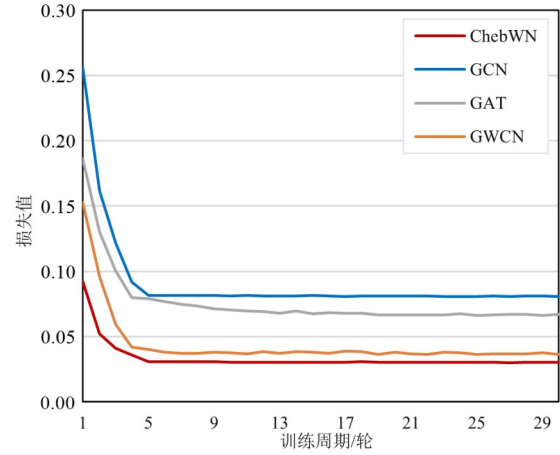


图 5 ChebWN、GCN、GAT 和 GWCN 收敛性能对比

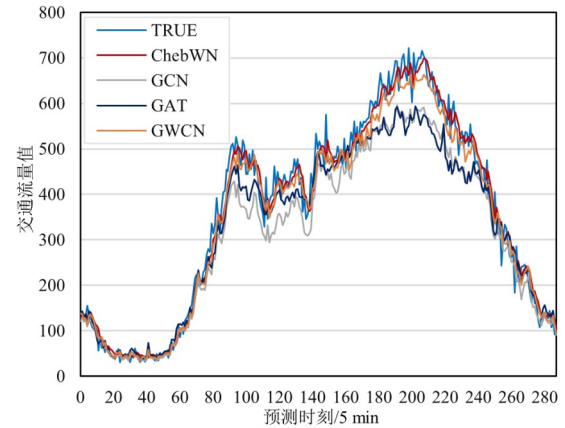


图 6 第 110 号路段节点的交通流量预测结果

值预测结果进行对比. 其中, 蓝色的曲线代表第 110 号路段真实的交通流量值, 红色曲线代表使用 ChebWN 模型预测的交通流量值. 由图可见, 红色曲线表示的 ChebWN 预测结果与真实值曲线拟合度最高. 这初步表明, ChebWN 在预测准确性方面优于其他四种模型. ChebWN 通过引入影响因子矩阵并使用切比雪夫多项式来逼近图信号, 能够充分挖掘路段间连通性和距离关系信息, 尤其在交通流量预测这种具有空间相关性的任务中展现了较高的准确性. 虽然 GCN 能够捕捉图结构数据的特征, 但其预测结果曲线与真实值相比存在一定误差, 特别是在交通流量波动较大的时段, 拟合度不如 ChebWN 模型.

图 7 展示了在测试集上四种实验模型最终取得的 MAE、RMSE 和 MAPE 的结果对比. ChebWN 在这三种交通流量预测误差度量指标中均取得了最小值, 这也进一步说明 ChebWN 预测交通流量的误差小, 准确性高. ChebWN 在 MAE 指标上取得了最小值, 表明其预测误差较小, 能够更稳定地提供接近真实值的预测结果. ChebWN 在 RMSE 指标上同样取得了最小值, 表明其不

仅误差小,而且大误差较少,预测结果更为平滑. ChebWN 在 MAPE 指标上也取得了最小值,表明其预测值与真实值的相对误差较低,能够较好地适应流量波动较大的情况.

