

车载边缘计算中任务卸载和服务缓存的联合智能优化

刘雷¹, 陈晨¹, 冯杰¹, 裴庆祺¹, 何辞², 窦志斌²

(1. 西安电子科技大学综合业务网理论及关键技术国家重点实验室, 陕西 西安 710071;

2. 中国电子科技集团公司第 54 研究所, 河北 石家庄 050081)

摘 要: 针对车载环境下有限的网络资源和大量用户需求之间的矛盾, 提出了智能驱动的车载边缘计算网络架构, 以实现网络资源的全面协同和智能管理。基于该架构, 设计了任务卸载和服务缓存的联合优化机制, 对用户任务卸载以及计算和缓存资源的调度进行了建模。鉴于车载网络的动态、随机和时变的特性, 利用异步分布式强化学习算法, 给出了最优的卸载决策和资源管理方案。实验结果表明, 与其他算法相比, 所提算法取得了明显的性能提升。

关键词: 车联网; 边缘计算; 计算卸载; 服务缓存; 智能优化

中图分类号: TN92

文献标识码: A

DOI: 10.11959/j.issn.1000-436x.2021017

Joint intelligent optimization of task offloading and service caching for vehicular edge computing

LIU Lei¹, CHEN Chen¹, FENG Jie¹, PEI Qingqi¹, HE Ci², DOU Zhibin²

1. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

2. The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China

Abstract: Given the contradiction between limited network resources and massive user demands in Internet of vehicles, an intelligent vehicular edge computing network architecture was proposed to achieve the comprehensive cooperation and intelligent management of network resources. Based on this architecture, a joint optimization scheme of task offloading and service caching was furtherly devised, which formulated an optimization problem about how to offload tasks and allocate computation and cache resources. In view of the dynamics, randomness and time variation of vehicular networks, an asynchronous distributed reinforcement learning algorithm was employed to obtain the optimal task offloading and resource management policy. Simulation results demonstrate that the proposed algorithm achieves significant performance improvement in comparison with the other schemes.

Keywords: Internet of vehicles, edge computing, computation offloading, service caching, intelligent optimization

1 引言

作为交通强国的重要抓手, 车联网在国家发展

战略中起着举足轻重的作用^[1-3]。随着车联网的飞速发展, 车辆变得愈发普及和智能化。由此, 催生了一大批车载应用, 涵盖信息服务、行驶安全和交通

收稿日期: 2020-09-16; 修回日期: 2020-12-03

通信作者: 陈晨, cc2000@mail.xidian.edu.cn

基金项目: 国家重点研发计划基金资助项目 (No.2020YFB1807500); 国家自然科学基金资助项目 (No.62072360, No.61571338, No.61672131, No.62001357, No.61901367); 陕西省重点研发计划基金资助项目 (No.2019ZDLGY13-07, No.2019ZDLGY13-04, No.2020JQ-844); 嵌入式系统与服务计算教育部重点实验室开放基金资助项目 (同济大学) (No.ESSCKF2019-05); 西安市科技计划基金资助项目 (No.20RGZN0005); 西安市重点实验室建设基金资助项目 (No.201805052-ZD3CG36)

Foundation Items: The National Key Research and Development Program of China (No.2020YFB1807500), The National Natural Science Foundation of China (No.62072360, No.61571338, No.61672131, No.62001357, No.61901367), The Key Research and Development Plan of Shaanxi Province (No.2019ZDLGY13-07, No.2019ZDLGY13-04, No.2020JQ-844), The Key Laboratory of Embedded System and Service Computing (Tongji University) (No.ESSCKF2019-05), Ministry of Education, Xi'an Science and Technology Plan (No.20RGZN0005), The Xi'an Key Laboratory of Mobile Edge Computing and Security (No.201805052-ZD3CG36)

效率各个方面^[4-6]。这些应用服务在给人们生活带来便利的同时，将会造成数据的几何增长，增加了网络的负荷，对网络带宽提出了更高的要求。车载边缘计算通过把移动边缘计算应用在车联网，可以实现计算和存储能力的下沉，能够极大缓解网络的带宽压力，有效降低任务的响应时延^[7-8]。

在复杂的车载网络环境下，为了保障大量用户多样化的服务需求，亟须设计有效的车载边缘计算机制^[9]。利用计算卸载技术，用户可以把任务卸载给具有丰富资源的边缘节点计算，有助于响应时延的减少。然而，现有的车载计算卸载工作，在用户端往往集中在本地处理，未能充分发掘邻居车辆的资源，而在边缘端大多侧重于计算资源的管理，忽视了其与服务缓存之间的关系。特别地，边缘端服务器为了计算用户卸载的任务，需要具备一定的计算资源，也需要提前缓存相应的服务应用。换言之，计算卸载和服务缓存彼此关联，相互耦合。考虑到路边设施存储资源的限制，如何通过服务缓存的决策保障计算卸载的质量是要解决的重要问题。鉴于车联网的动态、随机和时变特性，需要引入更加智能的算法实现网络通信、计算和缓存资源的有效管理，以应对传统数学方法的不足^[10]。

针对以上问题，本文首先设计了纵向和横向协同的智能车载边缘计算网络架构，然后通过分析网络通信、计算和服务缓存资源之间相互作用的机理，提出了通信、计算和服务缓存资源的联合优化模型，进而利用异步分布式强化学习实现了任务的灵活卸载和资源的智能管理。

