



# 计算机科学

COMPUTER SCIENCE

## 车载边缘计算网络中基于MAB的动态任务卸载方案研究

薛建彬, 王海牛, 关向瑞, 郁柏文

引用本文

薛建彬, 王海牛, 关向瑞, 郁柏文. 车载边缘计算网络中基于MAB的动态任务卸载方案研究[J]. 计算机科学, 2023, 50(11A): 230200186-9.

XUE Jianbin, WANG Hainiu, GUAN Xiangrui, YU Bowen. Study on Dynamic Task Offloading Scheme Based on MAB in Vehicular Edge Computing Network [J]. Computer Science, 2023, 50(11A): 230200186-9.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### [自动化红队测试中强化学习策略的实现与验证](#)

Implementation and Verification of Reinforcement Learning Strategy in Automated Red Teaming Testing

计算机科学, 2023, 50(11A): 230200162-6. <https://doi.org/10.11896/jsjcx.230200162>

### [基于SA-UCB算法的Android应用程序自动化测试方法](#)

Automated Testing Method of Android Applications Based on SA-UCB Algorithm

计算机科学, 2023, 50(11A): 221200145-7. <https://doi.org/10.11896/jsjcx.221200145>

### [应急通信场景下基于JTORATPAIA的NOMA-MEC系统研究](#)

Study on NOMA-MEC System Based on JTORATPAIA in Emergency Communication Scenarios

计算机科学, 2023, 50(11A): 221000240-8. <https://doi.org/10.11896/jsjcx.221000240>

### [用户公平保障的边缘服务缓存与任务卸载算法](#)

Fairness-aware Service Caching and Task Offloading with Cooperative Mobile Edge Computing

计算机科学, 2023, 50(11A): 230200095-8. <https://doi.org/10.11896/jsjcx.230200095>

### [毫米波MU-MIMO系统中自适应混合预编码器的设计](#)

Design of Adaptive Hybrid Precoder in mmWave MU-MIMO Systems

计算机科学, 2023, 50(11A): 221200047-5. <https://doi.org/10.11896/jsjcx.221200047>

# 车载边缘计算网络中基于 MAB 的动态任务卸载方案研究

薛建彬 王海牛 关向瑞 郁柏文

兰州理工大学 兰州 730050

(volvoxuejb@126.com)

**摘要** 将移动边缘计算技术应用到车载网络所形成的车载边缘计算系统,能够通过任务卸载为其他移动设备提供计算服务。然而,由于车载设备的移动性,导致了车载任务卸载环境是动态变化和不确定性的,具有快速变化的网络拓扑、无线信道状态和计算负载,这些不确定性让任务卸载过程非理想化。针对这些不确定性,考虑将 MEC 服务器的计算资源下沉到车载设备,研究车辆之间的任务卸载,并提出了一种解决方案,使得车辆能够在未知状态信息的前提下学习周围车辆的服务性能并卸载任务。基于多臂老虎机框架,设计了一种二阶探索的强化学习算法,以最大化用户平均卸载回报,并且在在一个卸载阶段结束后提出了一种服务集更新方式,以保证用户的服务质量。仿真结果表明,与现有的基于置信上限的算法相比,所提方案下的卸载回报提高了约 34%。

**关键词:** 6G; 车载边缘计算; 多臂老虎机; 任务卸载; 强化学习

**中图分类号** TN929.5

## Study on Dynamic Task Offloading Scheme Based on MAB in Vehicular Edge Computing Network

XUE Jianbin, WANG Hainiu, GUAN Xiangrui and YU Bowen

Lanzhou University of Technology, Lanzhou 730050, China

**Abstract** The mobile edge computing system formed by applying mobile edge computing technology to Internet of Vehicles can provide computing services for other mobile devices through task offloading. However, due to the mobility of vehicle equipment, the environment of vehicular task offloading is dynamic and uncertain, with rapidly changing network topology, wireless channel state and computing load. These uncertainties make the task offloading process non-idealized. In view of these uncertainties, the computing resources of the MEC server are sunk into the vehicle equipment to study the task offloading between vehicles, and a solution is proposed to enable vehicles to learn the service performance of surrounding vehicles and offload tasks without knowing the status information. Based on the multi-arm bandits framework, a second-order exploration reinforcement learning algorithm is designed to maximize the average offloading return of users, and a service set update method is proposed after the end of an offloading phase to ensure the quality of service for users. Simulation results show that, compared with the existing algorithm based on upper confidence bound, the offloading return under the proposed scheme is improved by about 34%.

**Keywords** 6th generation mobile communication technology, Vehicular edge computing, Multi-armed bandit, Task offloading, Reinforcement learning

## 1 引言

第六代移动网络(6G)有望通过支持天空地、海事和水下通信的异构网络来实现全连接世界<sup>[1]</sup>。近年来,物联网(IoT)和无线技术的发展使得车联网领域不断进步,智能汽车也被称为“带轮子的计算机”<sup>[2]</sup>,一些交通认知,自动驾驶和增强现实等前沿应用正在兴起,据 IHS Markit 预测,到 2025 年,道路上的车辆和车载设备的数量将达到近 20 亿,每辆车每天可以产生高达 30 TB 的数据<sup>[3]</sup>。这些新型未来车载应用带来的关键挑战之一是需要大量的计算资源和通信资源来处理其产生的高额、异构的数据量,以保证用户服务质量(QoS),而对车辆上有限的计算资源是一个重大挑战。从车辆自身的角度考虑,更新硬件设备是最简单且直接的方法,但会造成很高

的经济成本。因此,需要去探索更加高效的数据处理方式。

作为分布式计算应用的先例,云计算最先被提出,用于处理这些高额数据量。消耗资源的任务被卸载到强大的云中心,该中心执行所有必要的计算并将结果返回给移动终端<sup>[4]</sup>,然而云服务器的部署位置距离移动设备很远,两者之间的通信由于回程链路的影响会导致较高的卸载时延和抖动性能,并且随着用户数量和服务类型的快速增加,接入云平台的海量设备的数据聚合必然会造成网络负载过大<sup>[5]</sup>,为此引入了新型的计算模式移动边缘计算(MEC),即在网络边缘提供信息服务环境和云计算能力。将云计算扩展到网络边缘,MEC 服务器可以在网络边缘对物理距离更近的移动设备提供计算、存储等服务,缓解了大数据量卸载而导致的主干网络拥塞以及较高的回程时延等问题。

基金项目:甘肃省科技计划(23YFGA0062);甘肃省创新基金(2022A-215)

This work was supported by the Gansu Science and Technology Program(23YFGA0062) and Innovation Fund of Gansu Province(2022A-215).

通信作者:王海牛(wanghn0808@163.com)

将 MEC 与车载网络相结合是一种有前途的计算范式,一方面促进了 V2I 与 V2X 的发展,另一方面,车载设备所产生的与日俱增的多样化任务伴随着不计其数的数据量,这些数据量与车载设备的处理能力已不能相匹配,而 MEC 的提出正好缓解了这一关键问题,借助边缘计算,通信、存储和计算能力更加贴近终端用户<sup>[6]</sup>。通过将 MEC 服务器与路边单元(RSU)等基础设施联合部署形成共享资源<sup>[7]</sup>,进而形成了车辆边缘计算系统。车载边缘计算(VEC)能够向车载设备提供云计算服务,从而达到缓解海量数据压力的目的。

然而,不同于手机、平板电脑等移动终端,在车联网系统中由于运行更前沿的应用,车载设备会产生巨额数据量,配备有 MEC 服务器的 RSU 没有足够的计算资源去处理这些数据量,扩充 MEC 服务器的计算能力或者引入虚拟化技术,利用停放车辆资源来获得更多的计算资源也无法彻底解决高数据量问题。并且由于车辆的高机动性,网络拓扑和可用的计算资源变化很快,将较高的数据量通过有限的信道容量卸载至服务器,一方面会产生较高的传输时延,另一方面车载设备需要卸载的任务可能无法完全上传至 MEC 服务器,虽然有不少工作研究切换问题,但服务稳定性的问题亟待解决。