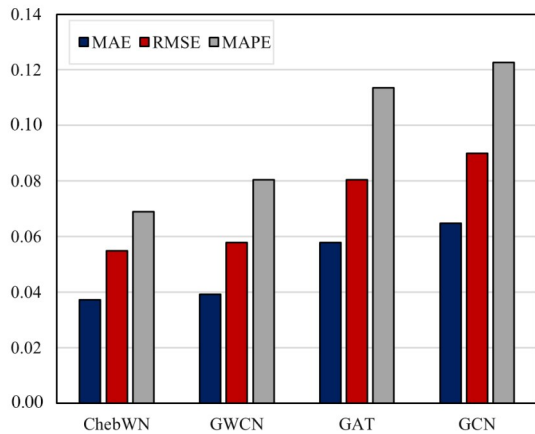


图7 MAE、RMSE和MAPE的结果对比

5.3 DBOA-AG 性能分析

本实验使用 30 000 个时间片的信道增益数据进行训练. 在车辆数量分别是 5、10、20、30 的情况下, 将 DBOA-AG 与 DPOM (DRL-based Partial Offloading Method)、DBOA-KNN、本地计算 (Local Computing, LC) 和卸载计算 (Offloading Computing, OC) 四种卸载算法进行对比分析. 为了区分 DBOA-KNN, 下文将本文提出的算法记为 DBOA-AG. DPOM、DBOA-KNN、LC 和 OC 概述如下.

(1) DPOM^[39]: 该方法在 DBOA 框架下, 采用深度强化学习技术来最大化无线供电边缘计算网络的总计算速率. 该方法依据神经网络的输出生成等间隔的 Z 个卸载方案.

(2) DBOA-KNN^[40]: 该方法在 DBOA 框架下, 使用 K 最邻近算法 (K -Nearest Neighbor, KNN) 来根据神经网络的输出生成最邻近的 Z 个卸载方案.

(3) LC: 所有的车辆都选择将任务在本地进行处理.

(4) OC: 所有的车辆都选择将任务卸载到边缘服务器进行处理.

在 DBOA 的训练过程中选取了 0.01、0.001、0.000 1 和 0.000 01 这四个数量级的学习率进行训练, 图 8 描述了在 3 000 个训练周期中的损失值变化情况. 在四种不同的学习率下, DBOA-AG 的损失值随着训练 epoch 的增加而减少, 这表明模型在训练过程中逐渐优化和收敛. 当学习率为 0.000 01 时, 整个训练周期的损失值保持较高水平, 这意味着学习率过小导致模型训练缓慢, 难以有效优化模型参数, 尽管最终也能趋于稳定, 但相

比于 0.01 和 0.001 的学习率而言, 训练效率较低, 因此不宜作为最优学习率的选择. 当学习率为 0.000 1 时, 损失值的收敛速度相较于 0.01 和 0.001 较为缓慢, 也不被考虑为最佳选择. 当学习率为 0.01 和 0.001 时, 损失值在前 400 个训练周期内急剧下滑, 并且最终趋于稳定. 这表明这两个学习率能够较快地优化模型, 在较短的训练周期内达到较低的损失值. 此外, 学习率为 0.001 时的最终损失值小于学习率为 0.01 时的最终损失值. 然而考虑到学习率过小可能会导致过拟合, 还需要进一步的对比试验来确认最优学习率.

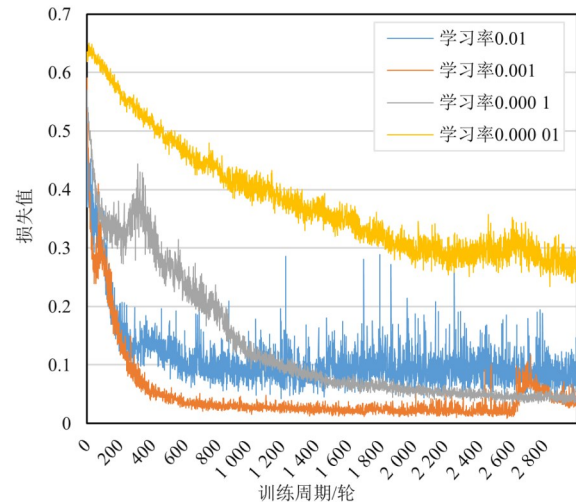


图8 不同学习率下 DBOA 在 3 000 训练周期中的损失值变化

由于每个时间片的信道增益数据不同, 得到的加权总计算速率也天差地别, 不容易看出其性能的变化趋势, 因此通过观察 DBOA-AG 得到的加权总计算速率和通过枚举得到的实际上的最大加权总计算速率的比值变化情况来比较不同学习率的性能, 并将这项指标称为归一化加权总计算速率 G^* . 图 9 描述了在学习率为 0.01 和 0.001 的前 800 个训练周期内, DBOA-AG 的归一化加权总计算速率 G^* 的变化情况. 归一化加权总计算速率 G^* 在前 400 个周期内波动较大, 这是由于训练过程中参数调整的不稳定性和初始阶段的收敛性差异造成的, 最终都稳定在了 1.0 左右. 但是, 学习率为 0.01 的收敛速度比学习率 0.001 更快, 且 G^* 更快地到达最优值. 因此, 在后续的对比如中选择 0.01 作为最优学习率.

