



硕士学位论文

车联网边缘计算资源分配与任务卸载方法 研究

Research on Resource Allocation and Task
Offloading for Vehicular Edge Computing
Networks

作 者：孙尉斌
导 师：孙彦景教授

中国矿业大学

2022 年 5 月

中图分类号 TN929.5

学校代码 10290

UDC 645

密 级 公开

中国矿业大学

硕士学位论文

车联网边缘计算资源分配与任务卸载方法研究

**Research on Resource Allocation and Task
Offloading for Vehicular Edge Computing Networks**

作 者 孙尉斌

导 师 孙彦景

申请学位 工学硕士学位

培养单位 信息与控制工程学院

学科专业 信息与通信工程

研究方向 车联网资源分配

答辩委员会主席 刘辉

评 阅 人 盲审

二〇二二年五月

致谢

时光如白驹过隙，转眼间研究生生活即将结束。回望在中国矿业大学学习生活的七年时光，十分感慨，2500多个充满奋斗与收获的夜晚已匆匆逝去。漫漫求学路上有不少迷茫与无助的时刻，幸得多位良师益友相助，才能顺利完成学业。在此论文完成之际，向在学习生涯中引导我、帮助我、鼓励我的人，道一声感谢。

由衷地感谢我的导师孙彦景教授与李松副教授对我的精心指导与细心关怀。在科研探索的过程中，孙老师凭借渊博的学识以及科研工作敏锐的洞察能力，总能够在指导中让我有醍醐灌顶的顿悟之感。在论文的写作过程中，李老师总是积极地与我讨论研究方案，给出了许多建设性的意见，并耐心地一遍遍审阅我的论文，以身作则地培养我严谨认真的科研态度。二位老师对我的谆谆教诲我会一直铭记于心，在此，对二位老师表达最诚挚的敬意与感谢！

感谢实验室团队的同学们。在读研的三年期间，我们互相帮助，一同学习，共同进步，与大家度过的美好时光是我这一生中的宝贵财富。由衷地祝各位同门前程似锦，愿我们友谊长存。

感谢我的父母与亲友对我的支持与鼓励，在我每一次遇到困难时给予我积极面对的勇气和力量，让我能够不断地超越自我，完成学业。也十分感谢女友宋妮娟一直以来的陪伴与支持。

最后，感谢在百忙中抽出时间评阅论文与参加答辩的各位专家和教授！

摘要

物联网与移动通信技术的发展促进了车联网的快速发展。而在车联网中,存在着大量诸如无人驾驶的计算密集型与时延敏感型应用,仅凭车辆自身有限的计算资源难以保障服务质量。移动边缘计算(Mobile Edge Computing, MEC)将云服务下放至网络的边缘端,既能够为车辆提供额外的计算资源,又能够避免因通信距离过长而导致的时延与能耗增加。因此,将移动边缘计算引入车联网中,构成的车辆边缘计算(Vehicular Edge Computing, VEC)成为了一个具有前景的解决方案。然而,边缘服务器的资源有限,在交通流量增大时难以同时满足系统中所有用户的计算需求。此外,车辆的移动性会使网络拓扑结构快速变化,从而频繁导致通信链路断裂、基站切换的情况出现,降低了计算卸载效率。因此,在车联网边缘计算场景中,如何设计高效的资源分配方案,如何利用网络中的空闲资源提高资源利用率以及如何针对车辆的移动性提出有效的任务卸载方案是本文的研究重点。本文的具体研究内容与创新点如下:

1. 针对交通流量大的车辆边缘计算网络中存在的资源受限问题,提出了基于通信与计算资源联合优化的集中式资源分配与任务卸载方案。在问题的建模中,引入了动态电压调节技术(Dynamic Voltage Scaling, DVS)技术,通过对车载计算单元工作频率、通信发射功率、车辆的任务卸载量与边缘服务器的计算资源分配进行联合优化,在满足任务时延约束的条件下优化了车辆用户的能耗。为了解决非凸优化问题,采用凸差规划(Difference of Convex, DC)将其解耦为两个凸的子问题,并利用拉格朗日对偶分解法进行联合迭代求解。最后通过仿真实验验证了所提模型与算法在能耗优化方面的有效性。

2. 针对车辆边缘计算网络中因车辆移动性而导致的任务卸载效率下降的问题,提出了一种基于强化学习的分布式在线学习任务卸载策略。为了应对高度动态的车辆边缘计算网络环境,将任务卸载过程建模为多臂赌博机(Multi-armed bandit, MAB)问题,优化目标为最小化平均任务时延。为了解决动态环境下的任务卸载问题,提出了基于多臂赌博机的分布式在线学习任务卸载算法,使车辆用户能够在任务卸载的过程中学习周围环境的信息,从而不断优化自身的卸载策略。为了提高算法的学习效率,设计了基于路边单元(Road Side Unit, RSU)辅助的用户协作机制,使用户可以共享彼此学习到的信息。理论分析表明,所提算法可获得对数形式的遗憾函数。仿真结果表明,所提算法与其他现有算法相比,更能适应复杂时变的车辆边缘计算网络环境。

3. 针对车辆边缘计算网络任务卸载过程中存在的用户资源竞争现象以及用户公平性难以保证的问题,提出了基于公平匹配多臂赌博机的分布式任务卸载策略。为了保障用户间的公平性,该方案以满足所有用户的服务质量需求为目标,

建立用户与计算服务提供者之间的匹配，用户按照匹配的结果进行任务卸载。为了解决时延约束下的平均能耗最小化问题，本文提出了一种基于李雅普诺夫优化的动态任务卸载算法，将优化问题解耦为任务卸载决策子问题与功率优化子问题。任务卸载决策子问题通过基于公平匹配多臂赌博机的算法进行求解，功率优化子问题通过基于 DC 规划的优化算法进行求解。仿真结果表明，所提算法能够有效地避免资源竞争的情况，同时保障用户间的公平性。

关键词：移动边缘计算；车联网；任务卸载；凸优化；强化学习

Abstract

The development of Internet of Things (IoT) and Mobile communication technology promotes the rapid development of the Internet of vehicles (IoV). However, there are a large number of computation-intensive and delay-sensitive applications in the Internet of vehicles, such as unmanned driving, which is difficult to guarantee the quality of service (QoS) requirements only by the limited computing resources of vehicles. Mobile edge computing (MEC), which brings cloud services to the edge of the network, could provide additional computing resources for the vehicles and avoid the increase of delay and energy consumption caused by the long communication distance. Therefore, by introducing mobile edge computing in to the Internet of vehicles, vehicular edge computing (VEC) has become a promising solution for above challenge. However, the resources in the edge server are also limited, which makes it difficult to simultaneously meet the computing needs of all users in the system when the traffic flow increases. In addition, the mobility of vehicles will make the network topology change rapidly, resulting in frequent communication link fracture and base station switching, which reduces the computation offloading efficiency. To sum up, this thesis focuses on designing efficient resource allocation schemes, improving resource utilization by using idle resources in the network and designing effective task offloading schemes to against vehicle's inherent mobility in the vehicular edge computing network. The main contributions of this thesis are as follows:

1. Aiming at the problem of resource limitation in vehicle edge computing network caused by heavy traffic flow, a centralized resource allocation and computation offloading scheme with joint optimization of communication and computation resources is proposed. In the system model, dynamic voltage scaling (DVS) technology is introduced to jointly optimize the operating frequency of on-board CPU, the communication transmitting power, the task load of vehicles and the computing resource allocation of edge servers. The energy consumption of vehicle is optimized under delay constraint. To solve the non-convex optimization problem, the Difference of Convex function (DC) programming is used to decouple the non-convex problem into two convex sub problems, and the Lagrange multiplier method is used to solve the problem jointly. Finally, the effectiveness of the proposed model and algorithm in energy consumption optimization is verified by simulation experiments.

2. Aiming at the decreasing of task offloading efficiency caused by the mobility of

vehicle nodes in vehicle edge computing network, a distributed online learning task offloading strategy based on reinforcement learning is proposed. In order to cope with the highly dynamic vehicle edge computing network environment, the task offloading process is modeled as a multi-armed bandit (MAB) problem and the optimization objective is to reduce the average task delay. An MAB based distributed online learning task offloading algorithm is proposed in this thesis to solve the task offloading problem with dynamic environment, which enables vehicle nodes to learn the information of the environment during task offloading, so as to continuously optimize its offloading strategy. To improve the learning efficiency of the algorithm, a Road Side Unit (RSU) assisted user collaboration mechanism is designed so that vehicles in the system can share the information they learned from each other. Theoretical analysis shows that the regret function of the proposed algorithm is logarithmic. Simulation results show that the proposed algorithm is more suitable for complex time-varying vehicle edge computing networks than other existing algorithms.

3. Aiming at the problem of resource competition and the difficulty in ensuring user fairness in the task offloading process of vehicle edge computing network, a distributed task offloading strategy based on fair matching multi-arm bandit algorithm is proposed. In order to ensure the fairness among users, this thesis establishes the matching between users and computing service providers to meet the Quality of Service requirements of all users. Users offloading tasks according to the matching results. In order to minimize the average energy consumption under delay constraints, a dynamic task offloading algorithm based on Lyapunov optimization is proposed in this thesis. The optimization problem is decoupled into task offloading decision sub problem and power optimization sub problem. The task offloading decision sub problem is solved by a fair matching multi-arm bandit based algorithm, and the power optimization sub problem is solved by the Difference of Convex function algorithm. The simulation results show that the proposed algorithm can effectively avoid resource competition and ensure the fairness between users.

Keywords: Mobile Edge Computing; Internet of vehicles; task offloading; convex optimization; reinforcement learning

目 录

摘要.....	I
目录.....	V
图清单.....	IX
表清单.....	X
变量注释表	XI
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 本文的研究内容及章节安排.....	5
2 相关理论及技术	7
2.1 移动边缘计算技术.....	7
2.2 强化学习理论.....	11
2.3 本章小结.....	14
3 基于通信与计算资源联合优化的 VEC 网络资源分配	15
3.1 引言.....	15
3.2 系统模型.....	15
3.3 基于凸差规划的通信与计算资源联合优化算法.....	20
3.4 仿真结果与分析.....	25
3.5 本章小结.....	30
4 基于多臂赌博机算法的动态 VEC 任务卸载	31
4.1 引言.....	31
4.2 系统模型.....	32
4.3 基于多臂赌博机的分布式在线学习任务卸载算法.....	35
4.4 仿真结果与分析.....	40
4.5 本章小结.....	44
5 基于公平匹配 MAB 的动态 VEC 任务卸载	45
5.1 引言.....	45
5.2 系统模型.....	45
5.3 基于公平匹配 MAB 的动态任务卸载算法.....	48
5.4 仿真结果与分析.....	55
5.5 本章小结.....	59

6 总结与展望	60
6.1 总结.....	60
6.2 展望.....	61
参考文献	62
作者简介	68
学位论文原创性声明	69
学位论文数据集	70

Contents

Abstract.....	I
Contents.....	V
List of Figures.....	IX
List of Tables.....	X
List of Variables.....	XI
1 Introduction.....	1
1.1 Research Background and Significance	1
1.2 Research Actuality	2
1.3 Research Contents and Organization of This Thesis	4
2 Related Theory and Technology.....	7
2.1 Mobile Edge Computing Technology.....	7
2.2 Reinforcement Learning Theory.....	11
2.4 Conclusions of the Chapter.....	14
3 Resource Allocation of Vehicular Edge Computing Network based on Joint Optimization of Communication and Computation Resources	15
3.1 Introduction.....	15
3.2 System Model	15
3.3 Joint Communication and Computation Optimization Algorithm based on the Difference of Convex Function Programming	20
3.4 Simulation Results and Analysis	26
3.5 Conclusions of the Chapter.....	30
4 Task Offloading of Dynamic Vehicular Edge Computing Network based on Multi-armed Bandit Algorithm	31
4.1 Introduction.....	31
4.2 System Model	32
4.3 Multi-armed Bandit based Distributed online learning algorithm	35
4.4 Simulation Results and Analysis	40
4.5 Conclusions of the Chapter.....	44
5 Task Offloading of Dynamic Vehicular Edge Computing Network based on Fair	

Matching MAB.....	45
5.1 Introduction.....	45
5.2 System Model	45
5.3 Fair Matching MAB based Dynamic Task Offloading Algorithm.....	48
5.4 Simulation Results and Analysis	55
5.5 Conclusions of the Chapter.....	59
6 Conclusions and Prospects.....	60
6.1 Conclusions.....	60
6.2 Prospects.....	61
References.....	62
Author's Resume.....	68
Declaration of Thesis Originality.....	69
Thesis/Dissertation Data Collection.....	70