2 相关工作

区别于一般的移动网络^[11]，车联网的典型特点在于车辆的快速移动。车辆的移动会导致网络拓扑的动态变化，决定车间的连通特性，从而影响任务的正常卸载。为此，车载边缘计算需要和车辆的移动性密切结合。文献[12]考虑网络负荷和任务卸载，研究了多服务器多用户场景下的资源管理。每辆车通过移动可以将任务选择性地卸载给期望的边缘服务器。文献[13]呈现了一个移动模型用于设计链路稳定性指标。基于该指标可以发现任务车辆周边可用的服务车辆，从中可以挑选满足任务车辆偏好和服务需求的车辆作为最优的服务提供者。不同于传统计算卸载工作主要考虑通信和计算资源的调

度，文献[14]设计的基于车辆移动的卸载机制同时也考虑了任务卸载时间的决策。特别地，任务车辆与服务器之间的数据传输速率随两者之间的距离动态变化，由此影响了任务的卸载时间。

在车载环境下，路边单元广泛部署于路测，通常作为主要的边缘服务器节点参与用户任务的处理。文献[15]考虑车辆的移动及其与关联的边缘服务器的连接时间，研究了负载卸载和任务调度问题。文献[16]提出的双端优化问题旨在同时保障用户端和服务器端的利益。以上工作主要侧重于单服务器场景，文献[17-18]则聚焦于多服务场景。文献[17]提出了具有高可靠性、低时延的车-设施通信架构，优化了车和基站的耦合及无线资源的管理。文献[18]的任务卸载机制则同时优化了服务器和传输模式的选择。

鉴于车联网的复杂特性，人工智能算法以其巨大的优势也被用于车载边缘计算，以实现资源的智能管理。文献[19]利用Q-学习算法实现闲置车辆资源和服务器资源的管理，以加强用户的服务质量。文献[20-21]均通过深度Q-学习联合优化了网络的通信、计算和缓存资源，旨在提升系统的整体收益。文献[22]则利用深度确定性策略梯度算法实现任务的调度和资源的管理，最大程度保障移动运营商的收益。

以上工作主要集中在车载计算卸载方面，忽视了车辆资源的发掘和服务缓存对计算卸载的影响。相比于文献[12-18]，文献[19-22]虽然采用智能方法实现任务的调度，但依然存在一定的局限性。为此，本文提出了计算卸载和服务缓存智能联合优化算法。

3 系统模型

本文构建了一个边缘智能驱动的车载网络架构，如图1所示。该架构包括三层，即用户层、边缘层和云层，特点介绍如下。

纵向协作。用户层位于网络的最底端，主要由车辆组成。部署于道路一侧的路边单元配置相应的边缘服务器，作为边缘层的关键节点。特别地，在边缘层引入智能模块，协助实现资源的有效管理和任务的灵活决策。云层位于网络的最上端，具有丰富的计算和存储资源。在用户和边缘服务器资源受限的情况下，云层可提供必要的资源支持。

横向协作。当车辆有任务处理时，可以选择本

地执行并通过邻居车辆计算任务，还可利用车-设施通信方式交由路边单元协助处理。路边单元的资源往往在空时维度分布不均：轻负载的服务器资源会呈现闲置状态造成浪费，过负载的服务器则对应接不暇的任务捉襟见肘。为此路边单元之间可以加强横向协作，通过任务迁移的策略，最大化网络资源的利用率。

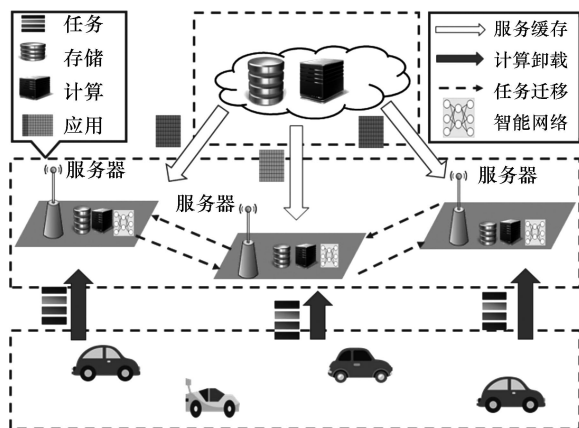


图 1 车载边缘计算网络架构

移动感知。由于高速的移动性，车辆可能频繁地在不同的路边单元之间切换。所以，需要能够基于对车辆移动行为的分析对车辆的轨迹准确定位，以便路边单元将计算结果顺利反馈给车辆。

假设 M 个路边单元均匀分布于道路一侧，组成集合 \mathcal{M} 。每个路边单元配备一个计算能力为 F_j 、存储资源为 S_j 的服务器。 N 个车辆自由移动在道路上，组成集合 \mathcal{N} 。每个车辆 i 携带一个任务，该任务可以表征为 $\{d_i, c_i\}$ ，其中， d_i 表示输入数据的大小， c_i 表示该任务的计算量。路边单元通过有线方式互联。用户与路边通过无线通信方式进行交互。车辆本地的卸载决策用 x_{i0} 表示，其中， $x_{i0} = 1$ 表示车辆在用户侧处理任务；车辆边缘的卸载决策用 x_{ij} 表示，其中， $x_{ij} = 1$ 表示车辆将任务卸载给路边单元 j 处理。特别地，当车辆执行边缘卸载处理时，优先邻近关联的路边单元。如果当前关联的路边单元负荷较重，则可以由该服务器将任务迁移至周边的路边单元。这样有利于负载均衡，提升资源的利用率，从而加强用户的服务体验。对于每个路边单元，为了实现任务的处理，需要安装相应的服务应用。换言之，当其存储了相应的服务应用，即缓存决策 $w_{ij} = 1$ 时，路边单元 j 能够处理车辆 i 卸载的任务；否则它需要从云端下载该应用，从而带来了额