图 10 展示了在车辆数量 M 为 10 的情况下, 不同固定动作生成数量 Z 对系统性能的影响, 具体比较了平均归一化加权总计算速率和平均归一化计算时延的变化趋势. 随着动作生成数量从 1 增加到 4, 系统的平均归一化加权总计算速率显著提升. 当动作生成数量达到 5 时, 平均归一化加权总计算速率接近 1, 表明此时已接近最优计算效率. 进一步增加动作生成数量虽然保持

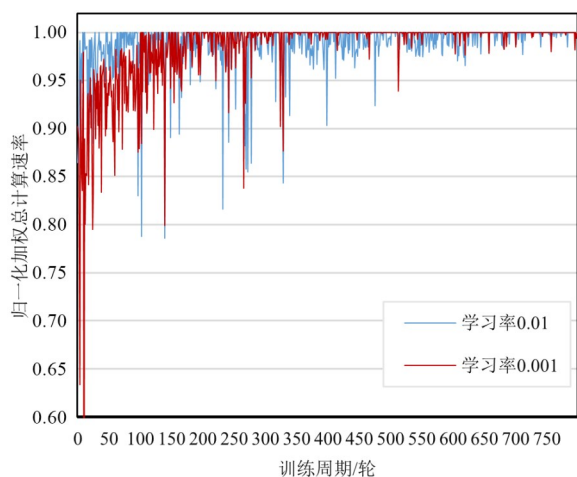


图9 不同学习率下归一化加权总计算速率变化趋势

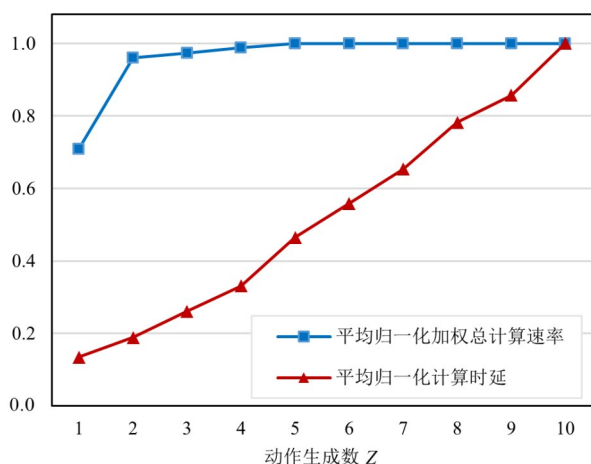


图10 不同固定动作生成数的平均归一化加权总计算速率和平均归一化计算时延

了较高的计算速率,但也导致了系统资源的额外消耗和计算时延的增加. 这表明,在车辆数量为10的情况下,固定动作生成数量 Z 设置为5已经能够实现较好的性能. 因此,为了进一步优化资源利用并降低计算时延,采用动态调整动作生成数量的方法是必要的.

图11展示了在不同车辆数量下,动态调整与固定动作生成数 Z 的单位时间片平均执行时延的比较. 初始情况下, Z 设置为等于车辆数 M . 结果表明,在车辆数量为5、10、20和30时,动态调整 Z 的平均执行时延始终低于固定动作生成数 Z 的平均执行时延. 随着车辆数量 M 的增加,固定动作生成数 Z 的平均时延呈现出指数级增长,而动态调整策略则有效地抑制了这一增长趋势. 这表明动态调整方案能够显著减少时延,因此在应对车辆高速移动性和车联网环境的动态变化方面展现出显著优势.