本文着重考虑利用车辆计算资源来进行任务卸载,将配备有 MEC 服务器的 RSU 中的计算资源分流至车载设备层中,实现资源“下沉”。这一方面缓解了 MEC 服务器的工作负担,另一方面,车载设备或者乘客可以利用附近有空闲计算资源的车辆来进行任务卸载。将需要卸载任务的车辆称为用户车辆(UVs),将提供计算服务的车辆称为协助计算车辆(ACVs)。每台用户车辆的卸载决策是独立的。

由于车载设备的高机动性,网络拓扑和无线信道变化剧烈,在实际场景考虑中无法忽视这种变化。每台用户车辆都需要获取周围车辆的具体信息,如服务车辆的可用计算能力、信道环境等状态信息,想要实时获取这些准确的状态信息,就必须不断和周围车辆进行信令交互,这种方式会累积大量的冗余信息,降低任务传输速率,因此用户无法事先获取这些信息。

在未知状态信息的情况下,引入强化学习理论让用户进行卸载决策,并在卸载任务的同时学习周围车辆的服务性能。在设计任务卸载算法时采取了传统的多臂老虎机(MAB)框架<sup>[8]</sup>。传统的 MAB 问题旨在进行探索和利用之间的权衡,玩家会探索动作集中的每一个选项,这些动作对应着不同的奖励分布,接着玩家利用学习到的信息去综合考虑探索(选择可能最优的动作)和利用(选择当下最优的动作)来选择经验上最优的动作。基于 MAB 框架提出了很多学习算法,如文献[9]提出的  $\epsilon$ -greedy、基于置信上限(UCB)算法的 UCB1, UCB2, UCB1-normal, 这些算法已应用在无线通信场景来学习未知环境。

然而,由于车载设备的高机动性,无线信道环境是时变的,并且协助计算车辆的计算能力也是随机的,这使得卸载成本更高,且随着时间的推移,用户对环境的学习效率越来越差。现有的一些基于 MAB 的方案并没有解决上述问题,需要研究更契合上述场景的新型 MAB 框架。本文的贡献主要包括:

1) 基于 MAB 理论,提出了一种基于 UCB 的二阶探索强化学习算法,该算法由用户执行,并为用户提供卸载决策以

最大化用户的平均卸载回报。该算法能够独立工作,不需要与 RSU 进行额外信令交互,并且能够在进行任务卸载时学习协助计算车辆的服务性能。该算法的计算复杂度低,不需要同其他车辆进行信息交互(如计算能力、无线信道状态等),易于在实际车联网中实现。

2) 考虑了协助计算车辆和用户车辆的高移动性,以及复杂的无线环境,提出了一种新的服务集更新方式,该方案进一步提高了用户服务质量。

3) 进行了大量的仿真实验,并且对比了不同算法下的延迟性能,仿真结果表明,本文算法能够使用户实现更低的时延与计算能耗。

## 2 相关工作

已有较多研究将 MEC 服务器部署在 RSU,为车载设备提供计算服务<sup>[10-11]</sup>,例如文献[12]提出了一种多路径传输工作负载均衡优化方案,该方案以边缘计算架构中的多路径传输为传输协议,支持车辆与边缘节点之间的通信,解决了基于距离的负载分配方法所引起的边缘节点过载问题。文献[10]为了缩短车辆用户的卸载延时,提出了一种多平台智能卸载和资源分配算法,该算法能够动态组织计算资源,进一步提高车载网络性能。文献[11]考虑了任务到达的随机性,将其公式化为车辆到达和任务生成的泊松过程,提出了一种负载感知的卸载方案,该方案进一步提高了车载网络性能。然而,以上方案并没有考虑车辆的移动性所引起的实际卸载效率低的问题,为了使得车辆用户能够在 RSU 覆盖范围内完成任务卸载,一些文献引入了多跳网络,以更高效率的传输路径向 RSU 上传卸载任务<sup>[13-14]</sup>,例如文献[13]利用多跳车载自组织网络(VANETs)来辅助车辆的计算卸载,建立了多跳路由路径的可靠性模型,通过二分搜索法执行具有较低中继和计算成本的卸载策略。为了进一步提高多跳自组织网络的卸载性能,文献[14]提出了一种 k-hop-limited 车载自组织网络,该网络能够判断一台用户车辆能否在 k 跳的限制范围内找到符合要求的传输路径,并且在有限的时间间隔内找到更优的路径,该方案仅考虑数据卸载路径的生命周期,并不考虑未来存在的候选 V2V2I 卸载路径。

尽管多跳的方式很好解决了 RSU 覆盖以及切换的问题,但是由于 MEC 服务器资源的有限性,用户可能分配到较少的计算资源,甚至进入较长的排队等待状态,这种情况仍然导致较高的时延。将 MEC 层的计算资源下沉到服务车辆层是一种可行方案,即用户车辆将产生的任务卸载到附近拥有空闲资源的服务车辆,以提高卸载性能。一部分文献研究了将车辆作为边缘服务器,以提供计算服务,如文献[15]提出了一种车辆辅助 MEC 的范例,该模式利用 Stackelberg 博弈理论建立动态激励模型,鼓励拥有空闲计算资源的车辆为各种类别的移动设备提供计算服务,与此同时文献[16]提出了将移动边缘计算技术引入到车载自组网中,构建了一个车载边缘计算系统,利用道路上车辆的计算资源提供广泛可靠的服务。同时提出了一种多目标任务调度算法,该算法能够观察 VEC 系统的状态,缩短了任务执行时间。为了进一步提高资源利用率,有一些工作利用了停放车辆的空闲计算资源来提高车载应用服务的性能,如文献[17]提出了车辆边缘计算和网络中的停放车辆(PV)辅助,以利用来自 PV 的未充分利用的

计算资源来增强边缘车辆网络的资源容量,还提出随机理论为用户制定卸载决策,通过用户满意度和工作量承担来设计效用函数,采用双边匹配游戏来公平地考虑每个请求者和 PV 的效用最大化。文献[18]将 PV 与智能停车相结合,提出了一种 VFC 感知的停车预约拍卖,引入了经济激励机制刺激停放车辆提供计算资源,进一步实现了资源下沉。文献[19]引入了一个卸载服务提供商,它部署 MEC 服务器并按照用户需求调度 PV 以处理卸载任务。研究有效的工作负载分配和以用户为中心的效用最大化,以优化网络范围的任务调度。必要时,卸载服务提供商代表卸载用户为 PV 设计基于合约的激励机制。以上工作有效地实现了计算资源下沉,缓解了 MEC 服务器的计算压力,然而由于车辆用户的高机动性,静态的停放车辆无法满足移动车辆产生任务的低时延需求。此时考虑调动移动中车辆的空闲计算资源是一项有前途的方案。

为了调动移动车辆的计算资源,文献[20]考虑了车辆的移动性,网络拓扑和可用资源变化很快,提出了一种在车载边缘计算(RVEC)中的中继任务卸载方案。该方案联合考虑 VEC 服务器与车载设备计算能力,任务在被传输到 VEC 服务器的同时被卸载在车对车中继中进行计算,这样的计算方式仍然以 VEC 服务器为主,对周围空闲车辆计算资源的利用程度较低。文献[21]认为车辆之间需要高效协作来管理具有低延迟的计算任务,所提方案使用一些车辆作为边缘节点,由这些边缘节点负责寻找任务处理器节点,所提模型使用基于模糊逻辑的方法通过综合考虑车辆计算能力、车辆移动性和车辆分布来形成边缘架构,通过修改模糊隶属函数和模糊规则来调整该方案,以适用于不同的场景。该方案在任务处理上有较好性能,但是二阶段的卸载方式的模型复杂度较高,仍无法满足低时延要求。为了降低模型复杂度,文献[22]提出了一种虚拟边缘,它由附近具有可用计算资源的车辆组成,该虚拟边缘的形成综合考虑了其成员之间的陪伴时间与计算能力,从而能够预测虚拟边缘的持续时间。该方案能够以用户为中心形成虚拟计算资源,并且 MEC 服务器没有参与计算服务,实现了资源下沉。文献[23]在此基础上考虑了链路可靠性、距离、相对速度等重要指标,提出了一种高效的分布式任务卸载方案,并将任务在卸载前分成若干子任务,以进一步提高服务性能。

