

# 车载边缘计算中基于深度强化学习的协同计算卸载方案



范艳芳 袁爽 蔡英 陈若愚

北京信息科技大学计算机学院 北京 100101

**摘要** 车载边缘计算(Vehicular Edge Computing, VEC)是一种可实现车联网低时延和高可靠性的关键技术,用户将计算任务卸载到移动边缘计算(Mobile Edge Computing, MEC)服务器上,不仅可以解决车载终端计算能力不足的问题,而且可以减少能耗,降低车联网通信服务的时延。然而,高速公路场景下车辆移动性与边缘服务器静态部署的矛盾给计算卸载的可靠性带来了挑战。针对高速公路环境的特点,研究了临近车辆提供计算服务的可能性。通过联合 MEC 服务器和车辆的计算资源,设计并实现了一个基于深度强化学习的协同计算卸载方案,以实现在满足任务时延约束的前提下最小化所有任务时延的目标。仿真实验结果表明,相比于没有车辆协同的方案,所提方案可以有效降低时延和计算卸载失败率。

**关键词:** 移动边缘计算;车载边缘计算;计算卸载;深度强化学习;协同计算

**中图法分类号** TN929.5

## Deep Reinforcement Learning-based Collaborative Computation Offloading Scheme in Vehicular Edge Computing

FAN Yan-fang, YUAN Shuang, CAI Ying and CHEN Ruo-yu

School of Computer, Beijing Information Science & Technology University, Beijing 100101, China

**Abstract** Vehicular edge computing (VEC) is a key technology that can realize low latency and high reliability of internet of vehicles. Users offload computing tasks to mobile edge computing (MEC) servers, which can not only solve the problem of insufficient computing capability of vehicles, but also reduce the energy consumption and the latency of communication service. However, the contradiction between the mobility of vehicles and the static deployment of edge servers in highway scenarios poses a challenge to the reliability of computing offloading. To solve this problem, this paper designs a collaborative deep reinforcement learning-based scheme for vehicles to adapt to the dynamic high-speed environment by combining the computing resources of MEC servers and neighboring vehicles. Simulation results show that compared with the scheme without vehicle collaboration, this scheme can reduce the delay and the failure rate of offloading.

**Keywords** Mobile edge computing, Vehicular edge computing, Computation offloading, Deep reinforcement learning, Collaborative computing

### 1 引言

随着 5G 时代的到来,新兴的各种计算密集型、延迟敏感型的应用和服务不断涌现,例如在线导航、路况识别与综合分析、高精度地图实时分发、碰撞预警、远程控车、辅助驾驶等,这对资源有限的移动设备带来了巨大挑战。移动边缘计算是用于解决计算任务的快速增长和用户计算能力有限之间的矛盾的模式<sup>[1]</sup>。MEC 将信息服务从云端下沉到无线接入网络

的边缘,为用户创建了高带宽、低时延的高性能服务环境。通过将计算任务从移动设备转移到临近的 MEC 服务器上进行处理,可以降低移动设备的执行时延和能耗,从而大大提高用户服务质量。

车载边缘计算是车联网与 MEC 结合的典型<sup>[2]</sup>。尽管 VEC 源自于 MEC,但 VEC 有它的独特之处:1)与 MEC 适用于物联网环境的场景不同,VEC 为智能交通系统提供更具体的服务;2)与 MEC 相比,基于 VEC 的计算卸载在时延与能

收稿日期:2020-10-03 返修日期:2021-01-05 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61672106);北京市自然科学基金(L192023);北京信息科技大学基金(2025028);北京信息科技大学研究生科技创新项目、网络文化与数字传播北京市重点实验室开放课题

This work was supported by the National Natural Science Foundation of China(61672106), Natural Science Foundation of Beijing(L192023), Foundation of Beijing Information Science & Technology University(2025028), Graduate Science and Technology Innovation Project of Beijing Information Science & Technology University and Open Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research.

通信作者:范艳芳(fyhappy@bistu.edu.cn)

耗方面有更严格的要求;3)与 MEC 适用的相对静止的物联网设备不同,VEC 主要适用于车辆终端和路边智能基础设施。车辆的移动性比物联网设备更强,尤其是在高速公路场景下,静态部署的轻量级边缘服务器无法满足高速移动车辆的低时延、高可靠的要求。因此,为了扩展 VEC 网络的计算服务范围并提高灵活性,本文利用高速公路中车辆具有的方向相同、速度相似的特点,研究使用临近移动车辆为 VEC 提供计算资源,以解决 MEC 服务器静态部署不足的问题。

另外,动态多变的高速公路环境下的资源分配问题是一个具有复杂的目标函数和多个约束的非凸优化问题,这是难以解决的<sup>[3]</sup>。人工智能是一种在解决通信网络中的无线电和计算资源分配问题时应对大数据挑战的高级工具<sup>[4]</sup>。深度强化学习(Deep Reinforcement Learning, DRL)在人工智能中扮演着重要的角色,在人机博弈、机器人控制和对话生成等许多领域都取得了巨大的成功。DRL 由于对时变环境具有强大的认知能力,在车辆网络中得到了广泛的应用<sup>[5]</sup>。因此,本文设计并实现了基于 DRL 的高效计算卸载方案。

本文的主要贡献在于充分研究了高速公路这一特定场景的特点,综合考虑了请求服务车辆周围车辆的移动性、计算能力和剩余电量等因素,将计算卸载决策问题转化为马尔可夫的决策过程,并使用深度强化学习算法做出最佳的卸载决策。仿真结果表明,所提方案可以在降低任务失败率和时延方面达到很好的效果。