下面的实验在平均归一化加权总计算速率和第20000个时间片下得到的加权总计算速率上将DBOA-AG、DPOM、DBOA-KNN、LC和OC进行了对比,并对比了DBOA-AG、DPOM、DBOA-KNN得到的平均卸载率,

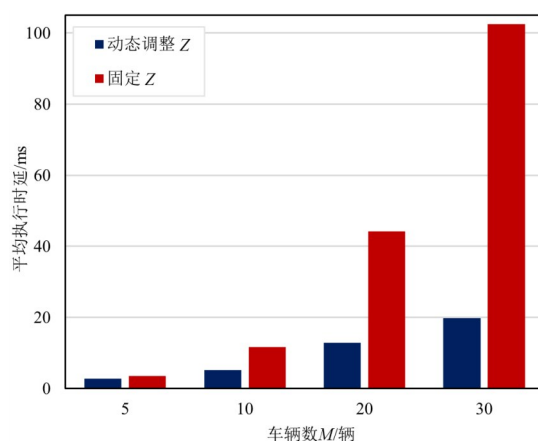


图11 在不同车辆数下动态调整和固定动作生成数 Z 的平均执行时延

来证明DBOA框架的性能优越性. 图12比较了DBOA-AG、DPOM、DBOA-KNN、LC和OC五种算法在不同车辆数 M 下的平均归一化加权总计算速率. 除了OC算法得到的平均归一化加权总计算速率是随着车辆数的增加持续减小外,其余四种任务卸载算法都是在 M 为10时的平均归一化加权总计算速率最大. 随后,随着 M 增加到20、30时,平均归一化加权总计算速率随之减小. 此外,在相同数量的车辆数下,DPOM和DBOA-KNN获得的平均 G^* 略低于DBOA-AG. 但是DBOA框架下得到的平均 G^* 都远大于其他两种卸载算法.

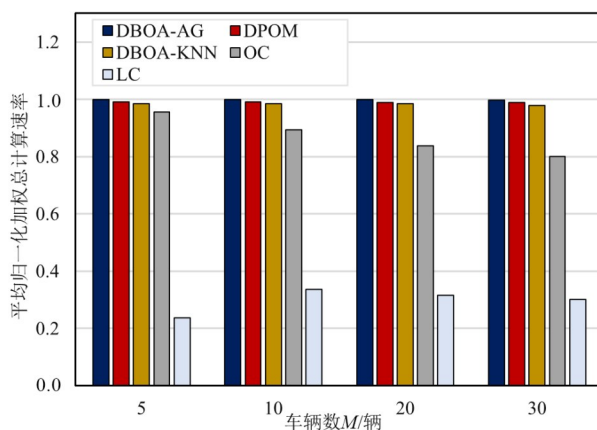


图12 在不同车辆数下的平均归一化加权总计算速率

图13对比了在第20000个时间片下五种任务卸载方法在不同车辆数下的加权总计算速率. 随着车辆数量的增加,五种算法的加权总计算速率都随之增加. 在车辆数相同情况下,DBOA算法框架下的卸载策略得到的加权总计算速率都大于OC和LC,并且DBOA-AG得到的加权总计算速率要大于DPOM和DBOA-KNN. DBOA-KNN采用KNN算法生成的卸载策略彼此更接近,容易陷入局部最优解;DPOM生成的卸载策略过于分散;而DBOA-AG生成的卸载策略较为分散且区别较大. 此外,本地计算速率和卸载计算速率所占的权重分

别是 1.0 和 1.5,所以在数据上 LC 获得的平均 G^* 和加权总计算速率要远远低于其他三种卸载算法. 而 OC 算法没有充分利用本地的计算资源,缺乏合适的边缘资源和本地资源调度机制,在性能上要劣于基于 DBOA 框架的任务卸载算法.