尽管上述工作考虑了移动车辆的计算资源,但是这些文献都没有考虑服务的持续性问题,即一个阶段结束之后,每一次卸载过程都需要通过算法来制定卸载决策,并且组合优化问题随着优化变量数目的增多而引起更高的算法复杂度。为了解决上述问题,一些文献引入了人工智能算法,文献[24]将 VEC 的任务卸载时间和卸载对象归结为有限时域马尔可夫决策过程(MDP),以优化通信、计算、切换和迁移的延迟。该模型考虑了随机驾驶员行为、不准确的样本数据和复杂的路径环境,在未知转移概率的情况下,提出了一种时间感知算法,该算法在马尔可夫决策过程中引入时延成本,以此来调整状态转移概率,从而获得更好的任务卸载性能。由于 MDP 的算法具有较高的算法复杂度,并且比较依赖状态转移概率,一些文献基于 MAB 理论<sup>[25]</sup>提出了性能更好的强化学习算法,如文献[26]基于 MAB 框架提出了一种可变探索因子的 UCB 置信算法,不同于传统的 UCB 算法,该算法能够感知

任务输入数据大小,并且按照不同的任务数据进行探索和利用之间的权衡,实现了较低的悔恨系数。文献[27]在文献[26]的基础上结合车联网,设计了一种更新算法,该算法不仅能够感知输入数据的大小,还能够进行事件感知,即在每一个阶段进行完成后将最新的阶段信息考虑到探索和利用之间的权衡,该算法相比<sup>[26]</sup>在性能上有进一步的提升,同时具有更低的悔恨系数,然而这两种因素的考虑在算法上具有耦合性,并且对于环境改变的感知能力较弱。

### 3 系统模型

如图 1 所示,本文以搭载 MEC 服务器的车载计算系统为研究场景。假设该场景由 1 台用户车辆,1 个 RSU 和  $N$  台配备有 MEC 服务器的协助计算车辆组成,其中用户车辆和协助计算车辆通过 V2V 链路通信,用户和 RSU 通过 V2I 链路通信,每当新的协助计算车辆进入系统时,便在 RSU 中申请记录,用户通过与 RSU 交互获取卸载权限。

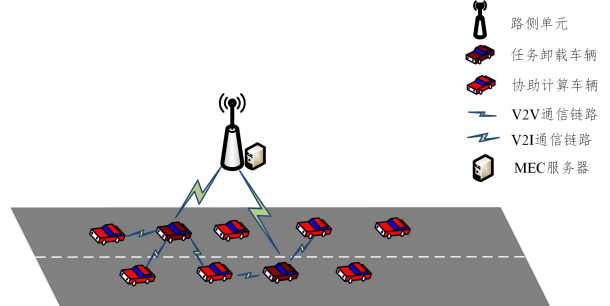


图 1 RSU 辅助下车载边缘计算网络模型

Fig. 1 Vehicular edge computing network model assisted by RSU

#### 3.1 服务车辆集合建立

需要请求任务卸载的用户车辆会在附近周围搜索可用空闲资源车辆,以形成服务车辆集合,附近的车辆会收到用户车辆的请求并对该车辆提供任务计算服务,该服务集合中的车辆满足良好的信道条件与连接稳定性。假设系统中有一个用户车辆与  $N$  台服务车辆所构成的车辆集  $N = \{ACV_1, ACV_2, \dots, ACV_N\}$ 。当用户车辆的任务计算完成之后,将会清空服务集列表,释放与服务集车辆的连接。

#### 3.2 通信模型

当服务集确定之后,用户进行任务上传过程,由于车载任务具有较高数据量的特性,用户不会一次性上传整个任务,而是需要将其分割。本文假定任务能够分割成  $M$  个子任务,其中第  $i$  个子任务  $subtask_i$  由一个二元组  $\{D_i, C_i\}$  构成,其中  $i \in \mathcal{M}, \mathcal{M} = (1, 2, \dots, M)$ , 且  $D_i$  表示任务大小,单位为 bit,  $C_i$  为计算该任务所需要的 CPU 周期数。由于任务的异构性,  $D_i$  与  $C_i$  之间没有相关性,每一个阶段用户都会卸载子任务,第  $i$  个子任务卸载到服务车辆  $ACV_n^i \in \mathcal{N}$ , 直至卸载完所有的子任务,考虑到实际中的路径损耗和随机性效应,用户和协助计算车辆之间的通信信道被建模为同分布频率平坦块衰落信道<sup>[28]</sup>。将  $ACV_n^i$  和用户之间的小尺度衰落信道功率增益表示为  $h_n^i$ 。假设有一个有界均值  $\mathbb{E}[h_n^i] = \bar{h}_n < \infty$ , 则在第  $i$  个子任务进行卸载时,  $ACV_n^i$  和用户之间的信道功率增益可以表示为:

$$H_n^i = h_n^i k_0 \|d_n^i\|^{-\theta} \quad (1)$$

其中,  $\theta$  表示路径损耗指数,  $k_0$  是一个常系数, 与  $(\lambda/4\pi)^2$  成

比例,  $d_n^i$  为用户与  $ACV_n^i$  之间的距离

因此定义子任务  $subtask_i$  的上传速率为:

$$R_n^i = B \log_2 \left( 1 + \frac{Ph_n^i}{\sigma^2 + I} \right) \quad (2)$$

综合式(1)与式(2),可以得到数据速率为:

$$R_n^i = B \log_2 \left( 1 + \frac{Ph_n^i k_0 \|d_n^i\|^{-\theta}}{\sigma^2 + I} \right) \quad (3)$$

其中,  $B$  为车载通信网络系统为用户分配的信道带宽,用于上传任务与信令信息等,  $P$  为用户传输功率,因此得到子任务  $subtask_i$  的上传时间为:

$$T_{i,n}^{up} = \frac{D_i}{R_n^i} \quad (4)$$

则用户向  $SV_n^i$  上传任务时的传输能耗为:

$$E_{i,n} = P T_{i,n}^{up} \quad (5)$$

### 3.3 计算模型

当接收到来自用户上传的任务之后,协助计算车辆会为该任务分配可用的计算资源。由于车载设备状态的未知性,车辆的计算能力无法事先获取,并且有可能同时服务多台用户车辆,服务车辆可以利用 DVFS 技术动态地调节 CPU 计算能力<sup>[29]</sup>,假设  $ACV_n$  为  $subtask_i$  提供的计算资源为  $f_n^i$ ,则对应的任务计算时延为:

$$T_{i,n}^{comp} = \frac{C_i}{f_n^i} \quad (6)$$

### 3.4 结果回传过程

当协助计算车辆完成子任务的计算之后,会将计算结果回传给用户,在最后一台服务车辆完成子任务计算并回传计算结果之后,完成总的卸载过程。由于计算结果的数据量非常小,并且下一阶段子任务的卸载过程往往伴随着本阶段子任务的结果回传过程,因此忽略掉结果回传所带来的影响。

## 4 问题公式化

为了较好评价协助计算车辆的服务性能,定义一种新的卸载效用函数。

$$\zeta_{i,n} \stackrel{\text{def}}{=} \alpha T_{i,n} + \beta E_{i,n} \quad (7)$$

该效用函数表示  $subtask_i$  在完成传输与计算,且用户收到来自协助计算车辆的计算结果之后,对本次卸载过程在时延与能耗端做出综合评价。其中  $T_{i,n} = T_{i,n}^{up} + T_{i,n}^{comp}$ ,表示时延分量,  $E_i$  为能耗分量,  $\alpha$  与  $\beta$  分别为做了归一化处理后的时延权重与能耗权重,满足  $\alpha, \beta \in (0, 1)$  且  $\alpha + \beta = 1$ ,当任务的时延敏感程度较高时,  $\alpha$  的值较大,反之亦然。

设计的目标函数是要通过调整用户车辆的卸载决策,来最小化平均卸载效用。因此任务卸载问题被公式化为:

$$P1: \min_{a_1, \dots, a_M} \frac{1}{M} \sum_{i=1}^M \zeta_i(i, a_i) \quad (8)$$