图清单

图序号	图名称	页码
图 2-1	移动边缘计算系统模型	8
Figure 2-1	System Architecture of Mobile Edge Computing	8
图 2-2	两种任务卸载方式	10
Figure 2-2	Two Task Offloading Methods	10
图 2-3	强化学习基本模型	11
Figure 2-3	Architecture of Reinforcement Learning	11
图 3-1	多用户单边缘服务器 VEC 场景	16
Figure 3-1	Scenario of VEC with Multi-Users and Single Edge Server	16
图 3-2	部分任务卸载示意图	17
Figure 3-2	Diagram of Partial Task Offloading	17
图 3-3	时延约束示意图	18
Figure 3-3	Diagram of Delay Constraints	18
图 3-4	不同任务量下的能耗	27
Figure 3-4	Energy Consumption for Different Task Size	27
图 3-5	不同时延约束下的能耗	27
Figure 3-5	Energy Consumption for Different Delay Constraints	27
图 3-6	不同时延约束下各用户被分配到的计算资源	28
Figure 3-6	Allocated Computation Resource for Different Delay Constraints	28
图 3-7	不同时延约束下各用户的任务卸载比例	29
Figure 3-7	Task Offloading Ratio for Different Delay Constraints	29
图 3-8	不同时延约束下的各用户的 CPU 频率	29
Figure 3-8	CPU Frequency for Different Delay Constraints	29
图 4-1	车辆边缘计算网络车辆间任务卸载模型图	32
Figure 4-1	Architecture of V2V Task Offloading Model in VEC Network	32
图 4-2	车辆间任务卸载流程图	33
Figure 4-2	Diagram of V2V Task Offloading	33
图 4-3	各算法的平均时延	42
Figure 4-3	Average Delay for Each Algorithm	42
图 4-4	各算法的累积遗憾	42
Figure 4-4	Cumulated Regret of Each Algorithm	42
图 4-5	各算法的最佳决策率	43
Figure 4-5	Best Decision Ratio of Each Algorithm	43
图 4-6	不同机制下 MDOLTO 算法的平均时延	43
Figure 4-6	The Average Delay of MDOLTO with Different Mechanism	43
图 4-7	不同权重参数下 MDOLTO 算法的性能	44
Figure 4-7	The Performance of MDOLTO with Different Weight Factor	44
图 5-1	车辆边缘计算网络多用户任务卸载场景	46
Figure 5-1	Task Offloading Scenario of Multi-user VEC Network	46

图 5-2	各算法的平均时延	57
Figure 5-2	Average Delay of Each Algorithm	57
图 5-3	各算法的平均能耗	57
Figure 5-3	Average Energy Consumption of Each Algorithm	57
图 5-4	不同时延约束下的平均能耗	58
Figure 5-4	Average Energy Consumption with Different Energy Consumption	58
图 5-5	不同控制权重参数下的平均时延	58
Figure 5-5	Average Delay with Different Weight Parameter	58
图 5-6	不同控制权重参数下的平均能耗	59
Figure 5-6	Average Energy Consumption with Different Weight Parameter	59

表清单

表序号	表名称	页码
表 2-1	移动云计算与移动边缘计算对比	9
Table 2-1	Comparison between MCC and MEC	9
表 3-1	多用户 VEC 系统参数配置表	26
Table 3-1	Parameters Assignment Table of Multi-User VEC System	26
表 3-2	用户参数配置表	27
Table 3-2	System Parameters Assignment Table of Users	27
表 4-1	动态 VEC 系统参数配置表	40
Table 4-1	Parameters Assignment Table of Dynamic VEC System	40
表 5-1	多用户动态 VEC 系统参数配置表	56
Table 5-1	Parameters Assignment Table of Multi-user Dynamic VEC System	56

变量注释表

φ	计算任务
L	任务量 (bits)
C	计算复杂度 (cycles/bit)
T	时延约束 (s)
r	数据率 (bps)
B	带宽 (Hz)
P	信号发射功率 (w)
σ	高斯白噪声功率 (w)
E	能耗 (J)
Q, λ	任务卸载比例
f	计算单元工作频率 (Hz)
V	用户分得的计算资源
N	提供计算服务的车辆集合
d	完成任务的时延 (s)
a	用户的卸载决策
u	服务车辆的服务能力 (bit/s)
η	权重参数
$\beta_{m,n}(t)$	二元任务卸载变量
$\tilde{\rho}$	二元碰撞指示变量

1 绪论

1 Introduction

1.1 研究背景及意义 (Research Background and Significance)

随着物联网技术与通信技术的快速发展, 人们生活的智能化水平逐步提高。为了应对车辆增多带来的交通拥堵、交通事故频发等问题, 智能交通系统 (Intelligent Transport System, ITS) 已成为了热门的研究领域^[1-3], 而车联网 (Internet of Vehicles, IOV) 作为 ITS 的核心技术之一, 同样也得到了广泛的研究^[4-6]。车辆通过部署雷达、传感器等感知设备对交通环境信息进行采集, 再按照一定的通信协议与其他车辆、人、基础设施进行无线通信与信息交互, 构成了车联网系统。通过对网络内的信息数据进行分析处理, 就能实现对人、车、路的智能调度与管理。近年来, 随着智能汽车的发展, 出现了一系列新型的车联网应用, 如自动驾驶、辅助驾驶、虚拟现实 (Virtual Reality, VR) 与增强现实 (Augmented Reality, AR) 等^[7-9]。这些应用通常包含大量计算密集型的计算任务, 而且对时延也十分敏感, 因此需要较多的计算资源与通信资源来保障其服务质量 (Quality of Service, QoS)。然而, 车载处理器受限于体积、重量与电池寿命, 往往计算资源有限, 因此无法满足上述应用的需求^[10]。

为了解决上述问题, 基于云计算的解决方案通过将任务卸载到云服务器端进行处理^[11-13], 云服务器汇集了大量的计算资源, 能够满足车辆的计算需求。然而大量计算任务的上传导致了网络数据量的急剧增加, 致使核心网拥塞, 加重了网络负担。此外, 车辆距离云服务中心通常较远, 远距离的信息传输也会带来较大的时延损耗, 而车联网中的新兴应用对时延有着严苛的要求。因此对于时延敏感型任务来说, 云计算通常难以满足其 QoS 需求。

为了应对网络中数据流量的爆炸式增长与一些新兴应用极低时延需求, 移动边缘计算 (Mobile edge computing, MEC) 将远端云服务下放到网络的边缘端, 拉近了服务器与用户的物理距离, 既减小了任务与网络传输的时延, 又减轻了核心网络的数据压力^[14-16]。通过任务卸载技术, 用户能够将一部分或全部任务卸载到 MEC 服务器, 以降低完成任务的时延, 同时减少用户自身能耗^[17]。将移动边缘计算与车联网相结合, 车辆边缘计算网络^[18-20] (Vehicular edge computing, VEC) 有助于解决车联网中计算密集型任务的时延敏感问题。在车辆边缘计算网络中, 边缘服务器通常部署在路边单元 (Road Side Unit, RSU) 侧, 车辆通过车-路通信 (Vehicle to Infrastructure, V2I) 的方式与其通信。由于边缘服务器与车辆的物理距离较近, 传输任务的时延与损耗将大大减少, 与此同时, 边缘服务器拥有的计

算与存储资源远大于车辆自身拥有的资源,因此能够帮助车辆处理计算密集型与时延敏感型的车联网应用。

然而,相较于集中式的云服务器,VEC 服务器的资源是有限的。在交通出行高峰,用户数量较多的情况下,VEC 服务器难以满足所有车辆的任务卸载需求。此外,由于车辆具有高速移动的特性,VEC 的网络拓扑结构时刻处于动态变化的状态,从而导致频繁的通信链路断裂、基站切换的情况,对网络的规划与管理造成了较大的困难。因此,针对车辆的移动特性,如何规划车联网边缘计算系统中的有限资源,利用网络中的空闲计算资源提高系统的资源使用效率,并提出有效的管理方案具有重要的理论研究价值及实际意义。针对车联网边缘计算系统网络资源受限的问题,本文研究了一种基于通信与计算资源联合优化的资源分配方案。为了应对车联网边缘计算系统动态网络环境对任务卸载造成的影响,本文研究了基于强化学习的分布式在线学习任务卸载策略。

1.2 国内外研究现状 (Research Actuality)

移动边缘计算能够在网络的边缘,即用户附近为其提供计算服务,从而降低任务时延,使用户获得更好、更安全的网络体验。基于上述特点,在车联网中引入 MEC,车辆可以通过任务卸载来满足计算密集型与时延敏感型应用的 QoS 需求。然而,MEC 服务器的资源也是有限的,无法负担网络中所有用户的任务卸载请求。此外,车辆的移动性使车联网的网络拓扑结构处于快速变化的状态,给任务卸载决策造成了较大困难。本文主要研究车辆边缘计算网络中的资源分配与任务卸载问题。因此,本小节将分别从 MEC 与 VEC 的资源分配与任务卸载决策问题这两方面来概述目前学术界的研究现状。

1.2.1 MEC 资源分配与任务卸载研究

作为 MEC 的关键技术之一,任务卸载使网络中的用户终端能够将计算密集的任务上传至计算资源更丰富的 MEC 服务器上执行,从而帮助计算资源与能量有限的用户以低时延与低能耗的代价完成计算任务。然而由于成本与体积的限制,MEC 服务器相较于云服务器来说是资源受限的。随着接入网络设备数量的增加,网络数据爆炸式增长,MEC 的计算资源难以满足网络中的计算需求。因此,如何在 MEC 网络中设计高效的资源分配与计算卸载方案以提高整个系统的资源利用效率,成为了国内外学者广泛研究的问题。

评价任务卸载方案的主要性能指标包括完成计算任务的时延与能耗。计算任务的时延一般包括任务上行链路的传输时延与 MEC 服务器处理任务的计算时延两部分。而完成计算任务的能耗一般指用户进行任务卸载而产生的上行链路传输能耗与运行计算任务产生的能耗。在现有的国内外研究中,许多研究者围绕能耗

优化^[21-25]、时延优化^[26-29]、以及能耗与时延折中^[30-33]分别展开了研究。

为了降低系统的能耗,文献[21]中,Hao 等人考虑了 MEC 系统中的任务缓存问题,提出了一个联合任务缓存与任务卸载的优化方案。文献[22]中,Chen 等人将动态电压调整技术(Dynamic Voltage Scaling, DVS)引入了任务卸载策略中,设备可以根据任务负载与时延约束动态调整 CPU 工作频率,从而达到最优的能量效率。You 等^[23]分别考虑了基于时分多址技术(Time Division Multiple Access, TDMA)与正交频分多址技术(Orthogonal Frequency Division Multiple Access, OFDMA)的多用户 MEC 系统,并在给定时延约束下优化了系统能耗。为了提高系统的能量效率,Wang 等^[24]将无线能量传输技术(Wireless Power Transfer, WPT)引入 MEC 系统中,设计了一个由多天线接入点供电的多用户 WPT-MEC 系统,通过联合优化波束成形的传输能量, CPU 工作频率以及卸载决策,使系统的总能耗达到最优。文献[25]提出了一种基于拉格朗日对偶分解的快速收敛算法,优化了多用户 MEC 系统种的用户总能耗。

为了降低任务的时延,Dai 等^[26]考虑了用户与 MEC 服务器间的负载平衡,提出了一种启发式的算法。文献[27]研究了资源有限情况下的时延最小化问题,并提出了一个有性能保障的多项式时间近似解。文献[28]中,Luo 等人基于马尔可夫决策理论(Markov Decision Process, MDP),根据信道状态与任务缓冲区队列设计了一个任务调度策略,优化了执行任务的时延。Soua 等^[29]考虑到突发任务到达的情况,利用排队论将服务器调度与上下行链路传输调度进行结合,从而最小化任务的平均时延。

文献[30]-[33]综合考虑了时延与能耗的优化。Kwak 等^[30]综合考虑了 MEC 系统中的本地负载、服务器负载与网络流量的负载,提出了一种基于李雅普诺夫优化的算法综合优化了系统完成任务的时延与能耗。为了优化多用户 MEC 系统的时延与能耗,Shi 等^[31]使用了可分离半定松弛法对通信资源分配策略与卸载决策进行了联合优化。Zhang 等^[32]在文献[31]的基础上加入了对计算资源分配策略的优化,进一步提升了算法的性能。Deng 等^[33]将信道选择与卸载决策建模成了一个 NP 难的整数优化问题,并提出了一种基于博弈论的分布式算法,实现了纳什均衡,优化了系统的总能耗与总时延。

从上述文献可以看出,目前已有许多学者针对 MEC 的任务卸载机制展开了研究,研究结果也充分证明了 MEC 系统中的用户能够通过任务卸载来减少完成任务的时延与能耗。因此,针对车联网中计算密集型与时延敏感型应用,可以考虑引入 MEC,提高车联网中的用户体验。下一小节将会介绍相关研究介绍。

1.2.2 VEC 资源分配与任务卸载研究

车联网的发展使车辆的智能化越来越高,而随着智能汽车的普及,出现了各

种新兴的车联网应用。这类应用服务通常在时延方面有严格的要求,而且会产生海量的数据,给车联网带来了巨大的挑战。移动边缘计算将丰富的计算与存储资源从云端迁移至网络的边缘,从而能够有效的减少用户完成任务的时延与能耗。因此,不少学者研究了如何将 MEC 技术应用于车联网中^[34-44]。

任务卸载作为减少处理任务时延与能耗的关键技术,得到了广泛的研究。Du 等^[34]同时考虑了车辆与 VEC 服务器的收益,提出了以最小化时延与能耗为目标的双边优化方法,综合优化了卸载决策与无线资源分配策略。Tareq 等^[35]考虑了一个多服务器的场景,提出了一种基于匹配理论的算法,通过对系统的计算资源与无线带宽资源分配策略进行优化,减少了任务时延。文献[36]提出了一种基于软件定义网络(Software-defined Network, SDN)的负载均衡任务卸载方案,优化了车辆计算任务的处理延迟。Sun 等^[37]考虑同时减少车辆与服务器的能耗,提出了一个基于单亲遗传算法(Partheno Genetic Algorithm, PGO)的混合智能优化算法,实现了最佳的任务卸载策略。为了提高 VEC 服务器的资源利用率,Zhang 等^[38]提出了一种基于契约理论的高效任务卸载策略。