外的时延开销。

3.1 移动模型

定义 δ_{ab} 为相邻车辆 a 和 b 的连通时间， R 为车辆的通信范围。令 $v_a(t)$ 和 $v_b(t)$ 分别为两车在 t 时刻的速度， $(x_a(t), y_a(t))$ 和 $(x_b(t), y_b(t))$ 分别为两车在 t 时刻的坐标。那么，两车的连通时间可以表示^[23]为

$$\delta_{ab} = \frac{R - \phi \sqrt{(y_a(t) - y_b(t))^2 + (x_a(t) - x_b(t))^2}}{|v_a(t) - \phi v_b(t)|} \quad (1)$$

其中，当 $\phi = -1$ 、 $\phi = 1$ 时，后车 a 和前车 b 同向行驶，且前车速度小于后车；当 $\phi = 1$ 、 $\phi = -1$ 时，后车 a 和前车 b 同向行驶，且前车速度大于后车；当 $\phi = -1$ 、 $\phi = -1$ 时，车辆 a 和车辆 b 位于不同车道且相向行驶；当 $\phi = 1$ ， $\phi = -1$ 时，车辆 a 和车辆 b 位于相同车道，且反向行驶。

3.2 通信模型

车-车通信模型。车-车通信采用基于分布式协调功能 (DCF, distributed coordination function) 的 IEEE 802.11p 协议。车辆利用 CSMA/CA 机制竞争信道。令 $E[s_n]$ 表示成功传输一个数据所需要的平均时隙数目， $E[s_l]$ 表示每个时隙的平均长度。那么，在车辆 i 和相邻车辆 k 之间成功传输一个数据所需的平均时延^[24]为

$$E[t_{ik}] = E[s_n]E[s_l] \quad (2)$$

车-设施通信模型。任务车辆在执行边缘卸载时，通过车-设施通信方式将任务上传给路边单元，其中通信采用 LTE-V2X 协议。定义 h_i 和 B_i 分别为车辆与路边单元之间的信道增益和信道带宽。令 ρ_i 表示用户的传输功率， σ^2 表示传输的噪声。那么，根据香农定理可得，数据的上传速率为

$$r_i = B_i \log \left(1 + \frac{\rho_i h_i}{\sigma^2} \right) \quad (3)$$

3.3 缓存模型

路边单元执行车辆卸载任务的前提在于其预先安装了所需要的服务应用。考虑到存储空间有限性，路边单元不可能缓存所有需要的服务应用。定义路边单元 j 的存储大小为 C_j ，任务车辆 i 服务应用的大小为 σ_i ，则有式(4)成立。

$$\sum_i w_{ij} \sigma_i < C_j \quad (4)$$

3.4 计算模型

任务车辆可以通过用户层计算和边缘层卸载 2 种

方式处理任务。下面,对两者的时延性能分别进行分析。

3.4.1 用户层计算

为了充分利用车辆资源,任务车辆除了可以在本地处理任务外,还可以借助其通信范围内的邻居车辆实现任务的计算。定义 f_i 为任务车辆 i 自身的计算能力。那么,车辆 i 通过本地计算方式处理自己任务所需要的时间可表示为

$$t_{i0}^e = \frac{c_i}{f_i} \quad (5)$$

定义 \mathcal{N}_i 为车辆 i 通信范围内的车辆集合。当任务车辆利用其邻居车辆 $k \in \mathcal{N}_i$ 计算任务时,时延包括任务在两车之间的传输时延、任务在邻居车辆的计算时延和结果的反馈时延。这里,本文忽略结果的反馈时延。对于传输时延而言,根据式(2)可求得平均传输时延 t_{ik} ; 对于计算时延来说,通过式(5)可以求得平均计算时延 t_{k0}^e 。

综上可得,完成任务车辆任务计算所需的最小时延为

$$t_{i0}^l = \min_{k \in \mathcal{N}_i} \{t_{i0}^e, t_{ik} + t_{k0}^e\} \quad (6)$$

约束条件为

$$t_{ik} + t_{k0}^e < \delta_{ik}$$

其中,约束条件是为了保障选定的邻居车辆能够在两车有效通信时间 δ_{ik} 内完成任务的处理和反馈。

3.4.2 边缘层卸载

当任务车辆执行边缘卸载时,一般包括以下阶段:任务上传、任务执行和结果反馈。本文忽略结果反馈的时延假设任务车辆 i 选择卸载的路边单元为 j , 分别对不同阶段的时延进行分析。