在问题 P1 中,  $a_i$  为优化变量,表示用户在卸载第  $i$  个子任务时选择协助计算车辆的下标,且  $a_i \in N$ 。

#### 4.1 已知状态信息下的决策分析

与延迟和能耗相关的状态信息由两类参数组成:第一类参数为每个子任务的参数,包括输入数据量  $D_i$  与计算强度  $C_i$ ,这类参数由用户车辆生成任务时掌握,第二类参数与协助计算车辆相关,分别为上行链路传输速率  $R_n^i$  与为用户分配的计算资源,其中影响上行链路传输速率的因素包含系统分配的信道带宽、信道增益、噪声以及无线干扰。如果用户在卸

载每一个子任务之前都能准确地获得上述状态信息,则卸载效用将达到最优。此时卸载决策变量可以表达为:

$$a_i = \arg \min_{i \in \mathcal{M}} \zeta_i(i, n) \quad (9)$$

然而,由于车辆的移动性,传输速率变化很快,实时信道状态信息很难获取。考虑用户之间进行协作会导致较高的信令开销,同时协助计算车辆的计算负载会动态变化,使其分配的 CPU 频率随时间变化而变化。此外,用户和协助计算车辆之间交换状态信息同样会导致很高的信令开销,具体在下一节中分析。因此,用户事先并不知道协助计算车辆的状态信息,并且在进行卸载时不能意识到能够提供较低能效的车辆。

#### 4.2 未知状态信息下的决策分析

为了解决不可获取状态信息的问题,引入强化学习机制,考虑将用户车辆看成一个智能体,即能够在卸载任务的同时学习协助计算车辆的服务性能。具体来说,用户会根据历史能效观测值选择协助计算车辆,而无需获取精确的传输速率和 CPU 频率。因此设计了一种强化学习算法,使用户卸载能效的期望最小化,问题 P1 可以转化为:

$$P2: \min_{a_1, \dots, a_M} \frac{1}{M} \mathbb{E} \left[ \sum_{i=1}^M \zeta_i(i, a_i) \right] \quad (10)$$

其中,  $\mathbb{E}[\cdot]$  表示期望,将式(7)展开,可以得到:

$$\zeta_i = \alpha (T_{i,n}^{up} + T_{i,n}^{comp}) + \beta P T_{i,n}^{up} = (\alpha + \beta P) T_{i,n}^{up} + \alpha T_{i,n}^{comp} \quad (11)$$

假设车载设备的发射功率  $P$  是恒定的,则  $\alpha + \beta P$  是一个常数。因此对式(11)右边除以常数  $\alpha + \beta P$ ,可以得到:

$$\zeta_i = T_{i,n}^{up} + \frac{\alpha}{\alpha + \beta P} T_{i,n}^{comp} \quad (12)$$

因此,可以将 P2 转化为 P3。

$$P3: \min_{a_1, \dots, a_M} \frac{1}{M} \mathbb{E} \left[ \sum_{i=1}^M \left( T_{i,n}^{up} + \frac{\alpha}{\alpha + \beta P} T_{i,n}^{comp} \right) \right] \quad (13)$$

通过分析问题 P3 可以得出当用户发射功率  $P$  大于 0 时,  $\frac{\alpha}{\alpha + \beta P}$  的取值范围在区间  $(0, 1)$  之内,也就是说,当考虑用户端的发送能耗时,协助计算车辆的计算时延权重会降低,这反映了当信道条件较差时,应该更多地考虑用户的上传时延。当权重因子  $\beta$  的值为 0 时,用户能效就成了  $T_{i,n}^{up} + T_{i,n}^{comp}$ 。

为了使得预期的回报更加直观,并且便于对用户回报进行归一化,设计了一个新的函数将卸载效用进行进一步转化,得到卸载回报。

$$U_{i,n} = U_{i,n}^{up} + U_{i,n}^{comp} = \frac{k^{up}}{T_{i,n}^{up}} + \frac{k^{comp}(\alpha + \beta P)}{\alpha T_{i,n}^{comp}} \quad (14)$$

其中,  $U_{i,n}^{up}$  与  $U_{i,n}^{comp}$  分别为传输回报与计算回报,  $k^{up}$  与  $k^{comp}$  分别为上传时延与计算时延的回报系数。因此目标优化函数转化为用户卸载回报总和,表达为问题 P4。

$$P4: \max_{a_1, \dots, a_M} \frac{1}{M} \mathbb{E} \left[ \sum_{i=1}^M \left( \frac{k^{up}}{T_{i,n}^{up}} + \frac{k^{comp}(\alpha + \beta P)}{\alpha T_{i,n}^{comp}} \right) \right] \quad (15)$$

与 P3 不同的是,在 P4 中,可以同时上传时延与计算时延灵敏反应,即当上传时延与计算时延其中一个发生变动时,都会较大程度影响卸载回报。

## 5 问题描述与解决

本节基于 MAB 框架,设计了一种任务卸载算法,该算法能够使得用户在未知周围信道环境的前提下,通过学习服务车辆的计算性能与信道状态信息,最大化用户的卸载回报。

#### 5.1 基于 MAB 的任务卸载算法

设计的算法将用户作为智能体,即决策者,将附近的协助

计算车辆视为“臂”,用户进行任务卸载,对应于选择一个“臂”进行推拉来获得奖励,协助计算车辆的计算能力与信道状态将会决定用户选择此服务车辆的卸载回报。每当进行一次任务卸载时,用户将会学习该车辆的计算性能,用于在下一阶段进行卸载时对服务集中的车辆进行评估,进一步优化用户车辆的卸载回报。现有关于 MAB 的算法并不能适用于该模型,因为该模型中的服务集的状态是随时间变化的,即在不同的子任务卸载阶段,“臂”会发生变化。用户车辆的高移动性所引起的信道环境变化将在该算法中解决。

算法 1 考虑了卸载回报变化对用户学习协助计算车辆服务性能的影响,不同于以往的工作,该算法将给予信道环境较差或服务能力不好的协助计算车辆一个更低的探索系数,使得用户以更高效率去寻找最优卸载车辆,降低学习悔恨。其中,  $\Delta_{i,n}$  为  $ACV_i^n$  的二阶平均收益差值,  $\epsilon$  为控制因子,是一个正系数,控制  $\Delta_{i,n}$  在探索项中的影响程度。

**算法 1** 基于 MAB 理论的二阶探索任务卸载算法

输入:  $D_i, C_i, P$

输出: 卸载决策  $b_{i+1}$

1. 对于所有的  $i=1, 2, \dots, M$
2. if  $i=1$
3. 用户搜索周围可提供计算资源的车辆,建立车辆服务集  $\mathcal{N}$ , 并且将初始子任务发送到所有协助计算车辆以获取其初始卸载回报  $U_{1,n}$
4. else:
5. 更新所有协助计算车辆的卸载回报  $\hat{U}_{i+1,n}$ :
 
$$\hat{U}_{i+1,n} = \bar{U}_{i,n} + \sqrt{\frac{\epsilon \Delta_{i,n} \beta \ln i}{n_{i,n}}} \quad (n=1, \dots, N) \quad (16)$$
6. 用户根据更新后的卸载回报选择最优协助计算车辆进行任务卸载
 
$$b_{i+1} = \arg \max_{n \in N} \hat{U}_{i+1,n} \quad (17)$$
7. 计算结果返回用户,观察所获得的  $U_{i+1,n}$
8. 更新  $\bar{U}_{i+1,n} \leftarrow \frac{\bar{U}_{i,n} n_{i,n} + U_{i+1,n}}{n_{i,n} + 1}$
9. 更新  $n_{i+1,n} \leftarrow n_{i,n} + 1$
10. 更新  $\Delta_{i+1,n} = U_{i+1,n} - U_{i,n}$

在算法 1 中,第 1 行为数据输入阶段,第 2-4 行为初始车辆服务集阶段,第 6-11 行为任务卸载阶段。

## 5.2 复杂度分析

在算法 1 中,卸载回报更新的复杂度为  $O(N)$ ,其中  $N$  为每一个阶段服务集的车辆数。第 7 行进行最优服务车辆决策,将服务集内的所有车辆进行排序,选择最优性能的服务车辆,其搜寻复杂度同样为  $O(N)$ 。第 9-11 行进行用户卸载回报、协助计算车辆选择次数以及  $\Delta_{i,n}$  的更新,其复杂度为  $O(1)$ ,因此在每个阶段算法复杂度为  $O(2N+1)$ 。当用户包含  $M$  个子任务,并且每个子任务的上传处理为相互独立时,其算法复杂度为  $O(2MN+M)$ ,化简可得复杂度为  $O(MN)$ 。