## 2 相关工作

随着智能交通系统的兴起和新的车辆应用的不断开发,车辆边缘计算被认为是最有前途的领域之一。在车载边缘计算中,为了解决边缘服务器计算能力不足而无法处理所有任务的问题,许多研究关注各种资源的充分利用。Zhang 等<sup>[6]</sup>考虑到了边缘服务器上的计算资源的负载平衡和车载网络的高动态性,引入了光纤无线(FiWi)技术,使用云端资源来增强车辆边缘计算网络。Khan 等<sup>[7]</sup>将任务请求分为卸载到边缘服务器的延迟非容忍的任务和卸载到云服务器的延迟容忍任务,并设计了车辆提供计算资源和存储资源的方案,以解决边缘服务器资源不足的问题。一些学者探索利用车辆资源来增强网络服务。Huang 等<sup>[8]</sup>提出了车辆邻居组的概念,方便了车辆与车辆之间共享相似的服务。Qiao 等<sup>[9]</sup>基于任务相似性和计算能力,将车辆划分为两个子云,提出了一种协同任务卸载方案,有效地减少了向 MEC 服务器传输相似任务的次数。Dai 等<sup>[10]</sup>研究了移动医疗应用程序的特点,使用附近合适的车辆帮助卸载移动医疗任务,提高了网络资源的利用率。Huang 等<sup>[11]</sup>和 Li 等<sup>[12]</sup>使用停车场内停车的资源来增加可用网络资源,设计了计算卸载算法来解决停车之间的工作量分配问题。

在进行计算卸载决策时,现有的研究多采用优化方法或博弈论方法。随着人工智能领域的不断发展,强化学习乃至深度强化学习也被应用于卸载决策。Dai 等<sup>[10]</sup>针对在满足应用程序完成时间的前提下,能量和资源消耗最小化的问题,提出了一种基于粒子群优化的方法来优化该计算卸载问题。

Zhang 等<sup>[6]</sup>使用基于博弈论的最近任务卸载算法和近似的负载均衡任务卸载算法来解决时延最小化的问题。Huang 等<sup>[11]</sup>将用户、MEC 服务器和停车场之间的关系建模为 Stackelberg 博弈模型,提出了一种基于次梯度的迭代算法,该算法可以降低用户的总体成本。Li 等<sup>[12]</sup>考虑到了提供服务的车辆具有自私性,提出了一个基于价格合同的三阶段 Stackelberg 博弈机制,用于最大化车辆、运营商和停车场代理商三者的效用。Liu 等<sup>[13]</sup>针对任务的特点,将任务分为不可分割的任务和可分割的任务,分别采用了一种改进的上置信界算法和 Q 学习算法来找到最有效的卸载策略。Ning 等<sup>[14]</sup>通过联合利用许可的蜂窝频谱和非许可的信道,为 5G 车载网络构建了一个智能卸载框架,提出了一种双边匹配算法来调度无执照频谱,并提出了一种基于 DRL 的方法来简化系统状态,以实现分布式卸载,进而最小化成本。

现有的研究方案较多地集中在如何扩展和充分利用 MEC 服务器资源上,尚缺乏对高速公路这种特定场景的研究。随着我国交通行业的发展,高速公路规模不断扩大,每天在高速公路上通行的车辆数量巨大。这些车辆的高速移动性使得仅通过卸载到静态部署的 MEC 服务器或城市周边的停车均无法实现高度可靠的计算卸载。因此,非常有必要针对高速公路场景的特点,设计相应的卸载方案。

## 3 系统模型

### 3.1 车载边缘计算架构

本文在车载边缘计算网络(见图 1)中考虑单向直行公路的场景,公路的一侧有一个覆盖半径为  $r$  的路边单元(Road Side Unit, RSU),RSU 上部署了 MEC 服务器,MEC 服务器的计算资源为  $f^{\text{MEC}}$ 。用集合  $V = \{1, 2, 3, \dots, v\}$  表示 RSU 覆盖范围内的所有车辆,对于任意车辆  $i$ ,设备的可用计算资源为  $f^i$ 。为了便于参考,表 1 列出了文中的关键符号及其定义。

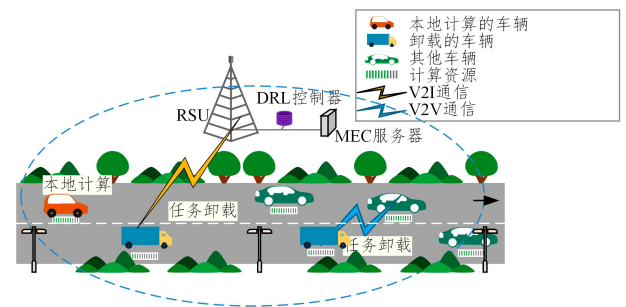


图 1 车载边缘计算网络框架

Fig. 1 Vehicular edge computing network framework

产生任务的车辆称为任务车辆(Task Vehicle, TaV),用集合  $V_t (V_t \subseteq V)$  表示。一些具有空闲资源且可能会提供服务的车辆,称为服务车辆(Candidate Service Vehicle, CsV),用集合  $V_c (V_c \subseteq V)$  表示。假设以 RSU 为坐标原点,道路为  $x$  轴,车辆行驶方向为正方向,每辆车以速度  $v_i$  行驶,则车辆  $i$  所在的位置坐标为  $l_i = (x_i, y_i)$ ,任务产生后该车辆保持在该范围内的时间为:

$$T_i^{v2i} = (\sqrt{r^2 - y_i^2} - x_i) / v_i \quad (1)$$

表1 符号定义

Table 1 Definition of notations

符号	定义
$a_i, a_{i,j}$	车辆 $i$ 的动作集, 车辆 $i$ 的卸载策略
$a_{i,t}, a_t, A$	$t$ 时刻车辆 $i$ 的动作集, 系统选择的动作, 动作空间
$b^i, b_1, b_2$	车辆 $i$ 剩余电量百分比, 剩余电量的大、小阈值
$B^{V2I}, B^{V2V}$	V2I, V2V 通信的带宽 (Hz)
$C_i$	任务所需的计算资源 (cycles)
$d_{i,j}$	车辆 $i, j$ 间的瞬时距离 (m)
$d_{range}^{V2I}, d_{range}^{V2V}$	V2I 最远通信距离, V2V 最远通信距离 (m)
$\bar{d}_i$	车辆 $i$ 在 RSU 覆盖范围内的平均距离 (m)
$\bar{d}_{i,j}$	任务产生后车辆 $i, j$ 能通信的平均距离 (m)
$D, D_i$	经验存储器, 任务的大小 (bit)
$f^i, f^{MEC}$	车辆 $i$ 自身的计算能力, MEC 的计算能力 (Hz)
$f_j^i, f_i^{Vc_i}$	车辆 $j$ 可用计算能力, 车辆 $i$ 的所有候选服务车辆的可用计算能力 (Hz)
$G$	车载设备的发送功率 (W)
$h, H_i, H_{i,j}$	上传链路的信道衰落因子, 车辆 $i$ 与 RSU、车辆 $i$ 与车辆 $j$ 通信的信道增益参数
$K$	梯度下降的步长
$l_i = (x_i, y_i)$	车辆 $i$ 的位置 (横坐标, 纵坐标)
$L, Loss(\theta)$	路径损耗, 损失函数
$mat$	车辆之间是否有可能成为彼此的候选服务车辆的矩阵
$N_0$	高斯白噪声功率 (dB)
$P_{i,j}^B, P_{i,j}^E, P_{i,j}^T$	车辆 $j$ 的剩余电量、可用计算资源、移动性对其能否成为服务车辆 $i$ 的候选车辆的影响
$Q(s_i, a, \theta), \theta$	主神经网络及其参数
$Q'(s_i, a, \theta'), \theta'$	目标神经网络及其参数
$r, r_i, R$	RSU 覆盖半径 (m), 即时奖励, 未来奖励
$s_t, S, S_{i,t}$	$t$ 时刻系统的环境、状态空间, $t$ 时刻车辆 $i$ 的环境
$\bar{r}_i^{V2I}, \bar{r}_{i,j}^{V2V}$	任务车辆 $i$ 与 RSU 之间、与车辆 $j$ 之间的平均数据传输速率 (bit/s)
$t_1, t_2$	两车保持在通信范围内时间的大、小阈值
$t_i(a_{i,j})$	计算卸载策略 $a_{i,j}$ 下的任务时延 (s)
$t_i^{Loc}, t_i^{V2I}, t_i^{V2V}$	本地计算、V2I 卸载、V2V 卸载的时延 (s)
$tmax_i$	任务最大可容忍延迟 (s)
$T_i$	任务车辆 $i$ 请求的任务
$T_i^{V2I}$	车辆 $i$ 的任务产生后在 RSU 覆盖范围内的行驶时间 (s)
$T_{i,j}^{V2V}$	任务产生后, 车辆 $i, j$ 的瞬时距离小于 V2V 通信最远通信距离的保持时间 (s)
$U$	样本数
$v, v_i$	车辆总数, 车辆 $i$ 的行驶速度 (m/s)
$V, Vc_i$	车辆集合, 车辆 $i$ 的候选服务车辆的集合
$Vc, Vt$	服务车辆集合, 任务车辆集合
$y, y'$	主神经网络的值, 目标神经网络的值
$\sigma, \alpha$	路径损耗因子, 学习速率, 折扣因子
$\tau$	判断车辆能否作为候选服务车辆的阈值
$\epsilon$	贪心算法的探索速率

任意 TaV  $i$  产生的任务由三元组  $T_i = (D_i, C_i, tmax_i)$  表示。其中,  $D_i$  为任务大小,  $C_i$  为任务所需的计算资源,  $tmax_i$  为任务最大可容忍延迟。用集合  $Vc_i, f_i^{Vc_i}$  分别表示任务车辆  $i$  周围的候选服务车辆集合及其可用计算资源。任务的处理方式有 3 种(本文暂不考虑多跳卸载): 本地处理、单跳 V2I (Vehicle-to-Infrastructure) 方式卸载到 MEC 服务器和单跳 V2V (Vehicle-to-Vehicle) 方式卸载到临近的候选服务车辆。使用 0-1 变量  $a_{i,j}$  来表示 TaV 执行任务的方式:

$$a_{i,j} = \begin{cases} 0, & j \in V, T_i \text{ 本地执行} \\ 1, & j = 0, T_i \text{ 卸载到 MEC 服务器} \\ 1, & j \in Vc_i, T_i \text{ 卸载到 CsV } j \end{cases} \quad (2)$$

对于任意 TaV  $i$ , 最多只能选择 3 种方式中的一种进行处理, 即:

$$\sum_{j \in V \cup \{0\}} a_{i,j} \leq 1 \quad (3)$$

任意一个 CsV  $j$  在同一时刻最多只能为一辆 TaV 提供服务, 即:

$$\sum_{i \in Vt} a_{i,j} \leq 1 \quad (4)$$

### 3.2 候选服务车辆判别模型

对于任务车辆  $i$  而言, 其候选服务车辆需要同时满足以下几点。

1) 任务卸载期间候选服务车辆与任务车辆的距离应满足通信要求。令  $T_{i,j}^{V2V}$  表示任务产生后, 车辆  $j$  保持在任务车辆  $i$  通信范围内 ( $d_{i,j} \leq d_{range}^{V2V}$ ) 的时间。TaV  $i$  和车辆  $j$  保持在通信范围内的时间越长, 车辆  $j$  成为候选服务车辆的可能性就越大。用  $P_{i,j}^T$  表示车辆  $j$  的移动性对其能否成为 TaV  $i$  的候选服务车辆的可能性的影响:

$$P_{i,j}^T = \begin{cases} 1, & T_{i,j}^{V2V} > t_1 \\ \sqrt{T_{i,j}^{V2V}(2t_1 - T_{i,j}^{V2V})} / t_1, & t_2 \leq T_{i,j}^{V2V} \leq t_1 \\ 0, & T_{i,j}^{V2V} < t_2 \end{cases} \quad (5)$$

其中,  $t_1$  表示两个车辆保持在通信范围内时间足够长的阈值的上限,  $t_2$  表示两个车辆保持在通信范围内时间阈值的下限,  $t_1 > t_2$ 。此外,  $\sqrt{T_{i,j}^{V2V}(2t_1 - T_{i,j}^{V2V})}$  表示可能性对移动性的敏感程度: 当  $T_{i,j}^{V2V}$  较小时, 车辆  $j$  为 TaV  $i$  提供服务的可能性随着  $T_{i,j}^{V2V}$  的减小快速下降; 当  $T_{i,j}^{V2V}$  较大时, 车辆  $j$  为 TaV  $i$  的可能性随着  $T_{i,j}^{V2V}$  的增大缓慢增大。