任务上传阶段。车辆 i 首先把任务上传给当前关联的路边单元 s_j , 该过程的传输时延取决于任务的大小和数据的传输速率。由式(3)可得

$$t_{ij}^u = \frac{d_i}{r_i} \quad (7)$$

任务执行阶段。根据所选定卸载服务器的位置,任务执行分为以下2种情况。

情况1 路边单元 s_i 和 j 相同。该情况下,任务在当前路边单元计算。如果路边单元存储了计算该任务所需的服务应用,则可以直接计算任务,所需时延取决于任务的算力需求和路边单元分配的计算资源;否则,还要考虑从云端下载相应服务应

用的额外时延 t_i^{cloud} 。综上,完成任务计算所需要的时间为

$$t_{ij}^e = \frac{c_i}{f_{ij}} + (1 - w_{ij}) t_i^{\text{cloud}} \quad (8)$$

其中, f_{ij} 表示路边单元给车辆分配的算力。

情况2 路边单元 s_i 和 j 不同。该情况下,需要考虑任务在两者之间的迁移时延。路边单元之间通过有线链路连接。假设 s_i 和 j 之间存在 $l_{s_i, j}$ 个链路,而每个链路的平均传输时延为 $t_{\text{one-link}}$, 那么,任务在2个路边单元之间的迁移时间为 $t_{s_i, j}^m = t_{\text{one-link}} l_{s_i, j}$ 。结合式(8),可得完成任务处理所需要的时间为

$$t_{ij}^e = \frac{c_i}{f_{ij}} + (1 - w_{ij}) t_i^{\text{cloud}} + t_{\text{one-link}} l_{s_i, j} \quad (9)$$

结果反馈阶段。一旦选定的路边单元完成任务的处理,就需要将结果反馈给车辆。由于移动性,需要考虑车辆此时是否可能驶出了起初关联的路边单元。因此,可将任务在上传和执行阶段的时间 T_{ij} 与车辆在起初关联服务器传输范围内的时间 ξ_{is_i} 做比较。其中, $T_{ij} = t_{ij}^u + t_{ij}^e$, ξ_{is_i} 取决于用户驶出服务器通信范围的时间和移动速度的比值。如果 $T_{ij} < \xi_{is_i}$, 可将结果首先传输给该路边单元,然后反馈给车辆;否则,需要对车辆的移动定位,判断当前位于哪个路边单元,以便将结果传输给该服务器,进而反馈给车辆。

3.5 计算卸载和服务缓存联合优化模型

本文旨在动态、随机和时变的车载环境下,面对有限网络资源和不同用户需求之间的矛盾,通过计算卸载和服务缓存资源联合优化,在保障用户服务需求的前提下,最小化系统整体的处理时延。鉴于此,设计目标函数如下

$$\min_{x, f, w} \sum_{i \in \mathcal{N}} \left(x_{i0} t_{i0}^l + \sum_{j \in \mathcal{M}} x_{ij} T_{ij} \right) \quad (10)$$

$$C1: x_{ij} \subseteq \{0, 1\}, \forall i \in \mathcal{N}, j \in \{0, \mathcal{M}\} \quad (11)$$

$$C2: \sum_{j \in \{0, \mathcal{M}\}} x_{ij} = 1, \forall i \in \mathcal{N} \quad (12)$$

$$C3: \sum_i x_{ij} f_{ij} < F_j, \forall j \in \{0, \mathcal{M}\} \quad (13)$$

$$C4: \sum_i w_{ij} \varpi_i < S_j, \forall j \in \{0, \mathcal{M}\} \quad (14)$$

$$C5: T_{ij} \leq \delta_i, \forall i \in \mathcal{N} \quad (15)$$

其中, $x = \{x_{ij}\}$, $w = \{w_{ij}\}$, $f = \{f_{ij}\}$ 。根据式(6)~式(9), 可分别得到 t_{i0}^l 和 T_{ij} 。这里, 假设车辆分配的带宽资源一样。限制性条件 C1 表示每个任务有用户层处理和边缘层卸载 2 种处理方式; C2 表示每个任务仅在一个地方执行; C3 表示服务器的计算资源限制; C4 表示服务器的缓存资源限制, 其中, ω_i 表示执行任务所需要的服务应用的大小; C5 表示车辆卸载给路边单位的任务应该在其离开关联的服务器传输范围之前完成, 其中, δ_i 表示用户和其关联的服务器的连接时间, 取决于用户驶出服务器通信范围的时间和移动速度的比值。

4 基于异步分布式强化学习的问题求解

鉴于车载网络的动态性、随机性和时变性, 人工智能算法相比于传统数学方法更适合资源的管理和任务的调度。相比较而言, Q-学习需要维护 Q 表格, 不适应于具有较多状态的网络。深度确定性策略梯度算法需要利用经验回放机制消除训练数据间的相关性。对于经验回放机制来说, 代理在与环境的每次交互都需耗费较多的资源, 而所采用的离策略学习方法只能基于旧策略生成的数据进行更新。所以, 考虑利用异步优势的 actor-critic 算法减少算法执行所需的开销, 同时基于实时的网络环境提供最优的卸载决策和资源管理。

利用异步优势的 actor-critic 算法对系统环境建模, 需要确定其状态空间、动作空间和奖励函数, 具体如下。

状态空间。状态空间 S 由车载网络的计算资源和缓存资源组成, $S = \{F_1, F_2, \dots, F_M, S_1, S_2, \dots, S_M\}$ 。其中, F_i 和 S_i 分别表示路边单元 i 的计算能力和存储能力。