与云服务器相比,边缘服务器的计算与存储资源都是有限的,在交通高峰期难以满足车联网的全部计算服务需求。因此,VEC 资源优化与分配成了一项具有挑战性的任务。针对 VEC 资源受限问题,文献^[39]提出了一个用户-边缘服务器-云服务器的三层框架,并提出了一种基于 Stackelberg 博弈的多层任务卸载方案。为了提高 VEC 的资源利用率,有些学者研究了如何利用网络中未充分利用的资源。Huang 等^[40]提出利用停泊车辆的资源,使其与 VEC 服务器协调运行,共同提供计算服务。为了提高停泊车辆共享资源的积极性,Li 等^[41]设计了一个激励机制,对提供计算资源的停泊车辆给与奖励。

除了资源受限外,车辆的移动性也给基于边缘计算的车联网带来了巨大挑战。车辆快速移动造成了网络拓扑与信道环境的频繁变化,复杂的通信环境对任务卸载决策与资源规划都造成了巨大影响。为此,有不少学者针对车辆的移动性展开了研究^[42-44]。针对车辆在服务未完成时就已驶离 VEC 服务器通信范围的问题,Zhang 等^[42]针对不同类型的计算任务,设计了一种基于行动预测的最优任务卸载方案,提高了任务的传输效率并满足了任务的时延要求。文献[43]考虑到车辆的移动性导致的卸载中断问题,将卸载任务划分成多个子单元,并根据车辆移动速度、服务器覆盖范围以及传输速率等实际约束条件,推导出相应的卸载比例,使任务卸载方案能够适应车速及信道环境的变化。Fan 等^[44]在多 VEC 服务器协作的场景下,提出了一种基于移动性感知的任务卸载方案,使车辆在满足任务时延约束的前提下有效降低了系统的开销。

1.3 本文的研究内容及章节安排 (Research Contents and Organization of This Thesis)

通过上述对 VEC 网络的资源分配与任务卸载研究现状分析可以发现,为了给车联网中的计算密集型应用与时延敏感型应用提供更好的服务质量,开发网络中的闲置资源并设计一个高效的资源分配方案,以及在动态变化的网络环境下提出一个高效的任务卸载方案是十分必要的。本文在不同 VEC 场景下针对上述问题提出了解决方案,主要的研究内容如下:

(1) 在交通高峰,车辆拥堵的 VEC 场景下,边缘服务器服务的用户增多,因此资源受限。为了充分利用网络中的有限资源,保障车辆的用户体验,本文设计了一个对车辆用户与边缘服务器的通信与计算资源进行联合优化的资源分配方案,在满足车辆任务时延约束的条件下对能耗进行了优化,其中能耗包括任务卸载产生的通信能耗与本地计算产生的计算能耗。该资源分配方案综合考虑了信道状况、车辆自身的计算能力以及任务的时延要求,对信号发射功率、车载计算单元工作频率以及边缘服务器分配的計算资源进行联合优化。该问题最终建模成了一个多优化变量耦合的非凸问题。为了解决这个问题,本文首先通过凸差 (Difference of Convex, DC) 规划将其解耦成两个凸的子问题,然后再利用拉格朗日对偶分解法对子问题进行联合迭代求解,得到了问题的最优解。仿真结果表明,该算法能够在保障任务时延约束的情况下减少用户的能耗。

(2) 在车辆高速移动的 VEC 场景下,网络拓扑、无线信道状态与服务器计算负载快速变化,动态变化的网络环境给任务卸载决策的制定带来了挑战。为此,本文提出了一种基于强化学习的分布式在线学习任务卸载方案,使车辆能够在进行任务卸载的同时学习到周围网络环境的相关信息,从而不断地优化自身卸载策略。为了减轻边缘服务器的压力,充分利用 VEC 中的空闲资源,本文基于车辆即是资源 (Vehicle as a Resource, VaaR) 的思想^[45],将有空闲计算资源的车辆作为计算服务的提供者,考虑车辆间的任务卸载问题。为了增强车辆对动态网络环境的适应性,本文提出了基于多臂赌博机的分布式在线学习任务卸载算法,同时设计了基于路边单元辅助的用户协作机制,使用户可以共享彼此学习到的信息。理论分析表明,本文所提算法具有对数形式的遗憾函数。仿真结果表明,本文所提算法与其他现有算法相比,更能适应动态的车辆边缘计算网络环境。

(3) 在第 2 点的基础上,为了应对分布式任务卸载策略可能带来的资源竞争与用户公平性难以保障的问题,本文提出了一种基于公平匹配多臂赌博机算法的动态任务卸载方案。该方案以满足用户的 QoS 需求为目标对用户与计算服务提供者进行匹配,从而避免了资源竞争的情况,满足了网络内所有用户的 QoS 需求。优化问题建模为时延约束下的平均能耗最小化问题,本文提出了一种基于李

雅普诺夫优化的动态任务卸载算法进行求解。该算法首先通过李雅普诺夫理论,将多时隙的优化问题转化为单时隙的优化问题,然后将优化问题结构为任务卸载决策子问题与功率优化子问题,并分别通过基于公平匹配多臂赌博机的任务卸载算法与基于凸差规划的优化算法进行求解。仿真结果表明,本文所提算法能够有效地避免资源竞争的情况,同时保障用户间的公平性。

本文的组织结构如下:

第一章为绪论,主要介绍了车联网边缘计算的研究背景与研究意义,然后总结了国内外关于 MEC 以及 VEC 的研究现状,最后阐明了本文的主要研究内容。

第二章介绍了本论文工作的相关理论与技术知识,主要从移动边缘计算与强化学习理论这两个方面进行介绍。首先介绍了移动边缘计算的基本框架、优势以及任务卸载技术的主要流程,然后对本文主要使用的数学优化方法——强化学习进行了介绍。

第三章研究了资源受限的 VEC 中的资源分配问题。首先对系统内用户的能耗与时延建立了数学模型,然后结合动态电压调整技术,提出了一种通信与计算资源联合优化的算法使系统能耗最小化,最后通过数值仿真验证了算法的有效性。

第四章研究了动态 VEC 网络环境下的任务卸载问题。首先给出了车辆无法准确获悉网络状态的动态网络系统模型,然后将车辆的任务卸载问题抽象成了一个多臂赌博机的问题。最后提出了基于多臂赌博机的分布式在线学习算法优化了用户的平均时延。

第五章研究了动态 VEC 网络环境下的分布式任务卸载中可能存在的用户竞争现象与用户公平性难以保证的问题。首先建立了动态网络环境下的任务卸载与资源竞争模型,然后将任务卸载问题建模为一个多臂赌博机问题,并在此基础上以用户时延约束为 QoS 需求,建立了用户与计算服务提供者之间的匹配。最后提出了一种基于李雅普诺夫优化的动态任务卸载算法,在优化用户平均能耗的同时有效地避免资源竞争的情况,并保障用户间的公平性。

第六章为总结与展望,对本文的主要工作内容与贡献进行了总结与归纳,并对后续的相关研究工作进行了展望。

2 相关理论及技术

2 Related Theory and Technology

本章主要对移动边缘计算的基本理论与框架以及本文后续工作会涉及到的强化学习理论基础进行介绍。

2.1 移动边缘计算技术 (Mobile Edge Computing Technology)

2.1.1 移动边缘计算概述

随着 5G 通信技术的发展与智能设备的普及,新型移动应用如 VR/AR、超高清视频传输、自动驾驶等正逐步走进人们的生活。这类应用通常是计算密集型与时延敏感型的,计算资源有限的移动智能设备难以满足其 QoS 需求。为了解决算力问题,移动云计算技术 (Mobile Cloud Computing, MCC) 通过高速率和高可靠性的空中接口在远程云计算中心为移动设备提供计算服务^[46-50]。然而,云计算中心与移动设备间较远的物理距离使通信传输时延增加,指数级增长的移动数据量也给回程网络带来了巨大的负担。

传统的 MCC 技术已无法满足移动终端计算密集型与时延敏感型应用的 QoS 需求,为此,研究者将云计算的概念集成到移动网络中,提出了移动边缘计算网络。MEC 的概念最初由欧洲电信标准协会 (European Telecommunications Standard Institute, ETSI) 在 2014 年提出,定义为一个在接近移动用户的无线网络中提供用户所需服务与云计算功能的网络架构^[51]。MEC 通过集成了网络功能虚拟化技术 (Network Functions Virtualization, NFV)^[52]、信息中心网络技术 (Information-centric Networks, ICN)^[53]与软件定义网络 (Software-defined networks, SDN)^[54]技术的虚拟平台实现。具体来说,单个边缘设备可以通过 NFV 技术创建多个虚拟机同时执行不同的操作或任务,从而为多个不同的移动设备提供计算服务。而 ICN 技术为 MEC 提供了另一种端到端的服务识别范式,将传统的以主机为中心的模式转变成为以信息为中心的模式,从而提供基于情境感知的计算服务。SDN 技术能够让 MEC 管理员通过功能抽象来管理业务,从而实现可扩展的动态计算。在实现上,MEC 服务器部署在蜂窝网络中的基站附近,为网络边缘端的移动用户提供计算、存储服务。此外,智能设备的飞速发展使其所具备的计算能力与日俱增,为了利用这部分设备的剩余资源,MEC 服务器也可以由异构的网络单元组成。在结构上,MEC 系统如图 2-1 所示,主要由三层体系结构组成:用户层,边缘服务器层与云计算层。用户层由一些移动终端设备构成,这类设备具有相对有限的计算能力,可以处理一些简单的任务,但无法满足计算密集型与时延敏感型任务的 QoS 需求。因此,设备将此类任务卸载至边缘服务器层进行运算,从

而在时延与能耗方面获得更好的性能。云计算层部署了大量的云计算服务器，相比于边缘服务器具备更多的计算与存储资源。然而云计算层与边缘设备的距离较远，传输时延较大，因此可以完成一些对时延要求不高的大型应用，如大数据挖掘与机器模型的训练等。

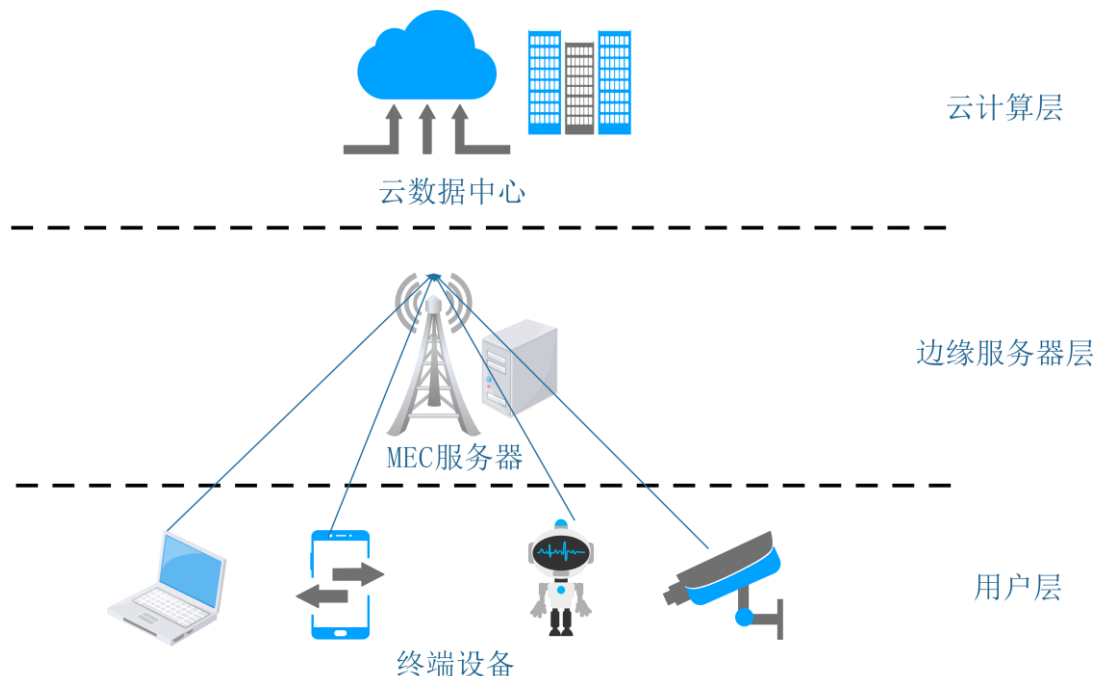


图 2-1 移动边缘计算系统模型

Figure 2-1 System Architecture of Mobile Edge Computing

MEC 作为 MCC 的补充与拓展，在与终端用户的距离、计算服务能力、任务时延方面都有较大的差别，具体如表 2-1 所示。相比之下，MEC 在能耗、时延、情景感知以及隐私安全性等方面都存在一定的优势。具体来说，在能耗方面，MEC 通过任务卸载技术，将计算密集型任务从移动终端设备转移至边缘服务器，从而能够降低设备的能量消耗。已有的研究表明，任务卸载技术能够使多媒体应用程序完成多达 44 倍的计算负载^[55]，或延长 AR 应用程序高达 50% 的使用时间^[56]。在时延方面，MCC 与移动终端设备的物理距离远，因此传播时延较大。此外，MCC 的信息需要通过包括无线接入网，回程网络在内的多个网络，网络中的流量控制与路由规划等网络管理操作也会带来一定的时延。相比之下，MEC 服务器传输距离短，协议简单，因此能够满足时延敏感型应用的需求。情景感知指 MEC 服务器在移动终端设备的附近提供计算服务时，通过获取到的设备位置、环境等用户信息，来为用户提供符合其兴趣偏好的内容或服务。这类用户信息也可以用来进行流量监测与路由调度，从而提高 MEC 网络的效率。在隐私保护方面，MCC 系统采用的是集中式的大型远程公共数据中心，用户信息资源高度集中，因此容易遭受攻击。而 MEC 系统采用分布式部署的模式，信息较为分散，不易成为攻击的目标。此外，MEC 系统可以将敏感的信息保留在本地处理，不