## 5.3 信令分析

假设在车联网系统中车辆计算能力与信道状态是连续已知的(通过发送信令信息或请求 RSU 路测单元),那么用户将会通过已知信息来计算最大的时延与能耗加权回报,并选择该回报所对应的服务车辆进行任务卸载,这是基于理想状况下或者说是静态下的假设。然而,在实际环境中,由于车辆的高移动性,用户想要获取周围服务车辆的服务能力信息和周围动态的信道状态信息是困难的,需要频繁向周围车辆或 RSU 路测单元发送信令信息,这样一来,将会造成不必要的

发射能耗与等待时延。

如果考虑多用户之间的信息反馈,系统的信令开销会变大。假设系统中有  $k$  台用户车辆,每个用户之间都需要共享服务车辆状态信息,且信息发送过程独立,可以得到当多个用户共同反馈回报信息时,所需要的信令数目为  $\frac{k(k-1)}{2}$ 。并且由于用户之间相对运动的影响,其车辆数  $k$  是一个随机变量,复杂的运动关系与时变的信道环境使得多用户之间的共同反馈非理想化。因此在模型中不考虑多用户联合信息反馈。

当用户车辆向周围车辆发送询问状态信息时,假设发送一台车辆消耗  $\tau$  的单位时延以及  $E$  的单位能耗,那么发送给  $m$  个车辆将需要  $m\tau$  个单位的时延消耗;当用户车辆请求 RSU 服务器时,由于车辆状态信息需要经过两次转移,即由服务车辆到 RSU,再经 RSU 服务器到达用户端,在忽略中转处理时延的情况下所产生的时延大于用户将询问信息直接发送给协助计算车辆的时延,因此其时延复杂度为  $O(m\tau)$ ,同理可以得到其能耗复杂度  $O(mE)$ 。

## 5.4 性能分析

本节将分析算法 1 的性能。算法 1 的提出建立在 MAB 理论之上,是在传统 UCB 算法上的改进。算法 1 综合考虑了无线信道环境变化与协助计算车辆负载变化对卸载回报的影响,并将两个阶段的卸载回报差作为算法中的探索项。

### 5.4.1 定义悔恨度

为了维护服务集中车载设备的服务性能,本文考虑将每个子任务的卸载看作一个阶段,因此定义集合  $T = \{1, 2, \dots, t, \dots, T\}$  表示用户卸载子任务的阶段数,假设  $\mu_t^*$  为第  $t$  阶段中最优的卸载回报,  $\mu_{t,c}$  为第  $t$  阶段用户实际获得的卸载回报。

由于学习过程会造成卸载回报损失,相比传统智能优化算法,算法 1 并不总是选择平均反馈最高的服务车辆,而是在未知先验条件的前提下选择尽可能最优的服务车辆,因此定义学习悔恨为所有阶段最优卸载回报与用户实际获得卸载回报的差值。

$$R_T = \sum_{t=1}^T \mathbb{E}[\mu_{t,c} - \mu_t^*] \quad (18)$$

### 5.4.2 在固定的信道环境中进行悔恨分析

本节假设无线信道不随时间改变,仅考虑服务集中车辆的计算能力。此时算法只需要学习协助计算车辆的计算能力,式(16)转化为:

$$\hat{U}_{i+1,n} = \bar{U}_{i,n} + \sqrt{\frac{\beta \ln i}{n_{i,n}}} \quad (19)$$

当环境变化不被考虑后,用户仅学习辅助计算车辆的计算能力,此时算法 1 将转变为传统的 UCB1 算法<sup>[9]</sup>。用户的学习悔恨描述为:

$$R_T = \sum_{t=1}^T \mathbb{E}[\mu_{t,c} - T\mu_t^*] \quad (20)$$

**定理 1** 当  $\beta=2$ ,且子任务卸载进行到  $T$  阶段时,用户在执行算法 1 后的学习悔恨上界为:

$$R_T \leq \left[ \sum_{n_j < \mu^*} \frac{8 \ln T}{\delta_j} + \left(1 + \frac{\pi^2}{3}\right) \sum_{n_j < \mu^*} \delta_j \right] \quad (21)$$

证明:参考文献[33]中的引理 1 以及文献[9]中的定理 1,当  $\beta=2$ ,且用户任务卸载进行至  $t$  阶段时,SV 被选择次数的期望可以表达为:

$$\mathbb{E}[C_j(t)] \leq \frac{8 \ln t}{\delta_j^2} + 1 + \frac{\pi^2}{3} \quad (22)$$

其中  $\delta_j = \mu_{j,c} - \mu_j^*$ , 表示  $SV_j$  与最优服务性能车辆的卸载回报差。

因此可以得到当任务卸载进行到第  $t$  阶段时, 用户在  $ACV_j$  累积的学习悔恨为:

$$\begin{aligned} R_j &= \delta_j \mathbb{E}[C_j(t)] \\ &\leq \delta_j \left( \frac{8 \ln t}{\delta_j^2} + 1 + \frac{\pi^2}{3} \right) \\ &= \frac{8 \ln t}{\delta_j} + \left( 1 + \frac{\pi^2}{3} \right) \delta_j \end{aligned} \quad (23)$$

式(27)为一台协助计算车辆的累积悔恨, 将其求和可以得到用户在所有协助计算车辆上卸载子任务的总悔恨。

$$\begin{aligned} R_T &= \sum_{t=1}^T \mathbb{E}[\mu_{t,c} - \mu_t^*] \\ &= \sum_{j=1}^N \delta_j \mathbb{E}[C_j(T)] \\ &\leq \left[ \sum_{\mu_j < \mu^*} \frac{8 \ln T}{\delta_j} + \left( 1 + \frac{\pi^2}{3} \right) \sum_{\mu_j < \mu^*} \delta_j \right] \end{aligned} \quad (24)$$

从定理 1 可以得到, 相比传统的优化策略, 算法 1 能够达到次线性学习悔恨。详细来说, 当  $\delta_j$  较大时,  $ACV_j$  的服务性能越差, 在该车辆上计算子任务的概率越低。当子任务卸载进行到  $T$  阶段时, 产生的学习悔恨为  $O(\ln T)$ 。

此外, 当任务卸载场景简化为具有固定服务集和固定信道状态的情况时, 所提算法简化为传统的置信上限算法(UCB), 并且文献[30-32]研究了学习悔恨的下界, 由文献[30]可以得到  $ACV_j$  被选择次数的期望为:

$$\mathbb{E}[C_j(t)] \geq \left( \frac{1}{D(j, b^*)} + o(1) \right) \ln t \quad (25)$$

其中,  $D(j, b^*) \stackrel{\text{def}}{=} \int l_j \ln \frac{l_j}{l^*}$  是卸载回报分布的 Kullback-Leibler 散度,  $l_j$  与  $l^*$  为  $ACV_j$  与  $ACV_{b^*}$  的卸载回报分布。

将式(26)代入式(22)中, 可以得到学习悔恨的下界:

$$\begin{aligned} R_T &= \sum_{j=1}^N \delta_j \mathbb{E}[C_j(t)] \\ &\geq \sum_{j=1}^N \delta_j \left( \frac{1}{D(j, a^*)} + o(1) \right) \ln t \\ &\geq \sum_{j=1}^N \frac{\delta_j \ln t}{D(j, a^*)} \end{aligned} \quad (26)$$

当信道环境固定之后, 算法 1 将只学习对车辆计算的能力, 这种学习方式符合传统的 MAB 算法, 即不考虑环境本身发生太大变化的情况下进行学习, 此时学习悔恨只跟服务集中的服务车辆与用户进行卸载的轮次有关, 当服务集中包含太多的服务车辆或者用户上传更多的子任务, 都会导致更高的学习悔恨。

#### 5.4.3 在变化的信道环境中进行悔恨分析

本节将分析算法 1 在变化信道环境中对学习悔恨的影响, 当考虑变化的无线环境时, 仅通过计算效用来评判服务车辆的计算能力并不实际。假设在车联网系统中存在 4 种调度场景: 1) 协助计算车辆计算能力强, 与用户之间的无线信道环境良好; 2) 协助计算车辆计算能力强, 与用户之间的无线信道环境差; 3) 协助计算车辆计算能力弱, 与用户之间的无线信道环境良好; 4) 协助计算车辆计算能力弱, 与用户之间的无线信道环境差。