动作空间。动作空间由车辆的卸载决策、路边单元的缓存和计算资源管理组成, $A = (x_i, w_i, f_i)$ 。其中, x_i 、 w_i 和 f_i 分别代表车辆 i 的卸载决策、路边单元存储和计算资源管理的集合, $x_i = \{x_{i0}, x_{i1}, \dots, x_{iM}\}$, $w_i = \{w_{i1}, w_{i2}, \dots, w_{iM}\}$, $f_i = \{f_{i1}, f_{i2}, \dots, f_{iM}\}$ 。

奖励函数。根据本文的目标函数, 即时回报设计为 $-\sum_{i \in \mathcal{N}} \left(x_{i0} t_{i0}^l + \sum_{j \in \mathcal{M}} x_{ij} T_{ij} \right)$ 。

异步优势的 actor-critic 算法中的公共神经网络包括多个线程, 每个线程具有和公共神经网络一样的 2 个模块: 策略 (actor) 网络和评价 (critic) 网

络。actor 网络用于优化参数为 θ 策略 $\pi(a_t | s_t; \theta)$; critic 网络尝试估计参数为 θ_v 的价值函数 $V(s_t; \theta)$ 。在时刻 t , actor 网络基于当前状态 s_t 执行动作 a_t , 得到奖赏 r_t 并进入下一个状态 s_{t+1} 。

利用优势函数 $A(a_t, s_t)$ 表示动作价值函数 $Q(a_t, s_t)$ 和状态价值函数 $V(s_t)$ 的差值, 如式(16)所示。

$$A(a_t, s_t) = Q(a_t, s_t) - V(s_t) \quad (16)$$

为了加速收敛, 采用 k 步采样近似估计 $Q(a_t, s_t)$, 如式(17)所示。

$$Q(a_t, s_t) \approx \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}; \theta_v) \quad (17)$$

其中, $\gamma \in (0, 1]$ 为折扣系数, r_{t+i} 表示即时奖赏, $V(s_t)$ 则通过 critic 网络得到。

对于策略损失函数, 有

$$f_\pi(\theta) = \log \pi(a_t | s_t; \theta) A(a_t, s_t) + \beta H(\pi(s_t; \theta)) \quad (18)$$

以参数 θ 为变量, 对策略损失函数微分可得

$$\nabla_\theta f_\pi(\theta) = \nabla_\theta \log \pi(a_t | s_t; \theta) A(a_t, s_t) + \beta \nabla_\theta H(\pi(s_t; \theta)) \quad (19)$$

对于价值函数损失函数, 有

$$f_v(\theta_v) = (R_t - V(s_t; \theta_v))^2 \quad (20)$$

其中, H 为策略的熵, β 为系数。

以参数 θ_v 为变量, 对价值函数损失函数微分可得

$$df_v(\theta_v) = 2(A(a_t, s_t) - V(s_t; \theta_v)) \nabla_{\theta_v} V(s_t; \theta_v) \quad (21)$$

基于 RMSProp 算法, 梯度估计可以表示为

$$g = ag + (1-a)\Delta\theta^2 \quad (22)$$

其中, a 表示动量, $\Delta\theta$ 表示损失函数的累计梯度。

RMSProp 算法可以通过式(23)进行梯度下降的更新。

$$\theta \leftarrow \theta - \eta \frac{\Delta\theta}{\sqrt{g + \varepsilon}} \quad (23)$$

其中, η 表示学习速率, ε 表示一个正数。

单个线程独立地与环境交互并获取经验, 彼此之间互不干扰。经过一定的交互之后, 每个线程独立地使用累计的梯度更新公共神经网络模型参数, 如图 2 所示。进而, 公共神经网络会分发自己的参数更新每个线程的神经网络参数, 指导线程与环境

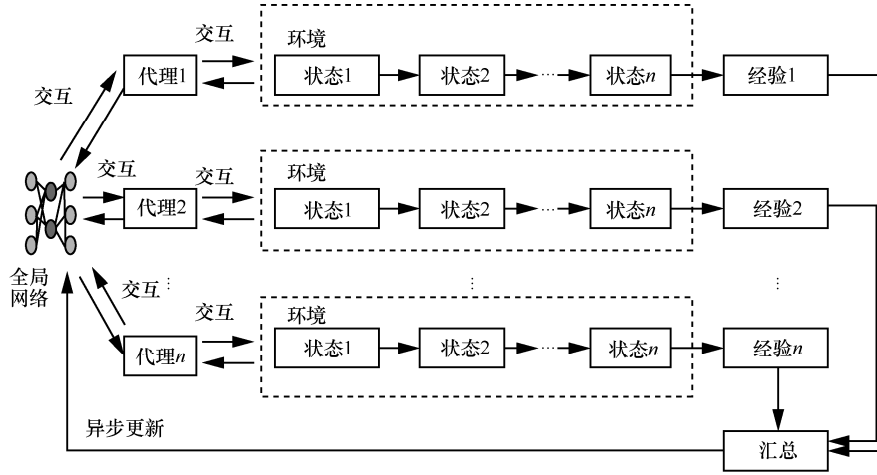


图 2 本文算法网络模型

的交互。本文算法详细描述如下。

算法 1 基于异步分布式强化学习的计算卸载和服务缓存联合优化机制

输入 车辆的属性需求和

输出 车辆的卸载决策，路边单元计算和缓存资源管理决策

初始化 定义 θ 和 Q_v 为全局网络中 actor 网络和 critic 网络的参数；定义 θ^i 和 Q_v^i 为局部网络中 actor 网络和 critic 网络的参数；设置全局计数器 $T=0$ ，设置局部步进计数器 $t=1$ ，设置 T_{\max} 、 t_g 、 γ 、 ε 、 t_{\max} 、学习的速率 η 和代理的数目 W