进行上传，从而减少信息泄露的风险。

表 2-1 移动云计算与移动边缘计算对比
Table 2-1 Comparison between MCC and MEC

	移动云计算	移动边缘计算
部署方式	集中式	分布式
与用户的距离	远	近
时延	较高	低
抖动	较大	小
计算资源	多	较少
存储资源	多	较少
隐私安全性	较低	高

2.1.2 任务卸载技术

任务卸载是 MEC 的重点研究技术^[57-59]，通过将计算密集型的任务卸载至 MEC 服务器，设备延长了电池寿命，减少了任务时延。任务卸载的流程具体可分为六个步骤：

1) 节点感知：当用户需要进行任务卸载时，会首先对周围的网络环境与能够提供计算服务的设备进行感知。用户通过感知信道状态与服务器的计算负载等信息，来进行接下来的卸载决策。

2) 卸载决策：卸载决策是任务卸载的核心步骤，用户根据自身的情况与感知到的外界信息，来决定是否进行任务卸载，以及任务卸载的目标。现已有许多研究以用户的时延或能耗为优化目标，提出相应的任务卸载策略，以提高系统的性能。

3) 任务分割：任务分割指的是将计算任务划分为多个子任务，放置在不同的设备上并行运算，这样有利于减少任务的时延。比如说，AR 应用中的图像渲染任务需要在本地进行运算，而目标识别任务则可以卸载到边缘服务器进行运算^[53]。然而并不是所有的任务都能够进行分割，而且在任务分割的过程中需要考虑到各个子任务的相关性。

4) 任务卸载：用户根据自身做出的卸载决策，将需要卸载的计算任务通过无线通信链路传输至服务器。研究者通常关注这一步骤中的上传速率，主要与信道状态与用户的信号发射功率有关。

5) 服务器执行计算：服务器接收到用户卸载的计算任务之后，通过网络功能虚拟化技术为该任务创建一个虚拟机，并分配相应的计算资源对该任务进行计算。

6) 计算结果返回：服务器在完成计算任务后，将计算结果通过无线通信链路回传给用户，这是任务卸载的最终步骤。一般情况下，计算结果远小于计算任

务量，因此计算结果通过下行链路返回时消耗的能量与时延通常被忽略。

在卸载决策中，根据任务类型的不同，任务卸载分为两种不同的模式，如图 2-2 所示，分别为二进制任务卸载与部分任务卸载，介绍如下：

1) 二进制任务卸载：当计算任务是高度集成的，无法分解时，计算任务就需要被当作一个整体在本地进行运算或卸载到 MEC 服务器进行运算，由于研究中通常用一个二进制符号来表示任务卸载决策，即“0”表示本地运算，“1”表示 MEC 服务器运算，因此这种任务卸载方式又被称为 0-1 任务卸载。

2) 部分任务卸载：当计算任务由多个模块组成时，就可以考虑将任务进行细颗粒度地划分为多个子任务，分别在本地与 MEC 服务器端并行处理，这种任务卸载方式称为部分任务卸载。相比较而言，部分任务卸载比二进制卸载更为灵活，被划分的多个子任务能够在不同的计算平台进行并行运算，从而进一步缩短任务的时延。在具体实践中，采用何种任务卸载方式主要取决于计算任务的类型。

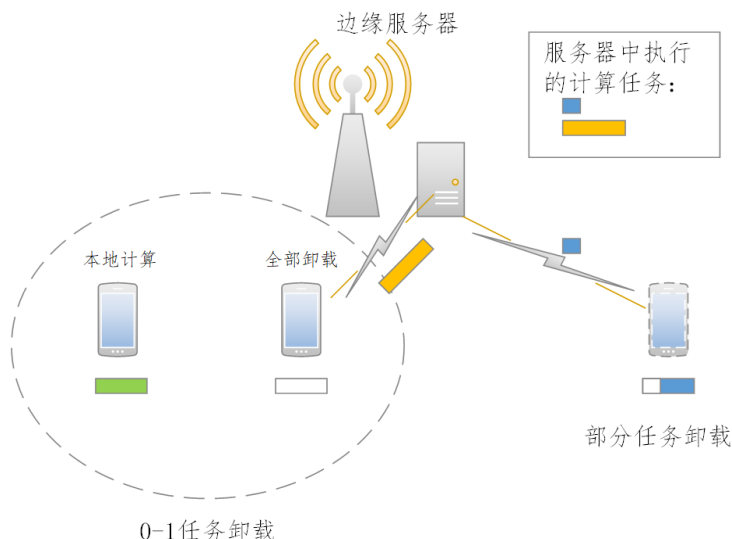


图 2-2 两种任务卸载方式

Figure 2-2 Two Task Offloading Methods

任务卸载过程会受到通信带宽、信道状态、移动设备性能、服务器可分配资源量等诸多因素的影响，因此合理任务卸载策略的设计是该领域的热门研究方向。在研究中，设计一个合理任务卸载决策的优化目标主要有三个：减少完成计算任务的能耗、减少任务时延以及在能耗与时延优化间达成一个均衡。任务卸载过程的时延与能耗主要来源于本地计算与 MEC 服务器计算两个方面。在本地计算中，时延与能耗都由本地设备处理计算任务产生。而在 MEC 服务器计算中，能耗主要来源于传输计算任务，而时延来源于三个方面：传输任务造成的传输时延，MEC 服务器处理计算任务的时延与计算结果回传的时延。

任务卸载决策的制定应考虑具体计算任务的需求。针对时延敏感型的应用，如自动驾驶，则需主要考虑任务时延的优化。而当设备自身电量受限时，如人体

健康状况检测设备,则应主要考虑完成计算任务能耗的优化。当处理一些对时延与能耗都有要求的应用时,则需要对时延与能耗分配一定的权重,优化两者的加权之和。

2.2 强化学习理论 (Reinforcement Learning Theory)

2.2.1 强化学习基本理论

强化学习是机器学习领域中的一个经典算法,它研究的是智能体在与动态环境交互的过程中不断试错,依据环境的反馈调整自身策略,从而最大化自身长期收益的过程^[60-63]。强化学习的基本模型如图 2-3 所示,从图中可以看出,智能体根据当前的状态 $s_i(t)$ 与已获得的回报 $r_i(t)$,更新自身策略 π ,采取动作 $a_i(t)$ 作用与环境,最后改变自身状态为 $s_i(t+1)$,得到动作 $a_i(t)$ 的回报 $r_i(t+1)$ 。智能体不断地与环境交互,从而学习到最佳的策略,最大化自身在动态环境下获得的长期累积收益。下面将介绍强化学习系统中除智能体与环境以外的三个主要组成部分:

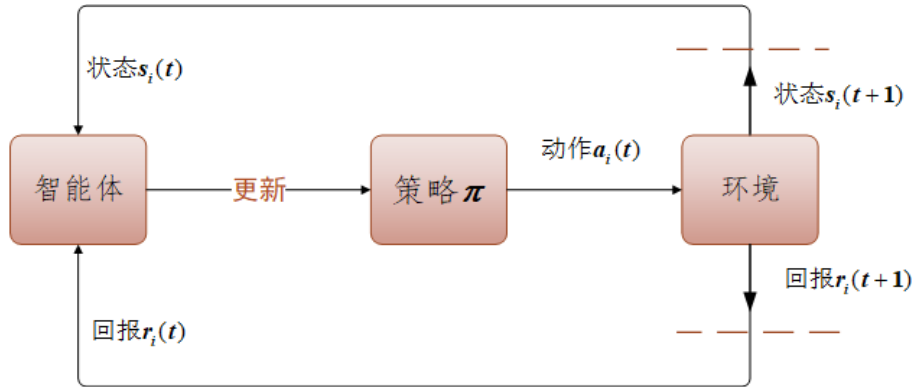


图 2-3 强化学习基本模型

Figure 2-3 Architecture of Reinforcement Learning

(1) 策略函数 π

策略函数为环境状态到动作之间的一个映射,记为 $\pi: S \rightarrow A$ 。智能体根据当前环境所处的状态,通过策略函数选择相应的动作,获得一定的收益。强化学习的目标是通过与环境交互,学习到最佳的策略,从而使长期收益最大化,因此策略函数是强化学习算法的核心。

(2) 回报函数 R

回报函数用于衡量智能体在采取动作 $a_i(t)$ 后得到的收益,即回报值 $r_i(t)$ 。 $r_i(t)$ 是一个标量,正数表示奖励,负数表示惩罚,回报函数根据动作给出相应的回报,以此为智能体指引正确的行动方向。

(3) 值函数 $V_{\pi}(s)$

值函数描述了智能体在状态 s_i 下,使用策略函数 π 所具备的长远收益趋势,具体表示为从当前状态到未来状态的加权收益之和的期望,数学表达式如下:

$$V_{\pi}(s) = E_{\pi} \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right) \quad (2-1)$$

其中 γ 表示折扣因子, $\gamma \in (0,1]$ 。

强化学习系统在 t 时刻的值函数由当前回报与未来状态产生的回报两部分组成, 表达式如下:

$$V_{\pi}(s_t) = r_t(t) + \gamma \sum_{s_{t+1} \in S} P(s_{t+1} \mid s_t, a_t) V_{\pi}(s_{t+1}) \quad (2-2)$$

2.2.2 多臂赌博机算法

在强化学习系统中, 智能体为了最大化自身收益, 往往采取两种策略: 第一种是“贪婪地”选择已知回报值最高的动作, 然而这样的策略往往不是全局最优的, 因为还有可能存在回报值更高的选项。因此, 第二种策略为探索新的选项, 利用学到的新知识不断地改进自身的策略, 从而提升自身收益。然而, 探索新选项有可能花费更大的成本, 降低智能体的收益与学习速率。因此, 如何在成本有限的情况下平衡上述两种策略是强化学习的重点研究问题, 即探索-利用 (EE, Exploration Exploitation) 问题。本节介绍的多臂赌博机问题本质上研究的就是 EE 问题。

多臂赌博机问题由 Robbins^[64]提出, 该问题的模型可表述为: 一个赌博机具有 K 个摇臂, 每个摇臂的收益为服从相关分布的随机变量, 用户选择一个摇臂下注, 获取一定的收益。起初, 用户没有任何关于摇臂收益的知识, 在不断下注后, 用户会学习到已下注摇臂的一些收益信息。若是有无限次下注机会, 用户就能学习到每个摇臂的收益分布信息, 从而做出最优的决策。多臂赌博机问题研究的是, 用户如何利用有限的下注机会最大化自身收益^[65]。此时用户需要考虑, 是根据已掌握的信息选择收益最高的摇臂, 还是探索未知的摇臂, 以期获得更高的收益。对于这两种策略的平衡就是强化学习中研究的探索-利用问题, 即 EE 问题。

对于多臂赌博机算法, 通常用累积遗憾 (Cumulated Regret) 来衡量其性能的优劣。累积遗憾表示的是已有策略带来的收益与最佳收益之间的差值, 用 $r(a_t^*)$ 表示时刻 t 的最大收益, $r(a_t)$ 表示策略得到的收益, 则累积遗憾 R_T 可表示如下:

$$R_T = E \left[\sum_{t=1}^T [r(a_t^*) - r(a_t)] \right] \quad (2-3)$$

对于多臂赌博机算法来说, R_T 随时间增长得越慢, 则该算法的性能越优。

在多臂赌博机问题中, 为了降低累积遗憾, 最大化自身收益, 用户既需要利用已知的最大回报摇臂, 又需要探索新的摇臂以获得更大的收益。然而实验次数是有限的, 因此用户需要在“探索”与“利用”间取得平衡。面对 EE 问题, 已

有研究提出了相应的解决方案，下面介绍比较经典的三种多臂赌博机算法： ε -贪婪算法、Softmax 算法与置信区间上界（Upper Bound Confidence, UCB）算法。

（1） ε -贪婪算法

ε -贪婪算法主要采取了贪婪算法的思想，即每次都选择已知收益最大的摇臂。这种纯粹的“利用”策略从长远看来无法为用户带来最大的收益，因此还需要适当地探索其余的摇臂。 ε -贪婪算法采取的方法是设定一个值 $\varepsilon \in (0,1)$ ，让用户以 ε 的概率进行“探索”，随机选择未知的摇臂，以 $1-\varepsilon$ 的概率进行“利用”，选择已知的平均收益最高的摇臂。在该算法中，摇臂 i 的平均收益 $\bar{R}_n(i)$ 如式(2-4)所示：

$$\bar{R}_n(i) = \frac{1}{n} \left[(n-1) \bar{R}_{n-1}(i) + r_n \right] \quad (2-4)$$

其中 n 表示选择摇臂 i 的次数， r_n 表示第 n 次选择该摇臂获得的收益。在 ε -贪婪算法中， ε 的值至关重要。当 ε 较大时，算法进行探索的概率会变大，这会使算法的收敛速度加快，但会降低平均收益。因为更多的探索虽然有助于寻找最佳的摇臂，但也会选择到许多收益较小的摇臂。而当 ε 较小时，平均收益会有所提高，但最终的收益较差。因为此时算法侧重于进行“利用”，寻找最优解的能力变差，从而陷入次优解。基于上述分析，应该在开始阶段进行较多的“探索”，而在后面的选择中更多侧重于“利用”，有研究^[67]基于这个思路设计了一个随时间变化的 ε 值 $\varepsilon = \frac{\log t}{t}$ ，并通过数学证明了其所设计的算法的累积遗憾上界可达到 $O(\log(t)^2)$ 。