2) 候选服务车辆应具有足够的计算能力为任务车辆提供服务。用  $P_{i,j}^E$  表示车辆  $j$  的可用计算能力对其能否成为 TaV  $i$  的候选服务车辆的可能性的影响:

$$P_{i,j}^E = \begin{cases} 1, & f_j^i > f^i \\ 0, & f_j^i \leq f^i \end{cases} \quad (6)$$

其中,  $f_j^i$  表示车辆  $j$  可以为 TaV  $i$  提供的计算资源。只有车辆  $j$  可提供的计算资源比 TaV  $i$  所需的计算资源多时, 车辆  $j$  才有可能成为车辆  $i$  的候选服务车辆。

3) 候选服务车辆应具有足够的剩余电量为任务车辆提供服务。用  $P_{i,j}^B$  表示车辆  $j$  当前剩余电量对车辆  $j$  能否成为 TaV 的候选服务车辆的可能性的影响:

$$P_{i,j}^B = \begin{cases} 1, & b_j^i > b_1 \\ \sqrt{b_j^i(2 - b_j^i)}, & b_2 \leq b_j^i \leq b_1 \\ 0, & b_j^i < b_2 \end{cases} \quad (7)$$

其中,  $b_j^i$  是  $t$  时刻车辆  $j$  的电池剩余电量百分比,  $b_1$  表示剩余电量充足时车辆  $j$  愿意提供服务的阈值,  $b_2$  表示剩余电量不足时车辆  $j$  拒绝提供服务的阈值。类似地,  $\sqrt{b_j^i(2 - b_j^i)}$  表示可能性对剩余电量的敏感程度: 电池剩余电量低时车辆  $j$  为 TaV  $i$  提供服务的可能性随着剩余电量的降低会快速下降; 电池剩余电量高时该可能性随着剩余电量的升高缓慢升高。

综上所述, 用  $P_{i,j}^T, P_{i,j}^E, P_{i,j}^B$  表示移动性、可提供计算能力、剩余电量的综合效果, 用  $\tau$  表示判断车辆能否作为候选服务车辆的阈值。当  $P_{i,j}^T, P_{i,j}^E, P_{i,j}^B \geq \tau$  时, 车辆  $j$  为 TaV  $i$  的候选服务车辆, 即  $j \in Vc_i$ , 否则,  $j \notin Vc_i$ 。用  $n$  行  $n$  列的矩阵  $mat$  表示所有车辆之间是否有可能成为彼此的候选服务车辆, 即:

$$mat = \begin{cases} 1, & P_{i,j}^T, P_{i,j}^E, P_{i,j}^B \geq \tau \\ 0, & P_{i,j}^T, P_{i,j}^E, P_{i,j}^B < \tau \end{cases} \quad (8)$$

### 3.3 通信模型

#### 3.3.1 V2I 通信模型

车辆与 RSU 之间是以 C-V2X (Cellular-V2X) 的直连链路进行通信的。本文将车辆到 RSU 的上传链路设定为平坦瑞利衰落信道,不考虑信道干扰。根据香农公式可以计算出车辆  $i$  从任务产生到驶出 RSU 范围期间上传链路的平均数据传输速率,计算式为:

$$\overline{r_i^{V2I}} = B^{V2I} \log_2 \left( 1 + \frac{GH_i}{N_0} \right) \quad (9)$$

其中,  $B^{V2I}$  代表上传链路信道的带宽,  $N_0$  为高斯白噪声功率,  $G$  表示车载设备的发送功率,  $H_i$  为信道增益参数。  $H_i = h^2 \overline{d_i}^{-\sigma}$ ,  $h$  为上传链路的信道衰落因子,  $\overline{d_i}^{-\sigma}$  为车辆与 RSU 之间的路径损耗,  $\sigma$  为路径损耗因子,  $\overline{d_i}$  为从任务产生到任务处理结束期间车辆  $i$  与 RSU 的平均距离。为简化问题,取最大容忍时间的平均距离,即:

$$\overline{d_i} = \frac{\int_t^{t+t_{\max}} d_i(t) dt}{T_i^{V2I}}$$

#### 3.3.2 V2V 通信模型

在 V2V 通信网络中,车与车之间的通信使用短程无线通信方式中的 IEEE 802.11p 协议。另外,本文的场景中 V2V 通信采用正交频率,可以减小车辆之间的相互影响。TaV  $i$  与 CsV  $j$  之间的平均数据传输速率可以表示为:

$$\overline{r_{i,j}^{V2V}} = B^{V2V} \log_2 \left( 1 + \frac{GH_{i,j}}{N_0} \right) \quad (10)$$

其中,信道增益参数  $H_{i,j} = h^2 L^{V2V}$ , 路径损耗  $L = 63.3 + 17.7 \log_{10}(\overline{d_{i,j}})$  dB,  $\overline{d_{i,j}}$  为 TaV  $i$  与 CsV  $j$  保持在通信范围内的平均距离。因此,路径损耗数值的形式为:  $L^{V2V} = 10^{L/10}$  dB。

### 3.4 计算模型

#### 3.4.1 本地计算模型

当 TaV 选择用自身的计算设备处理任务时,本地计算时延为:

$$t_i^{\text{loc}} = C_i / f^i \quad (11)$$

#### 3.4.2 V2I 计算模型

当 TaV 选择通过 V2I 卸载到 MEC 服务器时,卸载过程如下:TaV 上传任务到 RSU, MEC 服务器计算任务后,由 RSU 将结果返回给 TaV。由于任务计算结果的数据量很小,本文忽略结果的返回时间。因此, TaV  $i$  选择 V2I 卸载的任务时延为上传任务时间与计算任务时间之和,即:

$$t_i^{V2I} = D_i / \overline{r_i^{V2I}} + C_i / f^{\text{MEC}} \quad (12)$$

注意,在 V2I 卸载模式下,RSU 需要在 TaV 驶出 RSU 覆盖范围前将结果返回给 TaV,即:

$$t_i^{V2I} \leq T_i^{V2I} \quad (13)$$

#### 3.4.3 V2V 计算模型

当 TaV 选择通过 V2V 卸载到候选服务车辆  $j$  时,卸载过程如下:TaV  $i$  上传任务到 CsV  $j$ , CsV  $j$  计算任务  $T_i$  后将结果返回给 TaV  $i$ 。因此,忽略返回时延时, TaV  $i$  选择 V2V 卸载所需的任务时延为:

$$t_{i,j}^{V2V} = D_i / \overline{r_{i,j}^{V2V}} + C_i / f^j \quad (14)$$

综上所述,在任意时刻  $t$  下,根据计算卸载策略  $a_{i,j}$ ,

$T_i$  的任务时延为:

$$t_i(a_{i,j}) = \begin{cases} t_i^{\text{loc}}, & a_{i,j} = 0, j \in V \\ t_i^{V2I}, & a_{i,j} = 1, j = 0 \\ t_{i,j}^{V2V}, & a_{i,j} = 1, j \in Vc_i \end{cases} \quad (15)$$