迭代：

while $T < T_{\max}$ do

if $w=1$ to W do

重置全局梯度 $d\theta=0$ 和 $dQ_v=0$ ；同步局部参数 $\theta^i=\theta$ 和 $Q_v^i=Q_v$ ；设置 $t_0=t$ ，得到系统状态

repeat

根据策略函数获得动作，执行动作，观察奖励以及下一个状态

$t=t+1$

until $t-t_0==t_{\max}$

end if

if $t \% t_g == 0$ then

$R=V(S(t); Q_v^i)$

end if

for $i=t-1$ to t_0 do

$R=R(t)+\gamma R$

根据式(19)计算策略函数梯度 $\nabla_{\theta^i} f_{\pi}(\theta^i)$

计算累计梯度 $d\theta=d\theta+\nabla_{\theta^i} f_{\pi}(\theta^i)$

根据式(21)计算价值函数梯度 $\nabla_{Q_v^i} f_v(Q_v^i)$

计算累计梯度 $dQ_v=dQ_v+\nabla_{Q_v^i} f_v(Q_v^i)$

end for

根据式(23)异步更新权值参数 θ 和 Q_v

end while

5 仿真分析

本节利用 Python 对车载边缘计算卸载算法进行仿真验证，通过比较各算法随车辆数目、路边单元计算能力和存储能力的变化在时延和奖赏方面展现的性能，来评估不同算法的优劣。其中，实现的算法除了本文算法之外，还包括基于随机卸载策略 random processing 和完全卸载策略的 offloading processing。在车载环境下，设置一个云中心和 3 个路边单元。仿真参数如表 1 所示。车辆的计算能力分布于 [100, 500] Mcycle/s，边缘服务器计算能力分布于 [2, 6] Gcycle/s，边缘服务器缓存能力分布于 [200, 1 000] MB，车辆计算能力分布于 [100, 500] Mcycle/s，每个任务的计算强度为 297.62 cycle/bit。

表 1 仿真参数

参数	分布
边缘服务器计算能力/(Gcycle·s ⁻¹)	[2,6]
边缘服务器缓存能力/MB	[200,1 000]
车辆数目/辆	25、35、45
车辆计算能力/(Mcycle·s ⁻¹)	[100,500]
任务计算强度/(cycle·bit ⁻¹)	297.62

图 3 显示了车辆数目对不同算法时延的影响。此时，设置每个路边单元的计算能力为 2 GHz，存

储大小为 300 MB。从图 3 中可以发现，系统任务处理的时延随着车辆数目的增多而增加。这一方面是因为处理任务的增多，另外一方面是因为有限计算资源的竞争。在所有的算法中，random processing 的时延最大。相对于 offloading processing 和本文算法，当采用 random processing 时，车辆会承担较多任务的计算。由于车辆自身计算资源的限制，单独处理任务会造成较大的时延。offloading processing 取得了比 random processing 更好的性能。这主要归因于边缘服务器具有丰富的计算资源。边缘服务器参与任务的计算，会加快任务的处理，降低任务的时延。本文算法相对于以上 2 种算法，完成任务处理所需的时延最小，这是因为本文算法考虑了纵向的端、边和云的协作。为此，所有可用的资源均可以通过协同用于处理任务，提升了资源的利用效率，促进了时延的减少。特别地，在端侧，任务的处理不仅考虑了本地资源，也充分发掘了任务车辆一跳的邻居车辆资源。本文算法的目标在于最小化任务的时延，而所在用的深度强化学习策略能够适应车载网络的动态、随机和时变特性获取相应的最优解。

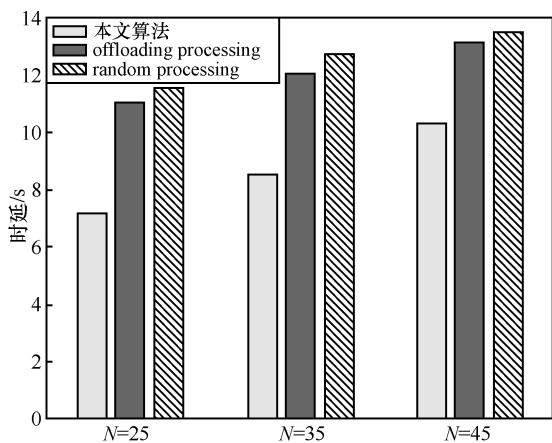


图 3 车辆数目对不同算法时延的影响

图 4 显示了边缘服务器的计算能力对不同算法时延的影响。随着边缘服务器的计算能力的增加，不同算法处理任务的时延随之减少。这是因为任务的计算与边缘服务器的资源呈正相关的关系。对于 random processing 而言，任务可以在端侧处理，也可以由边缘服务器计算。由于未能充分发掘边缘服务器的计算资源，random processing 所带来的时延最大。对于 offloading processing 而言，任务全部交由边缘服务器处理。虽然可以充分发挥边缘服务器