（2）Softmax 算法

Softmax 算法又称为玻尔兹曼探索策略（Boltzmann Exploration Strategy），它通过计算摇臂的平均收益 \bar{R}_n 来平衡“探索”策略与“利用”策略。在 Softmax 算法中，每个摇臂 i 被选中的概率 $p(i)$ 如下所示：

$$p(i) = \frac{e^{\frac{\bar{R}_n(i)}{\tau}}}{\sum_{k=1}^K e^{\frac{\bar{R}_n(k)}{\tau}}} \quad (2-5)$$

其中 $\tau \in R^+$ 被称为温度变量。从上式可以看出，摇臂的平均收益越大，则被选择的概率就越大。与此同时，当 τ 取值较大时，算法偏向于“探索”策略，而当 τ 取值较小时，算法就偏向于“利用”策略。

（3）UCB 算法

上述 ε -贪婪算法与 Softmax 算法都是在某个确定性策略的基础上利用一个

参数进行“探索”策略与“利用”策略的平衡，这样的算法具有一定的随机性，算法性能十分容易受参数数值的影响。UCB 算法则采用了完全不同的方法，它通过摇臂的平均收益与置信度来评估摇臂。具体来说，算法中每个摇臂的收益都有一个相应的置信区间，摇臂被选择的次数越多，关于它的收益信息就会越明确，它的置信区间也就会越窄。算法用摇臂的平均收益与置信区间之和作为指标值，并在每次决策中选择指标值最大的摇臂。摇臂 i 的指标值 x_i 表达式如下：

$$x_i = \bar{R}_n(i) + \sqrt{\frac{2 \ln n}{n_i}} \quad (2-6)$$

其中 $\bar{R}_n(i)$ 为摇臂 i 的平均收益， n 为实验进行的次数， n_i 为该摇臂被选择的次数。指标值 x_i 体现了 UCB 算法对“探索”与“利用”策略的均衡，式中的第一项 $\bar{R}_n(i)$

表示了该摇臂的期望回报，第二项 $\sqrt{\frac{2 \ln n}{n_i}}$ 则表示了该期望值的置信度。若 $\bar{R}_n(i)$

较大，则算法会趋向于“利用”策略。而当对某一摇臂的选择次数较少时，第二项的值较大，意味着该摇臂的评估置信度较低，需要多增加对该摇臂的探索，从而获得更高的置信度。这样就很好地平衡了“探索”与“利用”策略。此外，有许多学者对置信度进行了改进，比如 UCB-Tuned 算法^[66]衡量置信度时，将摇臂回报值的方差考虑在内。当对摇臂的选择次数增多时，摇臂的回报值也会趋于稳定，方差也就随之减小，整个算法的策略也就从“探索”转为了“利用”。该算法中的指标值 x_i 表达式如下：

$$x_i = \bar{R}_n(i) + \sqrt{\frac{\ln n}{n_i} \min\left(\frac{1}{4}, (\sigma_i^2 + \sqrt{\frac{2 \ln n}{n_i}})\right)} \quad (2-7)$$

其中 σ_i^2 为摇臂 i 的回报值的方差。

2.3 本章小结（Conclusions of the Chapter）

本章首先介绍了移动边缘计算技术提出的背景、定义与优势，并对任务卸载技术做了详细介绍。然后本章对强化学习理论的基本概念进行了介绍，并对其中的经典算法进行了分析，为后续章节的研究内容提供了理论基础。

3 基于通信与计算资源联合优化的 VEC 网络资源分配

3 Resource Allocation of Vehicular Edge Computing Network based on Joint Optimization of Communication and Computation Resources

3.1 引言 (Introduction)

在车联网中,诸如无人驾驶等计算密集型与时延敏感型的应用对计算资源的消耗巨大,仅凭车辆自身难以满足任务需求。为此,研究者提出将 MEC 技术引入车联网中,构成车辆边缘计算。在 VEC 中,车辆可以将任务卸载至临近的边缘服务器,靠服务器拓展自身计算能力,从而满足任务的 QoS 需求。然而,边缘服务器自身的资源也是有限的。当交通流量较大时,接入边缘服务器的设备增多,资源受限的服务器就难以满足所有用户的计算需求。因此,如何设计资源分配方案,提高网络资源利用效率成了一个至关重要的问题。

本章针对多用户 VEC 系统,引入动态电压调节技术^[67],设计了一种基于通信与计算资源联合优化的资源分配方案。该方案根据用户任务的时延约束以及信道状况,对车载计算单元工作频率、通信发射功率、车辆的任务卸载量以及边缘服务器的计算资源分配进行联合优化。优化问题最终建模成了一个多变量耦合的非凸问题,本章采用 DC 规划将其转化为两个凸的子问题,随后采用拉格朗日对偶分解法进行联合迭代求解。最后通过仿真实验验证了算法的有效性。

3.2 系统模型 (System Model)

如图 3-1 所示,本章考虑一个多用户单边缘服务器的 VEC 应用场景。在系统模型中,每个用户通过无线信道与服务器进行连接,用户通过任务卸载技术,在服务器的帮助下完成计算任务。本章采用的是部分任务卸载方式,即计算任务会被分割成两部分,分别在用户本地与服务器端进行运算。考虑到智能交通系统中存在车辆智能调度与管理等应用需求,车辆会将本地计算的任务结果上传至服务器进行进一步的分析。

假设系统中共有 K 个用户,用户编号用 k 表示, $k = \{1, 2, \dots, K\}$ 。用户 k 的计算任务用 φ_k 表示, φ_k 由三个变量构成: $\varphi_k = \{L_k, C_{u,k}, T_k\}$, 其中 L_k 表示计算任务的大小,单位为比特, $C_{u,k}$ 表示计算任务的复杂度,指计算 1 比特任务所需的 CPU 周期数,而 T_k 表示任务的时延约束。在本章中,用户的计算任务是可分割的,被

分割的子任务可以在不同的地方进行计算。通过任务分割与任务卸载，用户与边缘服务器能够并行地执行计算任务，从而提高计算效率。用户有三种执行计算任务的方式：1.全本地计算：不进行任务卸载，将任务全部进行本地运算；2.全任务卸载：将任务全卸载至边缘服务器进行运算；3.部分任务卸载：将任务分割成两部分，由本地与服务器进行运算。其中部分任务卸载可以看作是一种广义的任务卸载策略，全本地计算与全任务卸载则是部分任务卸载的两种特殊情况。本章引入了 DVS 技术，使用户可以根据任务的时延约束自行调整 CPU 工作频率，以降低计算能耗。

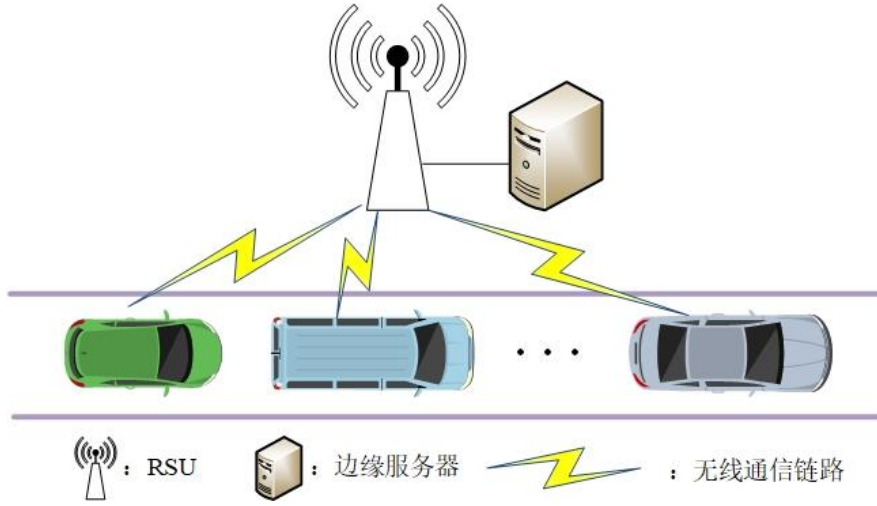


图 3-1 多用户单边缘服务器 VEC 场景

Figure 3-1 Scenario of VEC with Multi-Users and Single Edge Server

3.2.1 任务卸载模型

(1) 通信模型

本章采用部分任务卸载模式，其示意图如图 3-2 所示，其中 $Q_k \in [0,1]$ 表示用户 k 的任务卸载比例，即本地计算任务数据量占总任务数据量的比例。 $\beta \in [0,1]$ 用来描述本地计算结果的大小，表示为计算结果占计算任务大小的比例。对于一般的应用，本地计算结果无需上传，此时令 $\beta = 0$ 即可。用户 k 与服务器通信的上行链路传输速率表达式如下：

$$R_{i,k} = B \log_2 \left(1 + \frac{P_{i,k} h_k}{\sigma^2} \right) \quad \forall i \in \{1, 2\} \quad (3-1)$$

其中 $P_{i,k}$ 为用户 k 的信号发射功率， h_k 为信道增益， σ^2 为高斯白噪声的噪声功率。由于模型采用正交频分复用 (Orthogonal Frequency Division Multiplexing, OFDM) 技术^[68]，因此不考虑同信道干扰。而 $i = 1$ 指的是用户卸载任务至边缘服务器的过程， $i = 2$ 则指用户上传本地计算的任务结果的过程。

用户进行任务卸载的传输时延可表示为：

$$t_{1,k} = \frac{(1-Q_k)L_k}{R_{1,k}} \quad (3-2)$$

同样地，用户上传本地计算结果的传输时延可表示为：

$$t_{2,k} = \frac{\beta Q_k L_k}{R_{2,k}} \quad (3-3)$$

通信过程中的能耗由传输时间乘以传输功率得到，用户进行任务卸载的传输能耗表达式如下：

$$E_{comm,k}^s = \frac{(1-Q_k)L_k}{R_{1,k}} (P_{1,k} + P_{cir}) \quad (3-4)$$

其中 P_{cir} 为设备的电路损耗，是一个常数。

同理，用户上传本地计算结果的能耗可表示为：

$$E_{comm,k}^l = \frac{\beta Q_k L_k}{R_{2,k}} (P_{2,k} + P_{cir}) \quad (3-5)$$

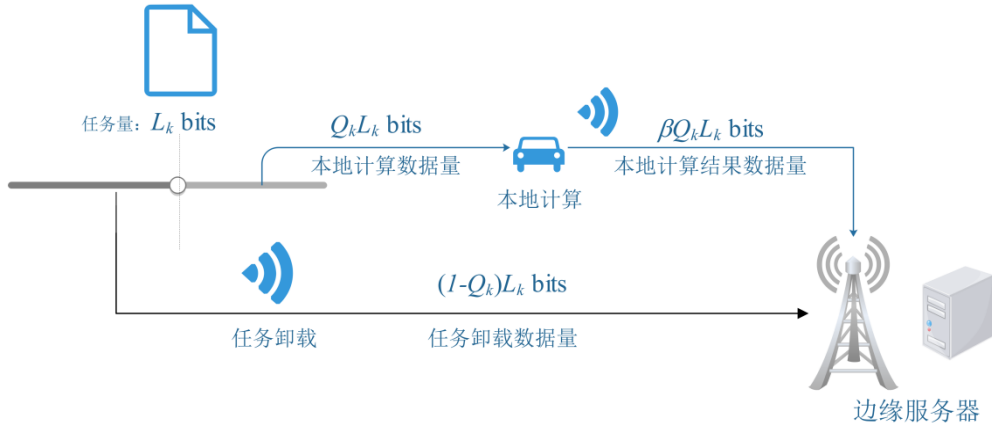


图 3-2 部分任务卸载示意图

Figure 3-2 Diagram of Partial Task Offloading

(2) 计算模型

文献[69]指出，设备芯片的能耗 E_l 与其 CPU 工作电压 V_{cir} 的平方成正比，即 $E_l \propto V_{cir}^2$ 。而文献[70]进一步指出，芯片的 CPU 工作频率 $f_{u,k}$ 与工作电压 V_{cir} 呈线性关系，即 $f_{u,k} \propto V_{cir}$ 。因此，设备运行一个 CPU 周期的能耗可表示为 $k_u f_{u,k}^2$ ，其中 k_u 为一个与芯片硬件架构有关的常数。用户 k 进行本地计算的能耗可表示如下：

$$E_{comp,k} = k_u Q_k L_k C_{u,k} f_{u,k}^2 \quad (3-6)$$

其中 $Q_k L_k$ 为运行本地计算任务所需的 CPU 周期数，将其与运行一个 CPU 周期的

能耗相乘，就能得到本地计算的能耗 $E_{comp,k}$ 。

从式(3-6)中可以看出，本地计算的能耗与车载计算单元的 CPU 工作频率的平方成正比。本章引入 DVS 技术，使用户在不违反时延约束的前提下尽量降低用户的 CPU 工作频率，进而降低用户的本地计算能耗。