### 3.5 问题描述

为了降低卸载失败率,本文在满足任务时延约束下以最小化所有任务时延为目标:

$$\begin{aligned} \min \sum_{i \in V_t} t_i(a_{i,j}) &= \min \left( (1 - a_{i,j}) t_i^{\text{loc}} + a_{i,0} t_i^{V2I} + \sum_{j \in Vc} a_{i,j} t_{i,j}^{V2V} \right) \\ \text{s. t. } &\begin{cases} a_{i,j} \in \{0, 1\} \\ \text{式(2), 式(3), 式(12), 式(13), 式(14)} \end{cases} \end{aligned} \quad (16)$$

该问题可以通过计算决策变量  $a_{i,j}$  来求解。其由于包含二元变量且存在许多约束条件,因此很难求解。本文使用深度强化学习算法求解最优策略。

## 4 基于深度强化学习的卸载算法

本节首先将满足时延约束条件的时延最小化问题转化为马尔可夫决策过程 (Markov Decision Process, MDP), 然后介绍 Q-learning、深度 Q 网络 (Deep Q Network, DQN) 和双层深度 Q 网络 (Double Deep Q Network, DDQN), 最后设计基于 DDQN 的计算卸载算法。

### 4.1 基于强化学习的计算卸载

MDP 可形式化为四元组,即  $\{S, A, P(s_{t+1} | s_t, a_t), R(s_t, a_t)\}$ , 其中,集合  $S$  表示环境的状态空间,集合  $A$  表示一组可能的动作,  $P(s_{t+1} | s_t, a_t)$  表示在状态  $s_t$  处执行动作  $a_t$  后转移为状态  $s_{t+1}$  的概率,  $R(s_t, a_t)$  表示在状态  $s_t$  执行动作  $a_t$  后接收到的奖励。MDP 模型的目标是在长期  $T$  上获得最大的累积奖励  $R$ 。

MDP 实质上是一个离散时间随机控制过程。MDP 的代理人 (agent) 与环境 (environment) 交互过程分为一系列子序列。一个任务卸载周期  $T$  被分为多个离散时间步 (time step), 终端设备在每个时间步  $t$  ( $t=1, 2, \dots$ ) 的子序列被称为一个片段 (episode)。在交互过程中,MDP 从一个随机的初始状态  $s_1$  开始迭代直至最终收敛。在每个状态  $s_t$  时,每个 TaV 从可选动作集中选择一个动作  $a_{i,t}$ , 然后代理人计算该动作对应的奖励  $r_t$ , 接着 TaV 进入下一个状态  $s_{t+1}$ 。下面将计算卸载问题 (16) 转化为 MDP。

#### 4.1.1 状态空间

TaV  $i$  的状态由其位置、速度、车辆为其他车辆提供服务的可行性矩阵、自身的计算能力、MEC 服务器计算能力,以及候选车辆的计算能力组成,即:

$$s_i = \{x_i, y_i, v_i, \text{mat}, f^i, f^{\text{MEC}}, f_{i,j}^{Vc_i}\} \quad (17)$$

则整个系统的状态  $S$  由所有车辆的位置、速度、计算能力、车辆为其他车辆提供服务的可行性矩阵、MEC 服务器计算能力组成。

#### 4.1.2 动作空间

在提议的 VEC 网络中,本文选择部署在 RSU 上的 DRL 控制器作为代理人,负责与环境的交互和计算决策。为了维持所有 TaV 的动作集维度的一致性,将 MEC 服务器和所有车辆都作为 TaV  $i$  的动作集。因此, TaV  $i$  的动作集表述为:

$$a_i = \{a_{i,0}, a_{i,1}, \dots, a_{i,j}, \dots, a_{i,v}\} \quad (18)$$

因此,整个系统的动作空间  $A$  由所有车辆的动作组成。

当在训练过程中选择非候选车辆、所选方式时延不满足任务的容忍需求,或者用户在任务处理完成前离开其选择服务车辆或 RSU 的通信范围时任务卸载失败,该动作无效。

#### 4.1.3 奖励函数

由于代理人以时延约束下最小化所有任务总时延为目标,因此即时奖励函数应该与时延成反比。为避免局部最优,需要保证奖励反馈到 DQN 中时更容易泛化,将 TaV 的任意行动的奖励都归一到  $[-1, 0]$ 。当选择无效的动作时,奖励为最小值,即  $-1$ 。当选择有效动作时,奖励函数为:

$$r_t = -t_t(a_{i,j})/tmax_i \quad (19)$$

#### 4.1.4 Q-learning 方法

由于很难在 MDP 问题中获得转移概率  $P(s_{t+1} | s_t, a_t)$ , 本文选择一种典型的无模型强化学习算法——Q-learning, 它非常适合用于解决车辆边缘计算网络中的决策问题。

累积的未来奖励为  $R_t = \sum_0^T \lambda r_t$ ,  $\lambda$  是折扣因子。本文的主要目标是使所有任务车辆的长期累积奖励最大化, 即  $\max \mathbb{E}(\sum_0^T \lambda r_t)$ 。

Q 函数定义为  $Q(s_t, a_t)$ , 则 Q 值表示给定状态  $s_t$  时, 采取动作  $a_t$  的质量(即期望回报), 表达式为:

$$Q^*(s_t, a_t) = \mathbb{E}(\sum_0^T \lambda r_t | s_t = s, a_t = a, \pi) \quad (20)$$