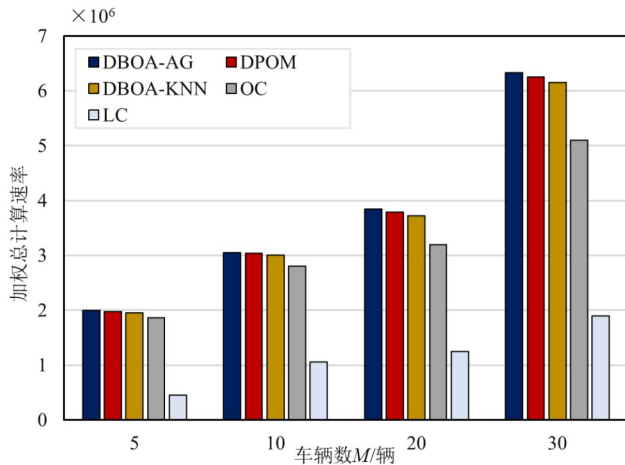


图 13 第 20 000 个时间片下不同车辆数的加权总计算速率

图 14 描述了 DBOA-AG、DPOM 和 DBOA-KNN 在不同车辆数下获得的平均任务卸载率. 在相同的车辆数下, DBOA-AG 获得的平均任务卸载率都要高于 DPOM 和 DBOA-KNN, 这表明 DBOA-AG 在任务卸载决策中更倾向于选择较优的卸载方案, 以最大化整体的计算速率. 此外, 随着车辆数的增加, 车辆之间对有限的边缘资源的竞争增大, 三种算法的平均任务卸载率都逐步减少.

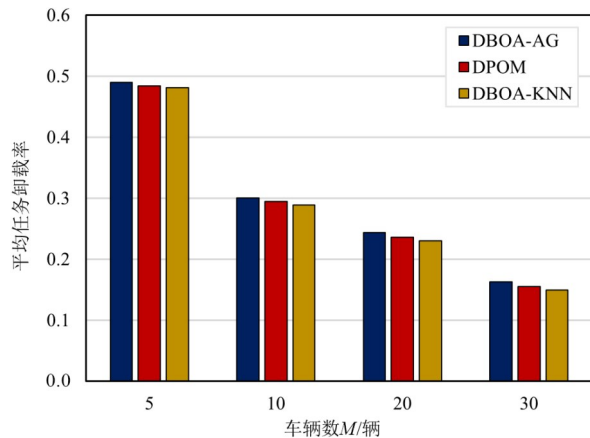


图 14 DBOA-AG、DPOM 和 DBOA-KNN 在不同车辆数下的平均任务卸载率

6 结论

本文首先设计一种车联网边缘计算模型, 然后针对由车联网环境动态多变导致边缘服务器过载的问题, 提出了切比雪夫图加权网络 ChebWN 来提前预测未来各路段的交通流量, 并根据 ChebWN 的预测结果提前

分配边缘服务器资源. 最后, 以在有限时间限制内最大化车联网系统的加权总计算速率为优化目标, 提出任务卸载 DBOA 方案. 大量的对比实验验证了 ChebWN 在预测交通流量方面的准确性, 以及 DBOA 在提升车联网服务响应速度方面的优越性.

然而, 能耗约束并不在本文的研究范围内, 未来的工作将会把能耗、信道状态等更多约束条件作为研究重点, 设计出更加适应现实场景的任务卸载算法. 在实际交通环境中, 常常会有隧道等车载设备接收不到网络信号的地形, 车辆必须选择本地计算. 此外, 解决分布式移动场景下的任务卸载问题是另一个研究目标.