的计算资源，但是，未能考虑计算资源和服务缓存资源的相互关系。边缘服务器因为缓存资源不足将从云端下载任务计算所需的服务应用，带来额外的时延。对于本文算法而言，它联合考虑了计算卸载和服务缓存，通过本地处理和边缘处理的合理调度，促使了计算资源和缓存资源的充分利用，进一步减少了任务的时延。此外，深度强化学习算法有利于在动态的网络环境当中做出最优的卸载决策，有效地处理好计算资源和服务缓存资源之间的关系，进而保障任务的快速处理。

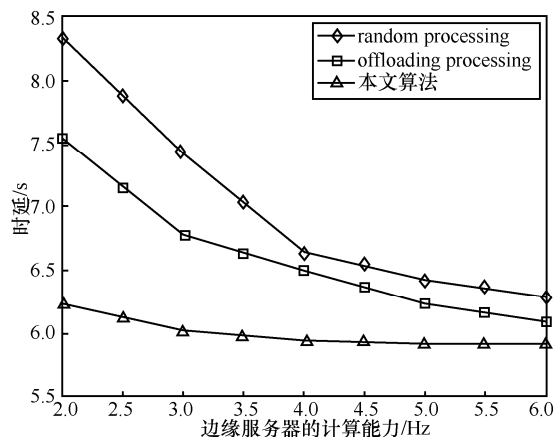


图 4 边缘服务器的计算能力对不同算法时延的影响

图 5 描述了边缘服务器的缓存能力对不同算法时延的影响。从图 5 中可以发现，随着边缘服务器缓存能力的增加，不同算法处理任务的时延随之减少。这主要是因为边缘服务器为了执行任务，需要安装相应的服务应用，否则就需要从云端下载，从而带来了额外的开销。当边缘服务器的缓存能力增加时，可以缓存更多任务处理所需要的服务应用。这样方便任务卸载给边缘服务器之后直接计算，从而降低了时延。

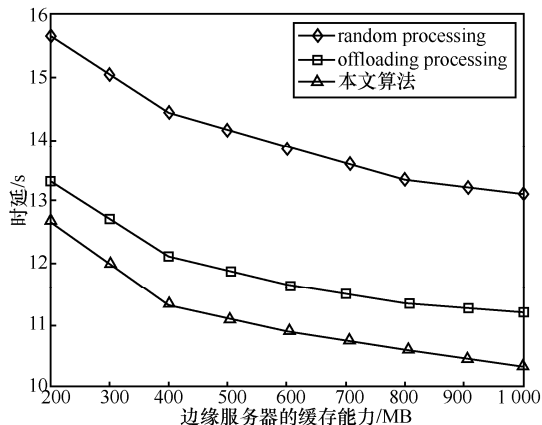


图 5 边缘服务器的缓存能力对不同算法时延的影响

图6描述了本文算法在不同学习速率场景下的收敛情况。其中,实线表示当actor和critic网络的学习速率分别为 1×10^{-5} 和 1×10^{-4} 时episode数目对奖励的影响。虚线表示当actor和critic网络的学习速率分别为 1×10^{-4} 和 1×10^{-3} 时episode数目对奖励的影响。从两者的比较可以发现,随着episode的增加,奖赏将会趋于稳定。

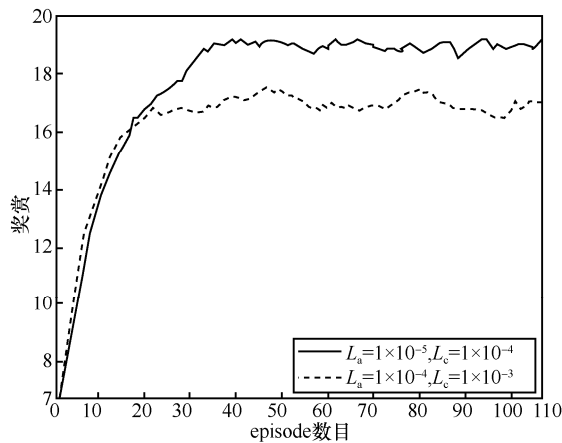


图6 本文算法在不同学习速率场景下的收敛情况

6 结束语

面对车联网中有限的网络资源,为了保障大量用户多样化的服务需求,本文提出了智能驱动的车载边缘计算架构。该架构实现了纵向端-边-云资源的协作和横向端侧、边侧资源的协同,有利于实现资源的最大化利用。基于该架构,探究了计算卸载和服务缓存相互作用的机理,进而提出了两者的联合优化模型。考虑到复杂的车载环境,利用异步优势的actor-critic算法,给出了最优的任务卸载的策略和资源管理方案。实验结果表明,相对于对比算法,本文算法在任务处理时延方面取得了良好的性能提升。

参考文献:

- [1] 中国信息通信研究院. 车联网白皮书[R]. 北京: 中国信息通信研究院, 2017.
CAICT. White paper of Internet of vehicles[R]. Beijing: CAICT, 2017.
- [2] 郭辉, 芮兰兰, 高志鹏. 车辆边缘网络中基于多参数MDP模型的动态服务迁移策略[J]. 通信学报, 2020, 41(1): 1-14.
GUO H, RUI L L, GAO Z P. Dynamic service migration strategy based on MDP model with multiple parameter in vehicular edge network[J]. Journal on Communications, 2020, 41(1): 1-14.
- [3] 张海波, 王子心, 贺晓帆. SDN和MEC架构下V2X卸载与资源分配[J]. 通信学报, 2020, 41(1): 114-124.
ZHANG H B, WANG Z X, HE X F. V2X offloading and resource allocation under SDN and MEC architecture[J]. Journal on Communications, 2020, 41(1): 114-124.
- [4] LIU L, CHEN C, QIU T, et al. A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs[J]. Vehicular Communications, 2018, 13: 78-88.
- [5] CHEN C, WANG C, QIU T, et al. Caching in vehicular named data networking: architecture, schemes and future directions[J]. IEEE Communications Surveys & Tutorials, 2020, 22(4): 2378-2407.
- [6] 彭鑫, 邓清勇, 田淑娟, 等. 多信道车联网V2R/V2V数据传输调度算法[J]. 通信学报, 2019, 40(3): 92-101.
PENG X, DENG Q Y, TIAN S J, et al. Data dissemination scheduling algorithm for V2R/V2V in multi-channel VANET[J]. Journal on Communications, 2019, 40(3): 92-101.
- [7] FENG J, YU F R, PEI Q, et al. Cooperative computation offloading and resource allocation for blockchain-enabled mobile edge computing: a deep reinforcement learning approach[J]. IEEE Internet of Things Journal, 2020, 7(7): 6214-6228.
- [8] FENG J, YU F R, PEI Q, et al. Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems[J]. IEEE Transactions on Wireless Communications, 2020, 19(6): 4321-4334.
- [9] LIU L, CHEN C, PEI Q, et al. Vehicular edge computing and networking: a survey[J]. Mobile Networks and Applications, 2020, doi: 10.1007/s11036-020-01624-1.
- [10] WANG T, TANG M B, CAO Y. Resource optimization protocol based on multicommunity model for intermittently connected mobile networks[J]. IEEE Systems Journal, 2019, 14(1): 410-421.
- [11] WANG T, CAO Y, ZHOU Y, et al. A survey on geographic routing protocols in delay/disruption tolerant networks[J]. International Journal of Distributed Sensor Networks, 2016, 12(2): 1-12.
- [12] DAI Y, XU D, MAHARJAN S, et al. Joint load balancing and offloading in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2018, 6(3): 4377-4387.
- [13] TAMANI N, BRIK B, LAGRAA N, et al. On link stability metric and fuzzy quantification for service selection in mobile vehicular cloud[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(5): 2050-2062.
- [14] YANG C, LIU Y, CHEN X, et al. Efficient mobility-aware task offloading for vehicular edge computing networks[J]. IEEE Access, 2019, 7: 26652-26664.
- [15] SORKHOH I, EBRAHIMI D, ATALLAH R, et al. Workload scheduling in vehicular networks with edge cloud capabilities[J]. IEEE Transactions on Vehicular Technology, 2019, 68(9): 8472-8486.
- [16] DU J, YU F R, CHU X, et al. Computation offloading and resource

allocation in vehicular networks based on dual-side cost minimization[J]. IEEE Transactions on Vehicular Technology, 2018, 68(2): 1079-1092.

- [17] TAREQ M M K, SEMIARI O, SALEHI M A, et al. Ultra reliable, low latency vehicle-to-infrastructure wireless communications with edge computing[C]//2018 IEEE Global Communications Conference. Piscataway: IEEE Press, 2018: 1-7.
- [18] ZHANG K, MAO Y, LENG S, 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.
- [19] LIU Y, YU H, XIE S, et al. 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.
- [20] HE Y, ZHAO N, YIN H. Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2017, 67(1): 44-55.
- [21] HU R Q. Mobility-aware edge caching and computing in vehicle networks: a deep reinforcement learning[J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 10190-10203.
- [22] NING Z, ZHANG K, WANG X, et al. Joint computing and caching in 5G-envisioned Internet of vehicles: a deep reinforcement learning-based traffic control system[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, doi: 10.1109/TITS.2020.2970276.
- [23] CHEN C, LIU L, QIU T, et al. ASGR: an artificial spider-web-based geographic routing in heterogeneous vehicular networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 20(5): 1604-1620.
- [24] PENG H, LI D, ABBOD K, et al. Performance analysis of IEEE 802.11p DCF for multiplatooning communications with autonomous vehicles[J]. IEEE Transactions on Vehicular Technology, 2016, 66(3): 2485-2498.

[作者简介]



刘雷 (1987-)，男，河南南阳人，博士，西安电子科技大学讲师，主要研究方向为车载边缘计算。



陈晨 (1977-)，男，陕西西安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为智能交通。



冯杰 (1987-)，女，陕西咸阳人，博士，西安电子科技大学副教授，主要研究方向为边缘智能。



裴庆祺 (1975-)，男，广西百色人，西安电子科技大学教授、博士生导师，主要研究方向为区块链。



何辞 (1983-)，女，湖北武汉人，中国电子科技集团公司第 54 研究所高级工程师，主要研究方向为空天信息网络。



窦志斌 (1980-)，男，山西阳泉人，中国电子科技集团公司第 54 研究所高级工程师，主要研究方向为空天信息网络。