用户 k 进行本地计算的时延可表示为：

$$t_{3,k} = \frac{C_{u,k} Q_k L_k}{f_{u,k}} \quad (3-7)$$

边缘服务器计算能力表示为 V_{\max} ，单位是比特/秒，由所有 MEC 用户共享。设用户 k 分到的计算资源为 V_k ，则其进行任务卸载的服务器计算时延可表示为：

$$t_{4,k} = \frac{(1-Q_k)L_k}{V_k} \quad (3-8)$$

(3) 时延约束模型

在本模型中，用户需要在规定的时延约束 T_k 内完成计算任务，且任务卸载与本地计算结果的上传无法同时进行。在这种状况下，用户完成计算任务的总时延会出现以下四种不同的情况，如图 4-3 所示：

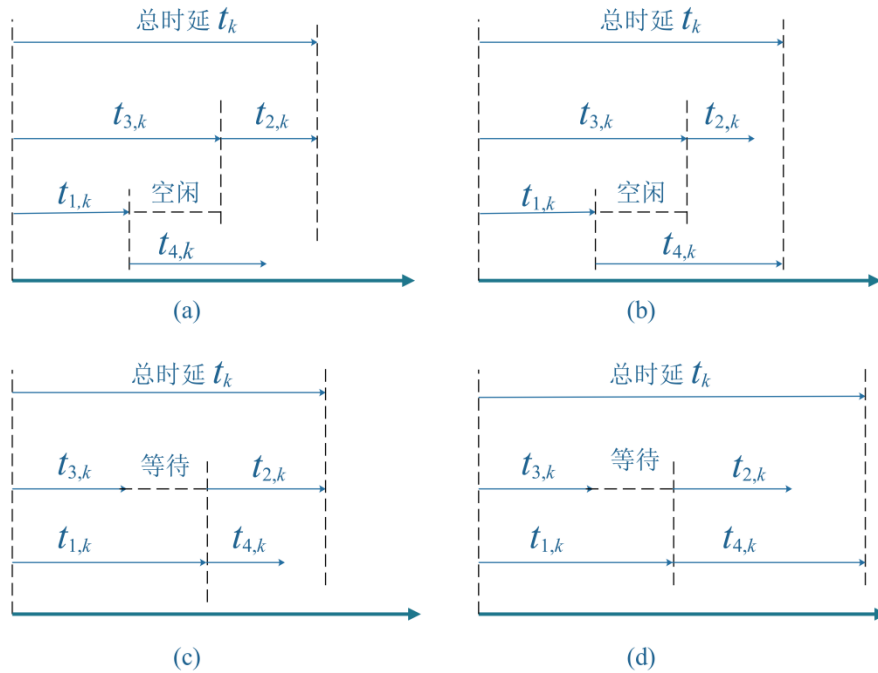


图 3-3 时延约束示意图

Figure 3-3 Diagram of Delay Constraints

这四种情况可分为两类进行讨论：

(1) 在图 3-3(a)与 3-3(b)中，本地计算时延 $t_{3,k}$ 比卸载任务的时延 $t_{1,k}$ 长，所以当用户卸载完任务后，信道将空闲一段时间。此时，用户 k 完成计算任务的总

时延可表示为 $t_k = \max \left[(t_{3,k} + t_{2,k}), (t_{1,k} + t_{4,k}) \right]$ 。

(2) 在图 3-3(c)与 3-3(d)中, 卸载任务的时延 $t_{1,k}$ 比本地计算时延 $t_{3,k}$ 长, 所以用户在本地计算完成之后, 需要等待卸载任务完成后才能上传本地计算结果。此时, 用户 k 完成计算任务的总时延可表示为 $t_k = t_{1,k} + \max(t_{2,k} + t_{4,k})$ 。

结合上述两类情况, 用户 k 完成计算任务的总时延可以用式(3-9)表示:

$$t_k = \max \left\{ t_{1,k} + t_{4,k}, \left[\max(t_{2,k}, t_{3,k}) + t_{2,k} \right] \right\} \quad (3-9)$$

则任务的时延约束可表示为:

$$\max \left\{ t_{1,k} + t_{4,k}, \left[\max(t_{2,k}, t_{3,k}) + t_{2,k} \right] \right\} \leq T_k \quad (3-10)$$

3.2.2 优化问题描述

本章的优化目标为最小化系统中所有用户的总能耗。根据 3.2.1 节中的数学模型, 优化问题表述如下:

$$[\mathbf{P1}] \quad \min_{P, Q, f_u, V} \sum_{k=1}^K (E_{comp,k} + E_{comm,k}^s + E_{comm,k}^l) \quad (3-11a)$$

$$s.t. \quad P_{i,k} \leq P_{\max} \quad (3-11b)$$

$$f_{u,k} \leq f_{\max} \quad (3-11c)$$

$$0 \leq Q_k \leq 1 \quad (3-11d)$$

$$\sum_{k=1}^K V_k \leq V_{\max} \quad (3-11e)$$

$$\max \left\{ t_{1,k} + t_{4,k}, \left[\max(t_{2,k} + t_{3,k}) + t_{2,k} \right] \right\} \leq T_k \quad (3-11f)$$

其中, 对于 $\forall i \in [1, 2]$ 与 $\forall k \in [1, 2, \dots, K]$, $\mathbf{P} = [P_{i,k}]$, $\mathbf{Q} = [Q_k]$, $\mathbf{f}_u = [f_{u,k}]$, $\mathbf{V} = [V_k]$ 。

优化问题 **P1** 中的约束(3-11b)与(3-11c)分别为传输功率约束与 CPU 工作频率约束, (3-11d)为任务卸载比例约束, (3-11e)为边缘服务器计算资源分配约束, (3-11f)为任务的时延约束。此外, **P1** 的优化目标式(3-11a)可结合式(3-4), (3-5)与(3-6)写为更详细的形式:

$$E_k = \frac{(1-Q_k)L_k}{R_{1,k}} (P_{1,k} + P_{cir}) + \frac{\beta Q_k L_k}{R_{2,k}} (P_{2,k} + P_{cir}) + k_u Q_k L_k C_{u,k} f_{u,k}^2 \quad (3-12)$$

从式(3-12)可以看出, 在优化问题 **P1** 中存在变量耦合的情况, 如 $P_{i,k}$ 与 $R_{i,k}$, Q_k 与 $f_{u,k}$ 。因此, **P1** 是一个非凸的优化问题, 难以进行求解。本章采用凸差规划

^[71]将其转化为两个凸的子问题进行求解。

3.3 基于凸差规划的通信与计算资源联合优化算法（Joint Communication and Computation Optimization Algorithm based on the Difference of Convex Function Programming）

3.3.1 算法描述

通过上一小节分析，可知优化问题 **P1** 较为复杂。因此本节首先考虑对 **P1** 进行简化，再将其解耦为两个分别针对通信资源与计算资源进行优化的子问题进行求解。

首先，文献[24]通过研究指出：为了节省计算能耗，CPU 在不同的机器周期内应保持相同的工作频率。此外，从式(3-12)可以看出，用户总能耗 E_k 与 CPU 工作频率 $f_{u,k}$ 的平方成正比。这意味着，为了减小用户的能耗， $f_{u,k}$ 应当在满足时延约束的情况下，尽可能地小。将式(3-2)，(3-3)，(3-7)与(3-8)带入时延约束式(3-11f)，可将时延约束表示为如下形式：

$$\frac{(1-Q_k)L_k}{R_{1,k}} + \frac{(1-Q_k)L_k}{V_k} \leq T_k \quad (3-13)$$

$$\frac{(1-Q_k)L_k}{R_{1,k}} + \frac{\beta Q_k L_k}{R_{2,k}} \leq T_k \quad (3-14)$$

$$\frac{\beta Q_k L_k}{R_{2,k}} + \frac{C_{u,k} Q_k L_k}{f_{u,k}} \leq T_k \quad (3-15)$$

在实际中，计算结果通常是一个很小的值^[72]，即 β 远小于 1。在这种情况下，为了简化数学分析，可以将式(3-15)的第一项忽略，因此可得：

$$\frac{C_{u,k} Q_k L_k}{f_{u,k}} \leq T_k \quad (3-16)$$

从式(3-16)可以推出 $f_{u,k}$ 的取值范围为 $f_{u,k} \geq \frac{C_{u,k} Q_k L_k}{T_k}$ 。而根据前面的分析，

CPU 工作频率 $f_{u,k}$ 应保持相同且尽可能地小，所以可以得到 $f_{u,k}$ 的最优解：

$$f_{u,k}^* = \frac{C_{u,k} Q_k L_k}{T_k} \quad (3-17)$$

此外，联合式(3-11c)与式(3-16)，可以得到一个关于卸载比例 Q_k 的约束：

$$Q_k \leq \frac{T_k f_{\max}}{C_{u,k} L_k} \quad (3-18)$$

将式(3-12), (3-17)与(3-18)带入 **P1**, 即可消除原优化问题中的优化变量 f_u 。
记化简后的优化问题为 **P1.1**, 如下所示:

$$[\mathbf{P1.1}] \quad \min_{P, Q, V} \sum_{k=1}^K (E_{comp,k} + E_{comm,k}^s + E_{comm,k}^l) \quad (3-19a)$$

$$s.t. \quad P_{i,k} \leq P_{\max} \quad (3-19b)$$

$$0 \leq Q_k \leq 1 \quad (3-19c)$$

$$Q_k \leq \frac{T_k f_{\max}}{C_{u,k} L_k} \quad (3-19d)$$

$$\sum_{k=1}^K V_k \leq V_{\max} \quad (3-19e)$$

$$\frac{(1-Q_k)L_k}{R_{1,k}} + \frac{(1-Q_k)L_k}{V_k} \leq T_k \quad (3-19f)$$

接下来将优化问题进行分解。根据文献[73]提出的分解理论, 可知:

$$\inf_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y}) = \inf_{\mathbf{x}} f^*(\mathbf{x}) \quad (3-20)$$

其中 $f^*(\mathbf{x}) = \inf_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ 。

因此, 可以考虑给定优化变量 Q 初始值 Q^0 , 先对优化问题 **P1.1** 的其余变量进行优化, 然后再对优化变量 Q 进行求解。将 Q^0 带入 **P1.1**, 得优化子问题 **P1.2**:

$$[\mathbf{P1.2}] \quad \min_{P, V} \sum_{k=1}^K \left(\frac{k_u (Q_k^0 L_k C_{u,k} L_k)^3}{T_k^2} + \frac{(1-Q_k^0)L_k}{R_{1,k}} (P_{1,k} + P_{cir}) \right) \quad (3-21a)$$

$$+ \frac{\beta Q_k^0 L_k}{R_{2,k}} (P_{2,k} + P_{cir})$$

$$s.t. \quad P_{i,k} \leq P_{\max} \quad (3-21b)$$

$$\sum_{k=1}^K V_k \leq V_{\max} \quad (3-21c)$$

$$\frac{(1-Q_k^0)L_k}{R_{1,k}} + \frac{(1-Q_k^0)L_k}{V_k} \leq T_k \quad (3-21f)$$

在 **P1.2** 中, 优化目标的第一项为常数, 因此主要考虑对第二项与第三项进

行优化。为了将该优化问题转化成凸优化问题，此处考虑引入辅助变量 $Z_{1,k}$ 与 $Z_{2,k}$ 。

根据凸优化理论^[74]， $\frac{y^2}{x}$ 的形式是凸的，而且由于 **P1.2** 为一个最小化的优化问题，

所以用 $Z_{1,k}^2$ 与 $Z_{2,k}^2$ 来表示优化目标中第二项与第三项的上限，表达式如下：

$$\frac{(1-Q_k^0)L_k}{R_{1,k}}(P_{1,k} + P_{\text{cir}}) \leq Z_{1,k}^2 \quad (3-22)$$

$$\frac{\beta Q_k^0 L_k}{R_{2,k}}(P_{2,k} + P_{\text{cir}}) \leq Z_{2,k}^2 \quad (3-23)$$

再将式(3-22)与(3-23)转化成分式的形式：

$$\frac{1}{R_{1,k}} - \frac{Z_{1,k}^2}{(1-Q_k^0)L_k(P_{1,k} + P_{\text{cir}})} \leq 0 \quad (3-24)$$

$$\frac{1}{R_{2,k}} - \frac{Z_{2,k}^2}{\beta Q_k^0 L_k(P_{2,k} + P_{\text{cir}})} \leq 0 \quad (3-25)$$

此时，不等式(3-24)与(3-25)左边项为两个凸函数相减的形式，因此式(3-24)与(3-25)为凸差约束。根据 DC 算法的思路^[71]，下一步将 DC 约束中的第二项线性化，建立凸近似的可行解。给定起始点 $Z_{1,k}^0$ ， $Z_{2,k}^0$ ， $P_{1,k}^0$ 与 $P_{2,k}^0$ ，对 DC 约束的第二项进行泰勒展开：

$$\frac{1}{R_{1,k}} - \frac{1}{(1-Q_k^0)L_k} \left(\frac{(Z_{1,k}^0)^2}{(P_{1,k}^0 + P_{\text{cir}})} + \frac{2Z_{1,k}^0(Z_{1,k} - Z_{1,k}^0)}{P_{1,k}^0 + P_{\text{cir}}} - \frac{(Z_{1,k}^0)^2(P_{1,k} - P_{1,k}^0)}{(P_{1,k}^0 + P_{\text{cir}})^2} \right) \leq 0 \quad (3-26)$$

$$\frac{1}{R_{2,k}} - \frac{1}{\beta Q_k^0 L_k} \left(\frac{(Z_{2,k}^0)^2}{(P_{2,k}^0 + P_{\text{cir}})} + \frac{2Z_{2,k}^0(Z_{2,k} - Z_{2,k}^0)}{P_{2,k}^0 + P_{\text{cir}}} - \frac{(Z_{2,k}^0)^2(P_{2,k} - P_{2,k}^0)}{(P_{2,k}^0 + P_{\text{cir}})^2} \right) \leq 0 \quad (3-27)$$