通过对前文公式的分析可以得出, 每一个阶段的卸载回报随着服务车辆计算能力与无线信道环境质量的提高而

提高, 因此考虑信道环境的变化, 使探索和利用达到平衡, 以实现更好的性能。

当无线信道的变化考虑在内时, 式(16)变成:

$$\hat{U}_{t+1,n} = \bar{U}_{t,n} + \sqrt{\frac{e^{\epsilon \Delta_{t,n}} \beta \ln t}{n_{t,n}}} \quad (27)$$

当  $\Delta_{t,n} < 0$  时, 后一阶段的卸载回报小于前一阶段的卸载回报,  $e^{\epsilon \Delta_{t,n}} \in (0, 1)$ , 探索项系数的值变小, 该服务车辆的服务稳定性逐渐变差, 在探索和利用之间, 用户将会以更大的可能性去选择其他协助计算车辆, 当  $\Delta_{t,n} = 0$  时, 后一阶段的用户效用等于前一阶段的效用,  $e^{\epsilon \Delta_{t,n}} = 1$ , 说明该车辆的服务性能较为稳定, 当  $\Delta_{t,n} > 0$  时, 后一阶段的卸载回报大于前一阶段的卸载回报,  $e^{\epsilon \Delta_{t,n}} \in (1, e^{2\epsilon})$ , 说明协助计算车辆的服务性能趋于良好, 用户选择该协助计算车辆的概率提高。

**定理 2** 假设  $X_1, X_2, \dots, X_n$  为随机变量, 满足  $\mathbb{E}[X_n | X_1, X_2, \dots, X_{n-1}] = \psi$ ,  $S_n = X_1 + X_2 + \dots + X_n$ , 当  $ACV_m$  的探索项引入有界自适应系数  $e^{\epsilon \Delta_t}$  后, 满足以下不等式:

$$\mathbb{P} \left\{ \overline{X_{t,m}} \geq \psi_{t,m} + \sqrt{\frac{e^{\epsilon \Delta_{t,m}} \beta \ln t}{n_{t,m}}} \right\} \leq (t e^{\epsilon \Delta_{t,m}})^{-4} \quad (28)$$

证明: 假设  $X_1, X_2, \dots, X_n$  为随机变量, 满足  $\mathbb{E}[X_n | X_1, X_2, \dots, X_{n-1}] = \psi$ ,  $S_n = X_1 + X_2 + \dots + X_n$ , 对于所有的  $a \geq 0$ , 由 Chernoff-Hoeffding 不等式可得:

$$\mathbb{P}[S_n \geq n\psi + a] \leq e^{-2a^2/n} \quad (29)$$

$$\mathbb{P}[S_n \leq n\psi - a] \leq e^{-2a^2/n} \quad (30)$$

令  $b = \frac{a}{n}$ , 则  $a = bn$ , 代入式(30)可以转化为式(31):

$$\mathbb{P} \left[ \frac{S_n}{n} \leq \psi - b \right] \leq e^{-2(\ln n)^2/n} \quad (31)$$

将  $b = \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}}$  代入式(31)可得:

$$\mathbb{P} \left[ \frac{S_n}{n} \leq \psi - \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \right] \leq e^{-2 \left( \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \right)^2 / n}$$

$$\mathbb{P} \left[ \frac{S_n}{n} \leq \psi - \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \right] \leq e^{-2 \frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \quad (32)$$

$$\mathbb{P} \left[ \overline{X_n} \leq \psi - \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \right] \leq e^{-2 e^{\epsilon \Delta_t} \beta \ln t}$$

$$\mathbb{P} \left[ \overline{X_n} \leq \psi - \sqrt{\frac{e^{\epsilon \Delta_t} \beta \ln t}{n}} \right] \leq e^{-4 e^{\epsilon \Delta_t} \ln t} = t^{-4 e^{\epsilon \Delta_t}}$$

对于  $ACV_m$ , 则有:

$$\mathbb{P} \left\{ \overline{X_{t,m}} \geq \psi_{t,m} + \sqrt{\frac{e^{\epsilon \Delta_{t,m}} \beta \ln t}{n_{t,m}}} \right\} \leq (t e^{\epsilon \Delta_{t,m}})^{-4} \quad (33)$$

由于  $\epsilon$  与  $\Delta_t$  的有界性, 随着卸载阶段数  $t$  的增大, 后一项的值趋于 0。

从定理 1 可以看出, 当探索项引入自适应系数  $e^{\epsilon \Delta_{t,n}}$  后, 随着子任务卸载阶段  $t$  的增大, 用户的卸载回报超过  $\psi_{t,n} + \sqrt{\frac{e^{\epsilon \Delta_{t,n}} \beta \ln t}{n_{t,n}}}$  的概率趋于 0, 证明了新的探索系数对置信度的影响可忽略不计。

在之后通过仿真实验来分析算法 1 对学习悔恨以及用户能效的影响。

#### 5.4.4 服务集合更新算法

在实际车联网场景中, 由于协助计算车辆和用户车辆的高移动性, 以及复杂的无线环境, 固定服务集的方式具有太大的局限性, 并且过于频繁地更新服务集将使得算法学习能力

变差,学习悔恨也将随之变高。

因此,本节提出了一种用于更新和维护服务集质量的更新算法,使得用户在每一个卸载阶段获得更高的卸载能效。

#### 算法2 服务集更新算法

输入:  $U_{i,n}, U_i$

输出: 新一阶段服务集  $N(t)$

1. 对于所有的  $i \in (1, \dots, M)$

2. 对于所有的  $n \in (1, \dots, N)$

3.  $\mu_{i,n} = U_{i,n} - U_i$

4.  $\theta_i = \max[\mu_i] - \min[\mu_i]$

5.  $\gamma_{i,n} = \frac{\mu_{i,n} - \min[\mu_i]}{\theta_i}$

6. if  $\gamma_{i,n} < \rho_i$  then

7.  $SV_n \notin \mathcal{A}(t)$

8. 服务集重新搜索并连接一台新的协助计算车辆

9. else  $\gamma_{i,n} \geq \rho_i$

10.  $SV_n \in \mathcal{A}(t)$

算法2 将一个阶段服务集中所有协助计算车辆的卸载回报作为输入,通过卸载回报与阈值的差值来判断服务性能的好坏,进一步更新服务集。其中第4-6行为卸载回报归一化阶段,得到的  $\gamma_{i,n} \in (0, 1)$ ,表示  $ACV_n$  在计算子任务后得到的卸载回报与总的卸载回报之比。第7-11行为服务集更新过程,其中  $\rho_i \in (0, 1)$  为阈值,表示用户所能接受最差的卸载回报,当  $\gamma_{i,n} < \rho_i$  时,服务集  $N(t)$  将会剔除  $ACV_n$ ,并且引入一台新的  $ACV$ ,其置信度与原  $ACV_n$  保持一致;当  $\gamma_{i,n} \geq \rho_i$  时,  $ACV_n$  仍处在服务集  $N(t)$  中,直到本阶段  $N(t)$  更新完毕。

与传统的  $\epsilon$ -greedy 贪婪决策算法不同,算法2 在服务集更新时在一个阶段可以剔除多辆服务性能较差的协助计算车辆,并且保证了性能更优的协助计算车辆不会被剔除。在下一节仿真中分析算法2 的引入对用户服务性能的影响。

## 6 仿真分析

为了测试算法1 和算法2 的性能,本节进行了仿真。仿真工具使用 python 软件,在 pycharm 平台对结果进行数据可视化。具体参数设置如表1 所列。首先,与一些现有算法进行性能对比,分析性能表现与收敛性,其次分析了各参数对学习悔恨的影响。