参考文献

- [1] HAN J, PARK H, PARK K, et al. Field demonstration of advanced autonomous navigation technique for a fully unmanned surface vehicle in complex coastal traffic areas[J]. *Journal of Field Robotics*, 2023, 40(8): 1887-1905.
 - [2] MIQDADY T, DE OÑA R, CASAS J, et al. Studying traffic safety during the transition period between manual driving and autonomous driving: A simulation-based approach[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(6): 6690-6710.
 - [3] XIANG H L, ZHANG X Y, HU H S, et al. OptIForest: Optimal isolation forest for anomaly detection[C]//*Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, 2023: 2379-2387.
 - [4] JIANG K, SUN C, ZHOU H, et al. Intelligence-empowered mobile edge computing: Framework, issues, implementation, and outlook[J]. *IEEE Network*, 2021, 35(5): 74-82.
 - [5] LIN D, WU W W. Resource allocation in a secure Internet of battle vehicles through RF fingerprint recognition[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(5): 6879-6884.
 - [6] DAI C, LIU X G, CHEN W T, et al. A low-latency object detection algorithm for the edge devices of IoV systems[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(10): 11169-11178.
 - [7] ARMBRUST M, FOX A, GRIFFITH R, et al. A view of cloud computing[J]. *Communications of the ACM*, 2010, 53(4): 50-58.
 - [8] 李松, 王新荣, 王博文, 等. 基于随机网络演算的车联网边缘计算多跳任务卸载性能分析[J]. *电子与信息学报*, 2023, 45(7): 2459-2466.
- LI S, WANG X R, WANG B W, et al. Performance analysis of multi-hop task offloading of vehicular edge computing based on stochastic network calculus[J]. *Journal of Electronics & In-*

- formation Technology, 2023, 45(7): 2459-2466. (in Chinese)
- [9] REN J K, YU G D, HE Y H, et al. Collaborative cloud and edge computing for latency minimization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 5031-5044.
- [10] WANG F, XU J, WANG X, et al. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[J]. IEEE Transactions on Wireless Communications, 2018, 17(3): 1784-1797.
- [11] BUTE M S, FAN P Z, ZHANG L, et al. An efficient distributed task offloading scheme for vehicular edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2021, 70(12): 13149-13161.
- [12] GOUDARZI M, WU H M, PALANISWAMI M, et al. An application placement technique for concurrent IoT applications in edge and fog computing environments[J]. IEEE Transactions on Mobile Computing, 2021, 20(4): 1298-1311.
- [13] SHI W S, CAO J, ZHANG Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [14] SHAHZAD H, SZYMANSKI T H. A dynamic programming offloading algorithm for mobile cloud computing[C]//2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). Piscataway: IEEE, 2016: 1-5.
- [15] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.
- [16] WANG F, XU J, DING Z G. Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems[J]. IEEE Transactions on Communications, 2019, 67(3): 2450-2463.
- [17] YAO L, XU X L, BILAL M, et al. Dynamic edge computation offloading for internet of vehicles with deep reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(11): 12991-12999.
- [18] XU X L, WU F, BILAL M, et al. XRL-SHAP-cache: An explainable reinforcement learning approach for intelligent edge service caching in content delivery networks[J]. Science China Information Sciences, 2024, 67(7): 170303.
- [19] ZHANG J W, CHEN S H, WANG X D, et al. Dynamic reservation of edge servers *via* deep reinforcement learning for connected vehicles[J]. IEEE Transactions on Mobile Computing, 2023, 22(5): 2661-2674.
- [20] XU X L, YANG C Y, BILAL M, et al. Computation offloading for energy and delay trade-offs with traffic flow prediction in edge computing-enabled IoV[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(12): 15613-15623.
- [21] JIANG H B, DAI X X, XIAO Z, et al. Joint task offloading and resource allocation for energy-constrained mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(7): 4000-4015.
- [22] NING Z L, DONG P R, WANG X J, et al. Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks[J]. IEEE Transactions on Mobile Computing, 2022, 21(4): 1319-1333.
- [23] WAN S H, GU R H, UMER T, et al. Toward offloading internet of vehicles applications in 5G networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(7): 4151-4159.
- [24] DAI P L, SONG F, LIU K, et al. Edge intelligence for adaptive multimedia streaming in heterogeneous internet of vehicles[J]. IEEE Transactions on Mobile Computing, 2023, 22(3): 1464-1478.
- [25] LUO G Y, ZHOU H B, CHENG N, et al. Software-defined cooperative data sharing in edge computing assisted 5G-VANET[J]. IEEE Transactions on Mobile Computing, 2021, 20(3): 1212-1229.
- [26] XIAO K L, SHI W S, GAO Z P, et al. DAER: A resource preallocation algorithm of edge computing server by using blockchain in intelligent driving[J]. IEEE Internet of Things Journal, 2020, 7(10): 9291-9302.
- [27] WU Q, WANG W H, FAN P Y, et al. URLLC-awared resource allocation for heterogeneous vehicular edge computing[J]. IEEE Transactions on Vehicular Technology, 2024, 73(8): 11789-11805.
- [28] TU Y, LIN S K, QIAO J Z, et al. Deep traffic congestion prediction model based on road segment grouping[J]. Applied Intelligence, 2021, 51(11): 8519-8541.
- [29] WU F, SOUZA A, ZHANG T, et al. Simplifying graph convolutional networks[C]//Proceedings of the 36th International Conference on Machine Learning. Cham: Springer, 2019: 6861-6871.
- [30] FANG M Y, TANG L L, YANG X, et al. FTPG: A fine-grained traffic prediction method with graph attention network using big trace data[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 5163-5175.
- [31] BOZORGCHENANI A, MAGHSUDI S, TARCHI D, et al. Computation offloading in heterogeneous vehicular edge networks: On-line and off-policy bandit solutions[J]. IEEE Transactions on Mobile Computing, 2022, 21(12): 4233-4248.
- [32] ZHANG R, WU L B, CAO S Q, et al. A vehicular task offloading method with eliminating redundant tasks in 5G HetNets[J]. IEEE Transactions on Network and Service