此时，式(3-26)与(3-27)为凸约束。由于 $Z_{1,k}^2$ 与 $Z_{2,k}^2$ 为 **P1.2** 优化目标中第二项与第三项的上限，且第一项为常数，因此 **P1.2** 可进一步表示为如下形式：

$$[\mathbf{P1.3}] \quad \min_{P, Z, V} \sum_{k=1}^K (Z_{1,k}^2 + Z_{2,k}^2) \quad (3-28a)$$

$$s.t. \quad P_{i,k} \leq P_{\max} \quad (3-28b)$$

$$\sum_{k=1}^K V_k \leq V_{\max} \quad (3-28c)$$

$$\frac{(1-Q_k^0)L_k}{R_{1,k}} + \frac{(1-Q_k^0)L_k}{V_k} \leq T_k \quad (3-28d)$$

(3-26), (3-27)

其中, 对于 $\forall i \in [1, 2]$ 与 $k \in [1, 2, \dots, K]$, $\mathbf{Z} = [Z_{i,k}]$ 。

由凸优化理论可知, **P1.3** 是一个凸问题。因此, **P1.3** 可以用拉格朗日对偶分解法进行求解, 具体解法如下:

对于 **P1.3** 的约束条件式(3-28b), (3-28c), (3-28d), (3-26)与式(3-27), 分别定义拉格朗日乘子 $\lambda_{1,k} \geq 0$, $\lambda_{2,k} \geq 0$, $\mu \geq 0$, $\alpha_k \geq 0$, $\gamma_{1,k} \geq 0$ 与 $\gamma_{2,k} \geq 0$ 。则 **P1.3** 的拉格朗日函数可表示为:

$$\begin{aligned}
 L(P, Z, V, \lambda, \mu, \alpha, \gamma) &= \sum_{k=1}^K (Z_{1,k}^2 + Z_{2,k}^2) + \sum_{k=1}^K \lambda_{1,k} (P_{1,k} - P_{\max}) \\
 &+ \sum_{k=1}^K \lambda_{2,k} (P_{2,k} - P_{\max}) + \mu (\sum_{k=1}^K V_k - V_{\max}) \\
 &+ \sum_{k=1}^K \alpha_k \left(\frac{(1-Q_k^0)L_k}{R_{1,k}} + \frac{(1-Q_k^0)L_k}{V_k} - T_k \right) \\
 &+ \sum_{k=1}^K \gamma_{1,k} h_1(Z_{1,k}, P_{1,k}) + \sum_{k=1}^K \gamma_{2,k} h_2(Z_{2,k}, P_{2,k})
 \end{aligned} \tag{3-29}$$

其中 $h_1(Z_{1,k}, P_{1,k})$ 与 $h_2(Z_{2,k}, P_{2,k})$ 指约束(3-26)与(3-27)中不等式左边的项, 对于

$\forall i \in [1, 2]$ 与 $\forall k \in [1, 2, \dots, K]$, 有 $\lambda \triangleq [\lambda_{i,k}]$, $\gamma \triangleq [\gamma_{i,k}]$, $\alpha = [\alpha_k]$ 。

通过式(3-29)可得问题 **P1.3** 的拉格朗日对偶函数:

$$g(\lambda, \mu, \alpha, \gamma) = \min_{P, Z, V} L(P, Z, V, \lambda, \mu, \alpha, \gamma) \tag{3-30}$$

当拉格朗日乘子保持不变时, 式(3-30)中的 $\lambda_{1,k} P_{\max}$, $\lambda_{2,k} P_{\max}$, μV_{\max} 与 $\alpha_k T_k$ 为常数项, 因此拉格朗日对偶函数(3-30)可解耦为如下形式:

$$\begin{aligned}
 \min_{P, Z, V} & (Z_{1,k}^2 + Z_{2,k}^2 + \lambda_{1,k} P_{1,k} + \lambda_{2,k} P_{2,k} + \mu V_k \\
 & + \alpha_k \left(\frac{(1-Q_k^0)L_k}{R_{1,k}} + \frac{(1-Q_k^0)L_k}{V_k} \right) \\
 & + \gamma_{1,k} h_1(Z_{1,k}, P_{1,k}) + \gamma_{2,k} h_2(Z_{2,k}, P_{2,k}))
 \end{aligned} \tag{3-31a}$$

$$s.t. \quad 0 \leq P \leq P_{\max} \tag{3-31b}$$

$$0 \leq V \leq V_{\max} \tag{3-31c}$$

由凸函数定义可知, 解耦而成的问题 (3-31) 是一个凸问题, 且符合 Slater 条件, 因此最优解 (P^*, Z^*, V^*) 可以通过 KKT 条件^[74] (Karush-Kuhn-Tucker Condition) 求得。

将问题 (3-31) 的最优解带入拉格朗日对偶函数 (3-30)，就能够得到 **P1.3** 的拉格朗日对偶问题：

$$\begin{aligned} \max_{\lambda, \mu, \alpha, \gamma} & g(\lambda, \mu, \alpha, \gamma) \\ \text{s.t.} & \lambda \geq 0, \mu \geq 0, \alpha \geq 0, \gamma \geq 0 \end{aligned} \quad (3-32)$$

对偶问题(3-32)也是凸的，因此可以使用次梯度法^[75]进行求解。求解步骤如下：

1)在第一次循环时，给定初始点 $\lambda^0, \mu^0, \alpha^0, \gamma^0$ ，用上述拉格朗日对偶分解法求得对应的最优解 (P^*, Z^*, V^*) 。

2) 然后对拉格朗日对偶变量进行迭代，直至收敛。第 i 轮循环的更新表达式如下：

$$\lambda^{i+1} = \left[\lambda^i - \theta^i (P^{i*} - P_{\max}) \right]^+ \quad (3-33)$$

$$\mu^{i+1} = \left[\mu^i - \theta^i \left(\sum_{k=1}^K V^{i*} - V_{\max} \right) \right]^+ \quad (3-34)$$

$$\alpha^{i+1} = \left[\alpha^i - \theta^i \left(\frac{(1-Q^0)L}{R^{i*}} + \frac{(1-Q^0)L}{V^{i*}} - T \right) \right]^+ \quad (3-35)$$

$$\gamma^{i+1} = \left[\gamma^i - \theta^i h(Z^{i*}, P^{i*}) \right]^+ \quad (3-36)$$

其中，对于 $\forall k \in [1, 2, \dots, K]$ ， $L = [L_k]$ ， $T = [T_k]$ 。 $[x]^+ \triangleq \max(0, x)$ ， θ^i 是一个能够保证收敛的足够小的迭代步长。

至此，完成了对 **P1.3** 的求解，得到了最优解 $Z_{1,k}^*$ ， $Z_{2,k}^*$ ， $P_{1,k}^*$ 与 $P_{2,k}^*$ 。接下来，将这些变量作为新的起始点 $Z_{1,k}^1$ ， $Z_{2,k}^1$ ， $P_{1,k}^1$ 与 $P_{2,k}^1$ 再带入 **P1.3** 进行求解。当优化变量收敛时，就得到了 **P1.3** 关于给定初始值 Q^0 的最优解。

求解完第一个子问题后，将求得的最优解带回至 **P1.1** 中，再对优化变量 Q 进行求解。此时优化问题如下所示：

$$[\text{P1.4}] \quad \min_{Q, V} \sum_{k=1}^n \left(\frac{k_u (Q_k L_k C_{u,k} L_k)^3}{T_k^2} + \frac{(1-Q_k) L_k}{R_{1,k}^*} (P_{1,k}^* + P_{cir}) \right. \quad (3-37a)$$

$$\left. + \frac{\beta Q_k L_k}{R_{1,k}^*} (P_{2,k}^* + P_{cir}) \right)$$

$$\text{s.t.} \quad 0 \leq Q_k \leq 1 \quad (3-37b)$$

$$Q_k \leq \frac{T_k f_{\max}}{C_{u,k} L_k} \quad (3-37c)$$

$$\sum_{k=1}^K V_k \leq V_{\max} \quad (3-37d)$$

$$\frac{(1-Q_k)L_k}{R_{1,k}^*} + \frac{(1-Q_k)L_k}{V_k} \leq T_k \quad (3-37e)$$

P1.4 与 **P1.2** 的形式基本相同, 因此可以采用同样的求解方法。首先通过引入辅助变量的方式将约束(3-37e)转化为 DC 约束, 将 **P1.4** 转化为凸问题。然后再用拉格朗日对偶分解法求解 **P1.4** 得到最优解 Q^* , 再将 Q^* 作为新的起始点 Q^1 带入 **P1.2** 进行循环迭代。当 Q^* 收敛时, 对应的 P^* 与 V^* 即为 **P1** 的最优解。算法的整个求解流程如下所示:

算法 3.1: 基于凸差规划的通信与计算资源联合优化算法

输入: $Q_k^0, Z_{1,k}^0, Z_{2,k}^0, P_{1,k}^0, P_{2,k}^0, i$

输出: $P_{1,k}^*, P_{2,k}^*, Q_k^*, V_k^*$

1. 初始化: $i = 0$

2. Repeat

3. Repeat

4. 将 Q_k^i 代入 **P1.2**, 以 $Z_{1,k}^i, Z_{2,k}^i, P_{1,k}^i, P_{2,k}^i$ 为初始点, 利用泰勒展开, 将 **P1.2** 转化为 **P1.3**

5. 用拉格朗日对偶分解法求解 **P1.3**, 得到最优解 $Z_{1,k}^*, Z_{2,k}^*, P_{1,k}^*, P_{2,k}^*$

6. $[Z_{1,k}^i, Z_{2,k}^i] = [Z_{1,k}^*, Z_{2,k}^*], [P_{1,k}^i, P_{2,k}^i] = [P_{1,k}^*, P_{2,k}^*]$

7. Until $Z_{1,k}^*, Z_{2,k}^*, P_{1,k}^*, P_{2,k}^*$ 收敛

8. 将 $P_{1,k}^*$ 与 $P_{2,k}^*$ 带入 **P1.1**, 得到 **P1.4**。求解 **P1.4** 得最优解 Q_k^* 与 V_k^*

9. $i = i + 1$

10. $Q_k^i = Q_k^*$

11. Until Q_k^* 收敛

3.3.2 算法性能分析

本节主要对基于凸差规划的通信与计算资源联合算法的收敛性与计算复杂度进行分析。

首先分析算法的收敛性。算法 3.1 主要由内外两层循环构成, 其中内循环为用 DC 规划求解问题 **P1.2**, 外循环为循环迭代求解 **P1.2** 与 **P1.4**。DC 规划作为一

种成熟的优化方法，已被证明是收敛的^[71]，因此算法 3.1 的内循环是收敛的。而外循环则是对相同的优化目标，即用户总能耗，通过不同的优化变量进行优化，因此每次迭代后的优化目标值将会降低或保持不变。考虑到目标函数存在下界，因此算法 3.1 将会在有限次数的迭代后收敛。

接下来分析算法的复杂度。对于内循环，假设 DC 规划需要迭代的次数为 I_1 ，每次求解的计算复杂度为 $O(X)$ 。而对于外循环，假设求解 P1.4 需要迭代的次数为 I_2 ，由于优化方法相同，因此求解的计算复杂度也为 $O(X)$ 。系统的用户总数为 K ，所以算法 3.1 的计算复杂度为 $O(I_1 I_2 K^2 X^2)$ 。

3.4 仿真结果与分析（Simulation Results and Analysis）

本节将分析所提算法在不同时延约束下的能耗表现，并对算法的资源分配策略进行探究。

仿真设置 VEC 系统中的用户数量 $K=10$ ，边缘服务器的计算能力 V_{\max} 为 30Mbps/s，用户 k 到服务器的路径损耗表示为 $\beta_0 (\frac{d_k}{d_0})^{-\varepsilon}$ ，其中 β_0 为参考距离 d_0 的路径损耗， d_k 为用户 k 到服务器的距离。仿真中的其余参数设置如表 3-1 所示。

表 3-1 多用户 VEC 系统参数配置表

Table 3-1 Parameters Assignment Table of Multi-User VEC System

参数	值
路径损耗率 β_0	-60dB
参考距离 d_0	10m
路径损耗指数 ε	3
带宽 B	1MHz
高斯白噪声功率 σ^2	-70dBm
计算复杂度 C_u	10^3 cycles/bit
芯片硬件架构相关常数 k_u	10^{-27}
最大传输功率 P_{\max}	40dBm
电路损耗功率 P_{cir}	1W

图 3-4 与图 3-5 展示了不同任务负载与时延约束下的系统能耗表现。为了探究部分任务卸载的优势，仿真中采用了 3.2 节介绍的全本地计算与全任务卸载算法作为对照。

从图 3-4 可以看出，所有算法的能耗都会随着任务量的增加而增加，而采用了部分任务卸载的通信与计算资源联合优化算法具有最小的能耗。这是因为部分任务卸载能够使用户灵活地进行通信与计算资源的联合优化，从而以合理的方式

进行任务卸载。从图 3-5 中可以看出,当时延约束变得宽松时,三种算法的能耗都有所降低,且全本地计算的能耗与部分任务卸载的能耗相同,这是因为随着时延约束 T_k 的增加,设备可以通过降低 CPU 工作频率与选择合适传输功率的方式减少能耗。