当策略  $\pi$  对于所有状态都能使期望回报最大时, 策略  $\pi$  为最优策略。通过遵循 Bellman 准则, 最佳 Q 函数估计为:

$$Q^*(s_t, a_t) = \mathbb{E}_{a_{t+1}}[r_t + \lambda \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})] \quad (21)$$

通常, 在状态  $s_{t+1}$  上通过五元组  $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$  的信息以迭代方式获得 Q 函数, 此后更新的 Q 函数可以表示为:

$$Q_{t+1}^*(s, a) = (1 - \alpha) Q(s_t, a_t) + \alpha(r_t + \lambda \max_{a_{t+1} \in A} Q^*(s_{t+1}, a_{t+1})) \quad (22)$$

其中,  $\alpha \in [0, 1]$  表示学习速率, 通过使用适当的学习速率可确保  $Q_t(s, a)$  必定收敛于  $Q^*(s, a)$ 。最佳动作集  $A$  可以通过每个状态  $a$  重复更新 Q 值获得。

#### 4.2 基于深度强化学习的计算卸载

尽管通过使用 Q-learning 算法可以有效解决较少状态动作场景下的强化学习问题, 但是在高度动态的高速公路场景中, 系统的状态动作对很多, 使用 Q 表存储状态动作对的概率是一种很低效的做法。用深度神经网络(DNN)拟合 Q 表来产生动作, 即深度 Q 网络(DQN), 使得相近的状态可以得到相近的动作。DQN 有一个记忆库用于存储之前的经历。每次更新 DQN 时, 用随机抽取方法抽取一些之前的经历进行学习, 打乱了经历之间的相关性, 可以使神经网络的更新更加有效。

首先, 在 DQN 中构造两个结构相同但参数不同的神经网络, 一个用于预测 Q 估计值的主神经网络  $Q(s_t, a, \theta)$ , 它有最新的参数  $\theta$ , 其 Q 估计值为  $r_{t+1} + \lambda \max_{a \in A} Q(s_{t+1}, a, \theta)$ ; 另一个用于预测 Q 实际值的目标神经网络  $Q'(s_t, a, \theta')$ , 它使用的参数是一段以前时间的参数  $\theta'$ , 并保持一段时间不变, 其 Q 实际值为  $r_{t+1} + \lambda \max_{a \in A} Q(s_{t+1}, a, \theta')$ 。然后, 通过最小化损失

函数进行训练。损失函数为两个网络输出的差的平方, 参数更新使用随机梯度下降算法。

另外, 基本上 Q-learning 所有的目标 Q 值都是通过贪婪法直接得到的, 即在参数更新或值函数迭代过程中采用最大值操作, 这会导致过度估计。过度估计会影响到策略决策, 从而导致获取的策略不是最优, 而 DQN 算法无法解决 Q-learning 的过度估计问题。因此, 本文使用 DDQN, 通过解耦 Q-learning 中目标 Q 值动作的选择和目标 Q 值的计算来消除过度估计。则 DDQN 的主神经网络的 Q 估计值为:

$$y = r_{t+1} + \lambda \max_{a \in A} Q(s_{t+1}, \max_{a \in A} Q(s_{t+1}, a, \theta), \theta) \quad (23)$$

目标神经网络的 Q 实际值为:

$$y' = r_{t+1} + \lambda \max_{a \in A} Q(s_{t+1}, \max_{a \in A} Q(s_{t+1}, a, \theta), \theta') \quad (24)$$

损失函数为:

$$Loss(\theta) = (y' - y)^2 \quad (25)$$

#### 4.3 基于 DDQN 的计算卸载算法

在 VEC 中基于 DDQN 的计算卸载算法(见算法 1)的过程如下。首先, 初始化两个神经网络和经验存储器(算法 1 中的第 1 行)。其次, 在每一个片段中, DRL 代理人为获取主神经网络做如下操作: 根据式(8)计算车辆是否可服务矩阵, 得到车辆网络状态(算法 1 中的第 3 行)。接下来, 在每个时间片内, 利用  $\epsilon$ -贪心算法选择并执行动作  $a_t$ (算法 1 中的第 5-9 行): 利用 rand 函数随机生成探索新策略的概率, 当随机概率大于  $\epsilon$  时从动作集中随机选择一个动作; 否则, 选择对深度神经网络输入  $(s_t, a_t)$  后具有最大估计 Q 值的动作。在执行动作  $a_t$  后, 获取奖励值  $r_t$  和下一个状态对  $s_{t+1}$ (算法 1 中的第 10 行), 并将状态转换对  $(s_t, a_t, r_t, s_{t+1})$  存储在经验存储器  $D$  中(算法 1 中的第 11 行)。然后, DRL 代理人使用随机梯度下降法不断更新网络参数(算法 1 中的第 12-18 行): 取梯度下降算法的步长为  $K$ (算法 1 中的第 12 行), 从经验存储器中随机抽取  $U$  个样本(算法 1 中的第 13 行), 转换样本数据, 根据式(23)计算目标 Q 值  $y_t$ (算法 1 中的第 15 行), 通过计算损失函数对主深度神经网络的参数  $\theta$  进行更新(算法 1 中的第 16 行)。最后, 得到最佳策略(算法 1 中的第 19 行)。

##### 算法 1 VEC 中基于 DDQN 的计算卸载算法

1. 初始化 DNN  $Q, \theta, Q', \theta'$ , 经验存储器  $D$ ;
2. for each episode do
3. 根据式(8)计算车辆是否可服务矩阵  $\mathbf{mat}$ , 得到  $s_1$ ;
4. for  $t=1$  to  $T$  do
5. if  $\text{rand}() > \epsilon$  then
6.  $a_t = \text{rand}(a)$ ;
7. else
8.  $a_t = \arg \max_{a_t \in A} Q(s_t, a | \theta)$
9. end if
10.  $s_{t+1}, r_t \leftarrow \text{env}(a_t)$ ;
11. 存储  $(s_t, a_t, r_t, s_{t+1})$  到  $D$ ;
12. if  $t \equiv 0 \pmod K$  then
13. 从  $D$  随机抽取  $U$  个样本  $(s_j, a_j, r_j, s_{j+1})$ ;
14. 根据式(23)计算目标 Q 值  $y'$ ;
15. 根据式(24)计算损失函数以更新  $\theta$ ;
16. 更新  $Q'$  网络参数  $\theta'$ ;
17. end if