表1 仿真参数

Table 1 Simulation parameters

仿真参数	参数值
服务集中车辆数 $N$	10
噪声功率 $\sigma^2/W$	$10^{-13}$
用户的信道增益	$h = -17.8d^{-2}$
输入数据大小 $D/\text{Mbit}$	2~3
所需的 CPU 周期数/bit	1000 Cycles/bit
系统带宽 $B$	1 MHz
用户发射功率 $P/W$	0.5
用户与协助计算车辆相对位置变化/m	$[-5, 5]$
用户与协助计算车辆的距离 $d/\text{m}$	$[100, 1000]$
协助计算车辆计算能力/(Cycles/s)	$0.2 \times 10^9 \sim 0.6 \times 10^9$
子任务卸载阶段数	1800

图2 给出了算法1 与4种 MAB 强化学习算法的对比结果,包括传统的  $\epsilon$ -greedy 算法、置信上限算法,该算法最初是在文献[9]中提出,其中探索项的值为  $\sqrt{\frac{\beta \ln t}{nt, n}}$ ,该算法只能学习固定环境与固定车辆下的计算能力;还包括 AdaUCB

算法<sup>[26]</sup>,其中探索项的值为  $\sqrt{\frac{\beta(1-D_i)\ln t}{nt, n}}$ ,且  $D_i$  为归一化后的输入数据大小,该算法将  $1-D_i$  作为探索项系数,从而降低学习悔恨;还包括文献[30]提出的 ALTO 算法,其中探索项的值为  $\sqrt{\frac{\beta(1-D_i)\ln(t-t_n)}{nt, n}}$ ,该算法结合了输入感知与发生感知,进一步降低了学习悔恨。为了能够与最优的卸载决策进行对比,假设用户始终会将任务卸载到服务质量最优的车辆来分析所提算法的性能,当获得服务车辆的相关信息(计算能力、实时信道环境)之后,就能够利用现有的一些优化方法做出最优决策。在数值设计中,协助计算车辆的计算能力设置在  $0.2 \times 10^9$  Cycles/s 到  $0.6 \times 10^9$  Cycles/s 之间,并且初始化数值呈均匀分布。

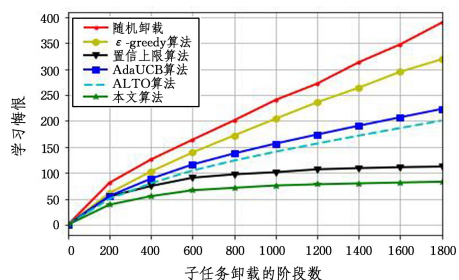


图2 本文算法与现有算法的悔恨值比较

Fig. 2 Regrets comparison between the proposed algorithm and existing algorithms

图2 给出了不同算法下的学习悔恨值。从该图中可以很明显获得两个信息,首先,算法1 在性能上优于其他几种算法;其次,该算法下的学习悔恨随着阶段数的增长呈次线性增长。

图3 给出了不同权重系数  $\beta$  对学习悔恨的影响。可以看出,当  $\beta=1$  时,学习悔恨达到最低水平,当  $\beta$  的值逐渐变大时,由于探索项的影响,用户更倾向于去探索服务集中其他车辆,此时学习悔恨值将随着  $\beta$  的增大而增大,并且变化趋势更快。文献[31]的研究表明,当  $\beta > 0.5$  时,达到次线性学习遗憾。本文实验结果显示,当  $\beta=1$  时学习悔恨较低,可能是因为将二阶收益差作为探索项系数后,用户需要较多次的探索才能找到最优的协助计算车辆。

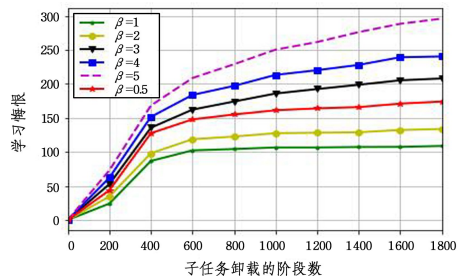


图3 不同  $\beta$  系数对学习悔恨的影响

Fig. 3 Influence of different  $\beta$  coefficients on learning regrets

图4 给出了不同控制因子  $\epsilon$  对学习悔恨的影响。从实验结果可以看出,当  $\epsilon=0$  时,学习悔恨达到较高水平,这是由于当  $\epsilon=0$  时,二阶收益差不再作为探索项系数,此时本文算法将转化为传统的置信上限(UCB)算法,该算法下的学习悔恨较高,意味着用户卸载回报低。当  $\epsilon=1$  时学习悔恨达到最低水平,当  $\epsilon > 1$  时,学习悔恨逐渐变高,原因是  $\epsilon$  与二阶收益差

是相乘的关系,当 $\epsilon$ 变大时,探索水平增高,用户更容易放弃最优车辆,将任务卸载到其他车辆上,导致卸载回报变低。

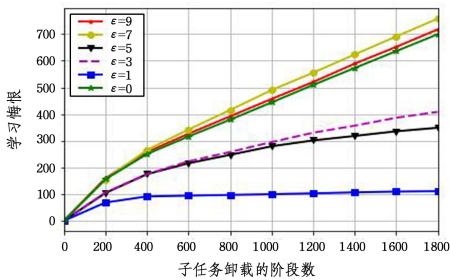


图4 不同 $\epsilon$ 系数对学习悔恨的影响

Fig. 4 Influence of different  $\epsilon$  coefficients on learning regrets

图5给出了不同算法下的时延性能。从实验结果可以看出,本文算法具有较低的时延性能,并且随着子任务卸载阶段数的增长趋于平稳,由于服务集和信道环境的变化,与本文算法相比,现有一些算法在时延方面的性能并不理想,其中 $\epsilon$ -greedy算法、置信上限算法以及AdaUCB算法下的时延随着子任务卸载阶段数的增高而变大,ALTO算法下的时延虽然没有较大变化,但其总体时延较大,与本文算法下的时延相差7ms左右。

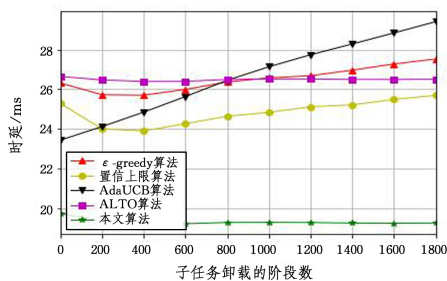


图5 本文算法与现有算法时延性能的比较

Fig. 5 Delay performance comparison between the proposed algorithm and existing algorithms

图6给出了在引入算法2后,不同阈值 $\rho$ 下协助计算车辆数对学习悔恨的影响。为了展现算法2对悔恨性能的影响,在仿真中引入了 $\rho=0$ 时的学习悔恨,即服务集处于非更新状态时的悔恨性能。

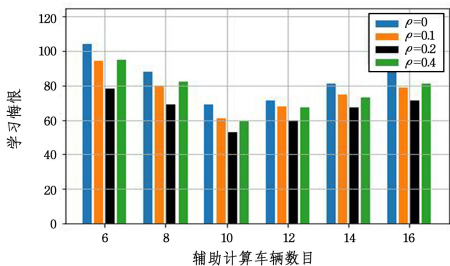


图6 不同阈值 $\rho$ 下协助计算车辆数对学习悔恨的影响

Fig. 6 Influence of vehicle number on learning regret at different threshold  $\rho$

从实验结果可以看出,当 $\rho=0$ 时,服务集中一直存在性能较差的协助计算车辆,而用户依然有概率会探索到该车辆,因此该阈值下的学习悔恨较高。当 $\rho=0.1$ 或 $\rho=0.2$ 时,学习悔恨降低,并且在 $\rho=0.2$ 时更低,这是由于服务集会筛选性能较差的协助计算车辆,并且新引入一台车辆。当 $\rho=0.4$ 时悔恨值却增大了,这是因为服务集更新太快,将一些性能

较好的协助计算车辆也进行了更新。与此同时,当协助计算车辆的数目控制在10~12时,学习悔恨达到最低,可能的原因是车辆较多时,用户探索的成本就越高;车辆较少时,其最大可用资源较低。

图7给出了当 $\rho=0.2$ 时,不同算法下协助计算车辆数目对学习悔恨的影响。从图7可以看出,与现有算法相比,在协助计算车辆数目相同的情况下,算法2的学习悔恨最低,并且置信上限算法的性能优于AdaUCB算法,这是因为服务集更新的引入所带来的性能提升优于输入数据量的考虑。