Management, 2023, 20(1): 456-470.

- [33] SHEN Q Q, HU B J, XIA E J. Dependency-aware task offloading and service caching in vehicular edge computing[J]. IEEE Transactions on Vehicular Technology, 2022, 71(12): 13182-13197.
- [34] HUI Y L, SU Z, LUAN T H, et al. A game theoretic scheme for collaborative vehicular task offloading in 5G HetNets[J]. IEEE Transactions on Vehicular Technology, 2020, 69(12): 16044-16056.
- [35] NING Z L, DONG P R, WANG X J, 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.
- [36] SHI J M, DU J, WANG J J, et al. Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning[J]. IEEE Transactions on Vehicular

Technology, 2020, 69(12): 16067-16081.

- [37] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2009, 20(1): 61-80.
- [38] SAHBI H. Learning laplacians in Chebyshev graph convolutional networks[C]//2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). Piscataway: IEEE, 2021: 2064-2075.
- [39] ZHANG S B, GU H, CHI K K, et al. DRL-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network[J]. IEEE Transactions on Wireless Communications, 2022, 21(12): 10934-10948.
- [40] HUANG L, BI S Z, ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. IEEE Transactions on Mobile Computing, 2020, 19(11): 2581-2593.

作者简介



许小龙 男,1988年8月出生于江苏省海安市。现为南京信息工程大学教授、博士生导师。主要研究方向为边缘智能、推荐系统、服务计算、大数据等。
E-mail: njuxlxu@gmail.com



齐连永 男,1982年12月出生于山东省寿光市。现为中国石油大学(华东)教授、博士生导师。主要研究方向为推荐系统与隐私计算。
E-mail: lianyongqi@qfnu.edu.cn



杨威 男,2000年10月出生于江苏省盐城市。现为南京信息工程大学硕士研究生。主要研究方向为车联网与边缘计算。
E-mail: xsyangwei@126.com



项昊龙 男,1993年6月出生于浙江省金华市。现为南京信息工程大学讲师。主要研究方向为异常检测与数据挖掘。
E-mail: hlxiang@nuist.edu.cn



杨辰翊 男,2000年11月出生于江苏省常州市。现为东南大学硕士研究生。主要研究方向为深度学习和边缘计算。
E-mail: 220236179@seu.edu.cn



窦万春 男,1971年8月出生于江苏省徐州市。现为南京大学教授、博士生导师。主要研究方向为大数据与边缘计算。
E-mail: douwc@nju.edu.cn



程勇 男,1980年9月出生于四川省江津市。现为南京信息工程大学正高级工程师。主要研究方向为人工智能应用与软件工程应用。
E-mail: yongcheng@nuist.edu.cn