18. end for  
 19. 选择  $A \sim \pi_{\theta}(S_t)$ ;  
 20. end for

## 5 实验结果

### 5.1 参数设置

本文使用 Python 和 TensorFlow 进行实验,取一段长 400 m、宽 10 m 的单向高速公路,公路上有车速为  $[15, 30]$  m/s 的 30 辆车。车辆与 RSU 和车辆之间的最大通信距离分别为  $d_{range}^{V2I} = 200$  m,  $d_{range}^{V2V} = 15$  m。任意车辆产生任务的概率  $P \in [0, 1]$ ,每个任务的数据大小  $D_i \in [0, 1]$  MB,需要的计算资源  $C_i \in (1, 1000)$  Megacycles,最大容忍时间  $tmax_i \in (0, 1]$  s。在 MEC 服务器的计算能力为  $f^{MEC} = 10$  GHz,设备的计算能力  $f^i \in [0.5, 1]$  GHz,RSU 带宽  $B^{V2I} = 15$  MHz,车辆带宽  $B^{V2V} = 10$  MHz。车辆的发送功率  $P = 400$  mW, 高斯白噪声功率  $N_0 = -100$  dB。信道衰落因子  $h = 1$ ,路径损耗  $\delta = 4$ 。剩余在通信范围内时间的两个阈值  $t_1 = tmax_i$ ,  $t_2 = tmax_i/2$ 。剩余电量的两个阈值  $b_1 = 80\%$ ,  $b_2 = 20\%$ 。判断车辆能否作为候选服务车辆的阈值为  $\epsilon = 0.5$ 。折扣因子  $\lambda = 0.9$ ,贪心算法的探索速率  $\epsilon = 0.9$ ,经验存储器大小  $U = 3000$ ,总片段数为 300,时间片段数为  $T = 50$ ,批大小为 32,学习率  $\alpha = 0.01$ ,步长  $K = 100$ 。

### 5.2 结果分析

为了评估本文方案,我们比较了 3 种方案:1)任务在本地执行(Only Local),所有任务车辆都选择在本地执行任务;2)MEC 卸载(MEC Offloading),在本地计算任务或将计算任务卸载到 MEC 服务器;3)车辆协同的 MEC 卸载(the proposed scheme),任务可以在本地计算,也可以卸载到 MEC 服务器或者卸载到候选服务车辆。

图 2 给出了本文算法在取不同折扣因子时的收敛性。折扣因子  $\lambda$  是对未来奖励的衰减值,代表了当前奖励与未来奖励的重要性, $\lambda$  越大则代理人越重视历史经验,以前的经验对未来奖励的影响就越大。折扣因子越小,收敛速度就越快,因为  $\lambda$  越大,代理人需要往前考虑的步数越多,训练难度就越高。另外,当  $\lambda = 0.4$  和  $\lambda = 0.9$  时,奖励的差别不大,原因在于本实验中设定的任务时延较小,使得模型的跨时间性很弱,从而导致每一个动作对未来奖励的影响较小。

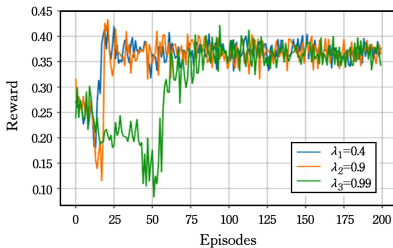


图 2 收敛结果

Fig. 2 Convergence results

图 3 给出了系统中车辆任务产生概率对本文方案的任务分配的影响情况。可以看出,在大部分情况下 V2I 卸载到 MEC 服务的任务和失败的任务占总任务的比重较高。这是因为在高速场景下,车辆移动速度过快导致 TaV 无法保证在

MEC 服务器或 CsV 的结果返回之前仍然在其通信范围内。并且,随着任务产生概率的增加,卸载到 MEC 服务器和其他车的占比不断下降,失败率不断增高。这是因为,系统中任务总量不断增多,而提供服务的 MEC 服务器资源不变,提供服务的候选服务车辆占比减少,这给 MEC 服务器和少量的候选服务车辆带来了巨大负担。

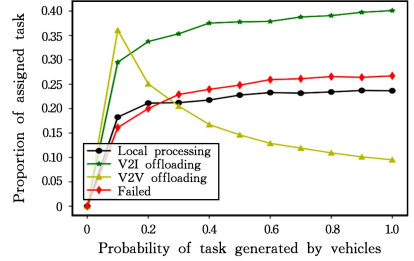


图 3 不同任务产生概率下任务的分配情况

Fig. 3 Impact of different task generation probabilities on task assignment

图 4 给出了任务最大容忍时间对本文方案的任务分配的影响情况。可以看出,随着最大容忍时间的增加,任务卸载失败率不断降低,这是因为时延要求的降低缓解了服务器和其他车辆的压力。选择本地计算和 V2V 卸载到临近车辆的比重在增加,这是因为尽管车辆计算资源较少,但是仍然能满足较高时延任务的卸载要求。

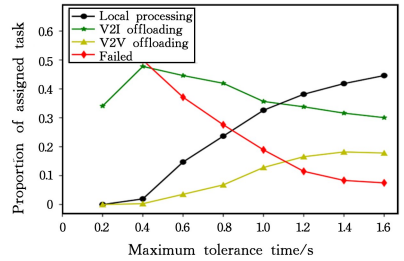


图 4 不同延迟约束下任务分配情况

Fig. 4 Impact of different tolerance time on task assignment

图 5 给出了车辆任务产生概率对不同方案平均时延的影响情况,可以看出本文提出的协同卸载的时延最小。随着任务产生概率的增加,MEC 卸载方案的时延较缓慢地下降,本文方案的平均时延有小幅度的上升。这是因为,对整个系统来说,MEC 服务器的可用计算资源是定值。而任务密度不断增大,导致 MEC 服务器的负载越来越重,而本文方案中的车辆之间可以互相服务,能有效降低任务时延。