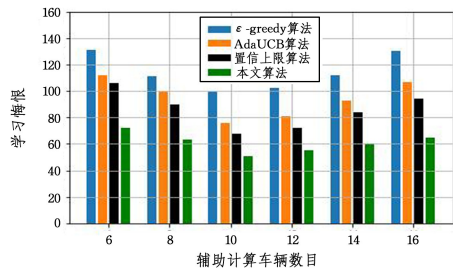


图7 不同算法下协助计算车辆数对学习悔恨的影响

Fig. 7 Influence of vehicle number on learning regret with different algorithms

**结束语** 本文研究了在车载边缘计算(VEC)系统中用户车辆的任务卸载问题,提出了一种启发式二阶探索强化学习算法,用于最大化用户的平均卸载回报。该算法使得用户车辆在缺乏状态信息的情况下,独立学习协助计算车辆的卸载回报性能。考虑到车载设备计算能力的变化与移动性所引起的信道环境特性,本文将基于多臂老虎机框架置信上限算法中的探索项重新进行设计,使该算法能够适合动态车载设备下的任务卸载环境,通过理论分析证明了该算法具有次线性学习悔恨。通过仿真分析,证明了所提算法在用户卸载回报方面有很大的改善,并且降低了卸载时延,在每一个卸载阶段结束之后,提出了一种服务集更新算法,用于维护协助计算车辆的服务性能,与非更新方式相比,该方案能够进一步降低学习悔恨,将卸载回报提升13%。

在实际场景中,车载设备不会无偿提供计算资源,之后的工作会考虑将资源定价引入车载边缘计算网络,激励更多的车载设备提供计算资源。同时考虑使用深度强化学习来解决更复杂的资源调配问题。

## 参考文献

- [1] QI W, LI Q, SONG Q, et al. Extensive edge intelligence for future vehicular networks in 6G[J]. IEEE Wireless Communications, 2021, 28(4): 128-135.
- [2] KHAYYAT M, ELGENDY I A, MUTHANNAA, et al. Advanced deep learning-based computational offloading for multi-level vehicular edge-cloud computing networks[J]. IEEE Access, 2020, 8: 137052-137062.
- [3] KHAYYAT M, ALSHAHRANI A, ALHARBI S, et al. Multi-level service-provisioning-based autonomous vehicle applications[J]. Sustainability, 2020, 12(6): 2497.
- [4] DINH T Q, TANG J, LA Q D, et al. Offloading in mobile edge computing: Task allocation and computational frequency scaling[J]. IEEE Transactions on Communications, 2017, 65(8): 3571-3584.

- [5] SUN J, GU Q, ZHENG T, et al. Joint communication and computing resource allocation in vehicular edge computing[J]. *International Journal of Distributed Sensor Networks*, 2019, 15(3): 1550147719837859.
- [6] HA K, PILLAI P, LEWIS G, et al. The impact of mobile multimedia applications on data center consolidation[C]//2013 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2013:166-176.
- [7] DAI Y, XU D, MAHARJAN S, et al. Joint offloading and resource allocation in vehicular edge computing and networks [C]//2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018:1-7.
- [8] CHEN S, TAO Y, YU D, et al. Distributed learning dynamics of multi-armed bandits for edge intelligence[J]. *Journal of Systems Architecture*, 2021, 114:101919.
- [9] AUER P, CESA-BIANCHI N, FISCHER P. Finite-time analysis of the multiarmed bandit problem[J]. *Machine Learning*, 2002, 47:235-256.
- [10] CUI Y, LIANG Y, WANG R. Resource allocation algorithm with multi-platform intelligent offloading in D2D-enabled vehicular networks[J]. *IEEE Access*, 2019, 7:21246-21253.
- [11] LI L, ZHOU H, XIONG S X, et al. Compound model of task arrivals and load-aware offloading for vehicular mobile edge computing networks[J]. *IEEE Access*, 2019, 7:26631-26640.
- [12] HAITAO Z, YI D, MENGKANG Z, et al. Multipath transmission workload balancing optimization scheme based on mobile edge computing in vehicular heterogeneous network[J]. *IEEE Access*, 2019, 7:116047-116055.
- [13] DENG Z, CAI Z, LIANG M. A multi-hop VANETs-assisted offloading strategy in vehicular mobile edge computing[J]. *IEEE Access*, 2020, 8:53062-53071.
- [14] HUANG C M, LAI C F. The delay-constrained and network-situation-aware V2V2I VANET data offloading based on the multi-access edge computing(MEC) architecture[J]. *IEEE Open Journal of Vehicular Technology*, 2020, 1:331-347.
- [15] XUE J, HU Q, AN Y, et al. Joint task offloading and resource allocation in vehicle-assisted multi-access edge computing[J]. *Computer Communications*, 2021, 177:77-85.
- [16] SUN J, GU Q, ZHENG T, et al. Joint communication and computing resource allocation in vehicular edge computing[J]. *International Journal of Distributed Sensor Networks*, 2019, 15(3): 1550147719837859.
- [17] HUANG X, YU R, XIE S, et al. Task-container matching game for computation offloading in vehicular edge computing and networks[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 22(10):6242-6255.
- [18] ZHANG Y, WANG C Y, WEI H Y. Parking reservation auction for parked vehicle assistance in vehicular fog computing[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(4): 3126-3139.
- [19] HUANG X, YU R, YE D, et al. Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2021, 70(4):3773-3787.
- [20] RAZA S, MIRZA M A, AHMAD S, et al. A vehicle to vehicle relay-based task offloading scheme in Vehicular Communication Networks[J]. *PeerJ Computer Science*, 2021, 7:e486.
- [21] BUDA S, GULENG S, WUC, et al. Collaborative vehicular edge computing towards greener ITS[J]. *IEEE Access*, 2020, 8: 63935-63944.
- [22] CHA N, WU C, YOSHINAGAT, et al. Virtual edge: Exploring computation offloading in collaborative vehicular edge computing[J]. *IEEE Access*, 2021, 9:37739-37751.
- [23] BUTE M S, FAN P, 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.
- [24] ZHANG X, ZHANG J, LIU Z, et al. MDP-based task offloading for vehicular edge computing under certain and uncertain transition probabilities[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(3):3296-3309.
- [25] AUER P, CESA-BIANCHI N, FISCHER P. Finite-time analysis of the multiarmed bandit problem[J]. *Machine learning*, 2002, 47(2):235-256.
- [26] WU H, GUO X, LIU X. Adaptive exploration-exploitation tradeoff for opportunistic bandits[C]//International Conference on Machine Learning. PMLR, 2018:5306-5314.
- [27] SUN Y, GUO X, SONG J, et al. Adaptive learning-based task offloading for vehicular edge computing systems[J]. *IEEE Transactions on vehicular technology*, 2019, 68(4):3061-3074.
- [28] MAO Y, ZHANG J, SONG S H, et al. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems[J]. *IEEE Transactions on Wireless Communications*, 2017, 16(9):5994-6009.
- [29] GU X, ZHANG G, CAO Y. Cooperative mobile edge computing-cloud computing in Internet of vehicle: Architecture and energy-efficient workload allocation[J]. *Transactions on Emerging Telecommunications Technologies*, 2021, 32(8):e4095.
- [30] SALOMON A, AUDIBERT J Y, ALAOU I E. Regret lower bounds and extended Upper Confidence Bounds policies in stochastic multi-armed bandit problem[J]. *arXiv:1112.3827*, 2011.
- [31] BUBECK S. Bandits games and clustering foundations[OL]. Université des Sciences et Technologie de Lille-Lille I, 2010. <https://sol.cljtsed.com/scholar?q=Bandits+games+and+clustering+foundations>.
- [32] BUBECK S, CESA-BIANCHI N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems[J]. *Foundations and Trends © in Machine Learning*, 2012, 5(1):1-122.
- [33] BNAYA Z, PUZIS R, STERNR, et al. Social network search as a volatile multi-armed bandit problem[J]. *Human*, 2013, 2(2): 84.



**XUE Jianbin**, born in 1973, Ph.D, associate professor. His main research interests include wireless communication theory and technology, mobile edge computing technology, etc.



**WANG Hainiu**, born in 1997, postgraduate. His main research interests include resource allocation, vehicular edge computing, and reinforcement learning.