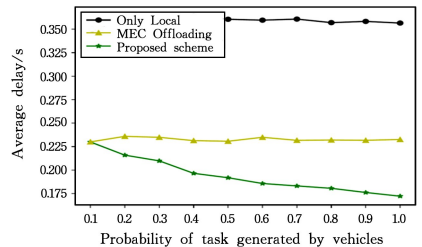


图 5 不同任务产生概率下多个方案的平均延迟变化

Fig. 5 Average delay of multiple schemes under different task generation probabilities

图 6 给出了车辆任务产生概率对不同方案任务失败率的影响情况。可以看出,随着任务产生概率的增加,3 种方案的任务处理的失败率有所增长,因为整个系统计算资源总量有限,而任务密度却在增加。与 MEC 卸载方案相比,本文方案的失败率增长得较快,这是因为在任务产生率低时,空闲资源的车辆占比较大,系统中更多的车辆可以成为候选服务车辆,进而能有效减小 MEC 任务的卸载失败率。

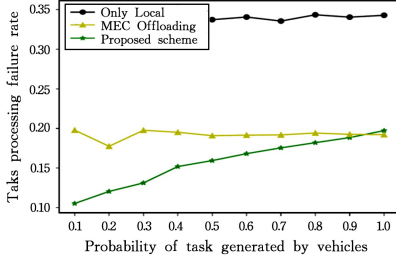


图 6 不同任务产生概率下多个方案的任务失败率变化情况  
Fig. 6 Failure rate of multiple schemes under different task generation probabilities

图 7 给出了不同容忍时间对不同方案失败率的影响。可以看出,随着任务容忍时间的增加,3 种方案的失败率都在降低。尽管这 3 种方案在失败率上的效果相近,但是仍可以看出本文方案在大部分时延范围内比其他两个方案好,进一步说明了本文方案的有效性。

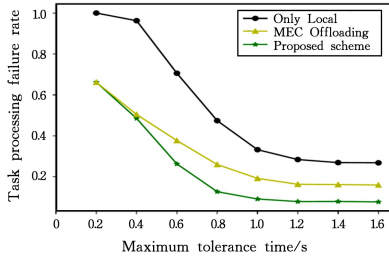


图 7 不同容忍时间对不同方案失败率的影响  
Fig. 7 Impact of different tolerance time on failure rate of different schemes

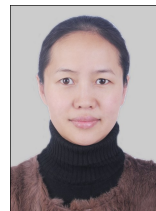
**结束语** 本文提出了一种基于深度强化学习的车辆协同的计算卸载方案,该方案可以缓解高速公路场景下车辆高速移动与 MEC 服务器静态部署的矛盾。本文首先研究了车辆移动性、剩余电量、可用资源等会影响车辆对外提供服务的因素;然后,通过将时延容忍约束下的任务时延最小化问题建模为马尔可夫决策过程,采用强化学习方法进行问题求解;最后,设计并实现了基于 DDQN 的协同卸载算法,以适应动态变化的高速公路环境。仿真结果证明了所提出的计算卸载方案的可行性和可靠性。在未来的工作中,我们将研究车辆协同的计算卸载方案中的激励机制。

## 参考文献

- [1] ETSI. MEC in 5G networks [OL]. [https://www.etsi.org/images/files/ETSIWhitePapers/etsi\\_wp28\\_mec\\_in\\_5G\\_FINAL.pdf](https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf).
- [2] YUAN S, FAN Y, CAI Y. A survey on computation offloading

for vehicular edge computing[C]// Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City (ICIT 2019). New York: ACM Press, 2019: 107-112.

- [3] YU P, ZHANG J, LI W, et al. Energy-efficient resource allocation method in mobile edge network based on double deep Q-learning [J]. Journal on Communications, 2020, 41 (12): 148-161.
- [4] MEHDI M, ALA A F, SAMEH S, et al. Deep Learning for IoT Big Data and Streaming Analytics; A Survey [J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2923-296.
- [5] NING Z L, FENG Y F, COLLOTTA M, et al. Deep learning in edge of vehicles: Exploring relationship for data transmission [J]. IEEE Transactions on Industrial Informatics, 2019, 15(10): 5737-5746.
- [6] ZHANG J, GUO H, LIU H, et al. Task offloading in vehicular edge computing networks: A load-balancing solution [J]. IEEE Transactions on Vehicular Technology, 2020, 69(2): 2092-2104.
- [7] KHAN I, TAO X, RAHMAN G M S, et al. Advanced Energy-Efficient Computation Offloading Using Deep Reinforcement Learning in MTC Edge Computing [J]. IEEE Access, 2020, 8: 82867-82875.
- [8] HUANG X, YU R, KANG J, et al. Exploring mobile edge computing for 5g-enabled software defined vehicular networks [J]. IEEE Wireless Commun, 2017, 24(6): 55-63.
- [9] QIAO G, LENG S, ZHANG K, et al. Collaborative Task Offloading in Vehicular Edge Multi-Access Networks [J]. IEEE Communications Magazine, 2018, 56(8): 48-54.
- [10] DAI S, WANG H L, GAO Z, et al. An adaptive computation offloading mechanism for mobile health applications [J]. IEEE Transactions on Vehicular Technology, 2020, 69(1): 998-1007.
- [11] HUANG X, YU R, LIU J, et al. Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications [J]. IEEE Access, 2018, 6: 66649-66663.
- [12] LI Y, YANG B, CHEN Z, et al. A Contract-Stackelberg Offloading Incentive Mechanism for Vehicular Parked-Edge Computing Networks[C]// Proceedings of 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring). Piscataway: IEEE Press, 2019: 1-5.
- [13] LIU J, WANG S, WANG J, et al. A task-oriented computation offloading algorithm for intelligent vehicle network with mobile edge computing [J]. IEEE Access, 2019, 7: 180491-180502.
- [14] NING Z, DONG P, WANG X, et al. When Deep Reinforcement Learning Meets 5G Vehicular Networks: A Distributed Offloading Framework for Traffic Big Data [J]. IEEE Transactions on Industrial Informatics, 2019, 16(2): 1352-1361.



**FAN Yan-fang**, born in 1979, Ph.D, is a member of China Computer Federation. Her main research interests include information security, vehicular networking and edge computing.