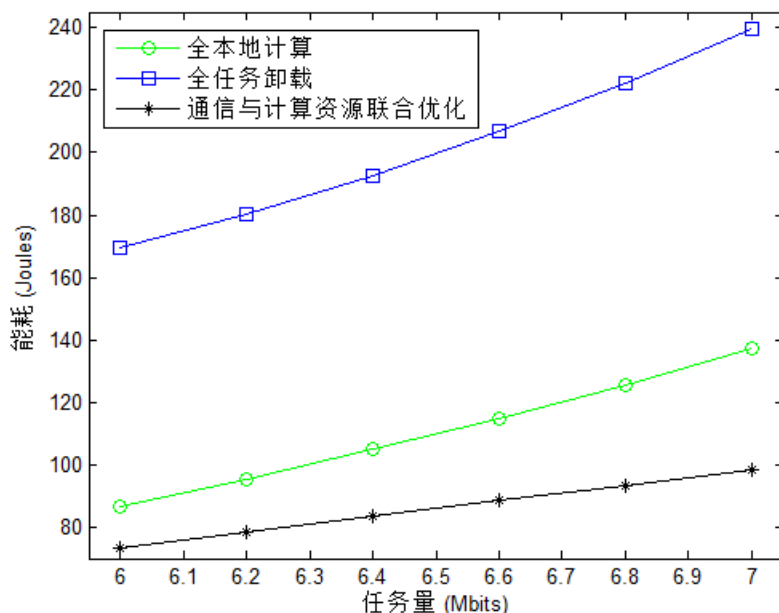


图 3-4 不同任务量下的能耗

Figure 3-4 Energy Consumption for Different Task Size

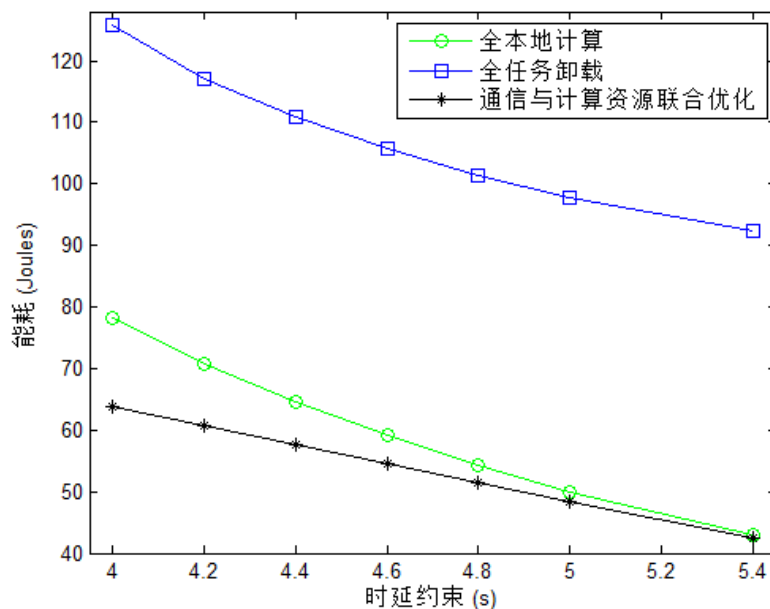


图 3-5 不同时延约束下的能耗

Figure 3-5 Energy Consumption for Different Delay Constraints

为了探究时延约束对资源分配策略的具体影响,本小节针对三个具有不同通信条件与计算能力的用户进行了分析,这三个用户的参数如表 3-2 所示。

表 3-2 用户参数配置表

Table 3-2 Parameters Assignment Table of Users

用户	最大 CPU 频率 f_{\max}	与服务器的距离
1	4GHz	320m
2	2GHz	240m
3	4GHz	240m

图 3-6 展示了在不同时延约束下，三个用户得到的服务器计算资源情况。可以看出，在时延约束较紧的情况下，服务器为用户 2 分配了大量的计算资源。这是因为用户 2 的计算能力不足，需要服务器提供较多的计算资源来完成任务。而当时延约束变得宽松时，服务器分配给用户 2 与用户 3 的资源十分接近。这是因为在 DVS 技术的调节下，用户 3 降低了自身的 CPU 工作频率，与用户 2 的 CPU 工作频率较为接近，且它们的通信条件也相同。用户 1 总是比其余两个用户占用更少的服务器计算资源，这是因为用户 1 与服务器距离较远，通信损耗较大，因此用户 1 更倾向于进行本地计算。

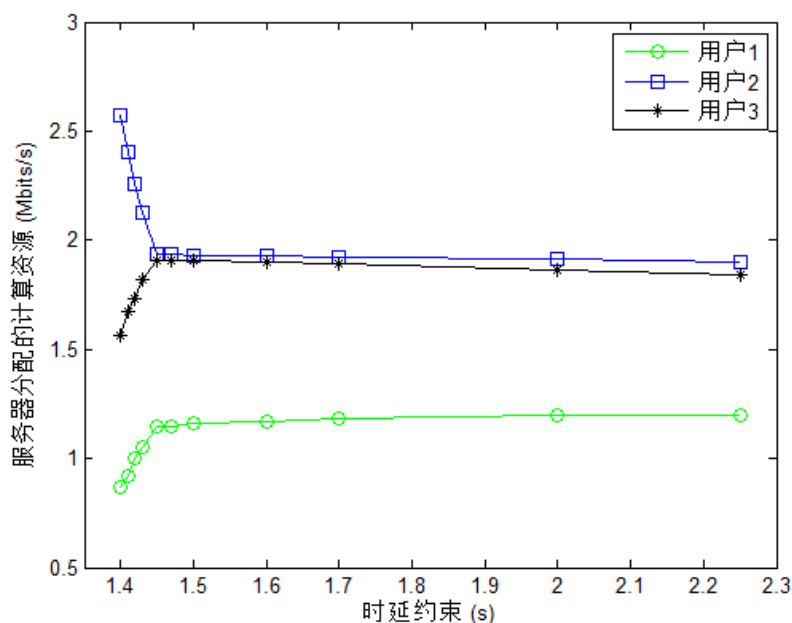


图 3-6 不同时延约束下各用户被分配到的计算资源

Figure 3-6 Allocated Computation Resource for Different Delay Constraints

图 3-7 展示了在不同时延约束下，三个用户各自的最优任务卸载比。从图中可以看出，随着时延约束的放松，任务卸载比有一个先减小后增大的趋势。这是因为当时延约束较为严格时，将一部分任务卸载至边缘服务器更有利于减少能耗。而当时延约束较为宽松时，降低 CPU 工作频率进行更多的本地计算更为节能。这一观点将在图 3-8 进一步得到验证。此外，随着时延约束的放宽，设备 2 与设备 3 的任务卸载比例几乎一致，这与上述分析服务器资源分配相同的原因一致。

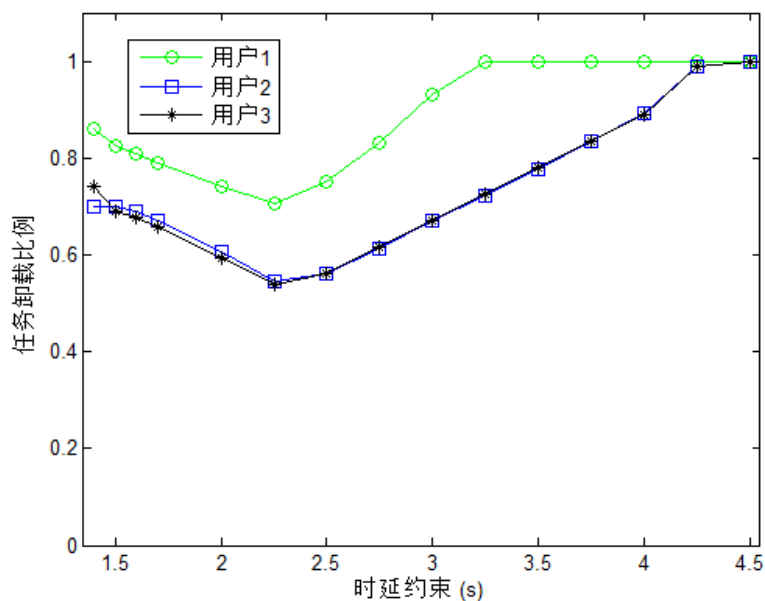


图 3-7 不同时延约束下各用户的任务卸载比例

Figure 3-7 Task Offloading Ratio for Different Delay Constraints

图 3-8 展示了不同时延约束下各用户 CPU 工作频率的变化。从图 3-8 可以看出，随着时延约束的放宽，各用户的 CPU 工作频率呈下降的趋势。这个仿真结果与理论分析一致，即降低 CPU 工作频率能够有效地降低能耗。将图 3-7 与图 3-8 进行对比可以发现，随着任务卸载比增加时，用户 CPU 工作频率保持不变。这意味着在此工作频率下，本地计算比任务卸载更节能，随着时延约束的放宽，用户选择将更多的任务放在本地进行计算。

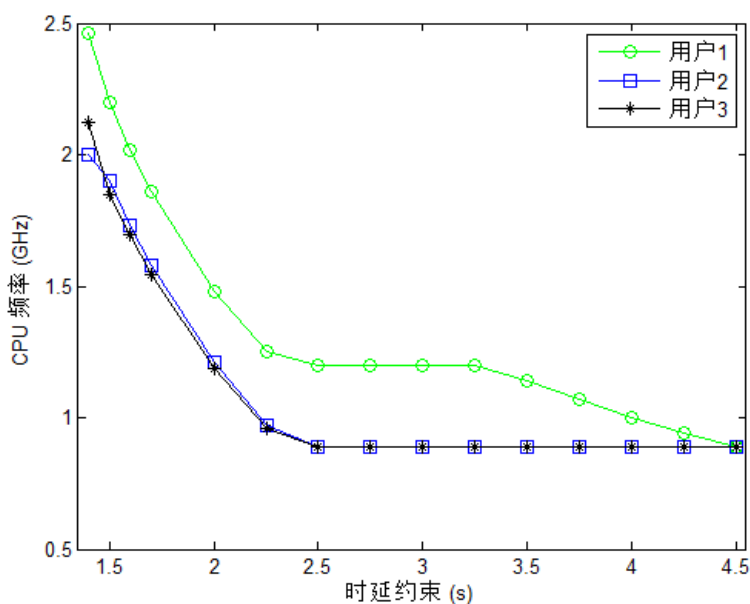


图 3-8 不同时延约束下的各用户的 CPU 频率

Figure 3-8 CPU Frequency for Different Delay Constraints

3.5 本章小结 (Conclusions of the Chapter)

针对资源受限 VEC 系统中的资源分配问题, 本章利用 DVS 技术, 提出了基于通信与计算资源联合优化的资源分配方案。为了解决非凸优化问题, 本章提出了基于 DC 规划的优化算法, 将原问题转化成了两个凸子问题进行求解。最终通过仿真实验验证了算法在能耗方面的优化性能

4 基于多臂赌博机算法的动态 VEC 任务卸载

4 Task Offloading of Dynamic Vehicular Edge Computing Network based on Multi-armed Bandit Algorithm

4.1 引言 (Introduction)

第三章针对交通流量密集的 VEC 网络中存在的资源受限问题, 提出了一个基于 DVS 技术的任务卸载与资源分配方案, 通过对系统内的通信与计算资源进行联合优化, 最终优化了系统中各用户的总能耗。该优化方案是集中式的, 它假设网络中的全局信息已知, 由边缘服务器对系统内的资源进行统筹规划。这种集中式的优化方案对于交通流量密集, 车辆移动性受限的 VEC 网络来说是可行的。然而, 在交通流量正常, 道路通畅的 VEC 网络中, 车辆具有较高的移动性, 这使得网络中的信道状态信息与拓扑结构处于快速变化的状态, 想要获得实时的网络全局信息就变得十分困难。在这种情况下, 集中式优化方案的有效性将会有所降低。

针对由车辆移动性导致的动态 VEC 网络链路稳定性差与网络拓扑结构快速变化的问题, 本章提出了基于强化学习的分布式在线学习任务卸载方案。考虑到现在的智能车辆配备了越来越多的计算资源, 且车辆高速移动会带来频繁的服务器切换问题, 本章基于车联网中“车辆即是资源^[45]”以及 MEC 中自组织云^[76]的概念, 研究车辆与车辆间 (V2V) 进行的任务卸载, 即网络中的车辆可以用自己的空闲资源为其他有需要的用户提供计算服务。本章将有任务卸载需求的车辆称为任务车辆 (Task Vehicle, TaV), 提供计算服务的车辆称为服务车辆 (Service Vehicle, SeV)。而由于车辆的空闲资源是随时间波动的, 因此 TaV 需要对周围提供计算服务的 SeV 进行评估, 从而做出高效的任務卸载决策。本章将 V2V 任务卸载问题建模为一个多臂赌博机 (MAB) 问题, 将需要进行任务卸载的车辆当作智能体, 将提供计算服务的车辆当作摇臂, 将 SeV 的服务能力作为摇臂的奖励, 车辆进行任务卸载的过程就是智能体选择摇臂的过程。车辆的移动性导致了 TaV 周围的 SeV 数量是动态变化的, 在 MAB 模型中表现为摇臂数量处于动态变化的状态, 这使得传统的多臂赌博机算法难以适用于本章所提问题。为此, 本章提出了基于多臂赌博机的分布式在线学习任务卸载算法, 对传统的多臂赌博机算法进行了改进, 使 TaV 能够在任务卸载的过程中学习到动态网络环境下的 SeV 服务能力。此外, 本章还设计了基于路边单元辅助的用户协作机制, 使网络中的用户

可以共享彼此学习到的信息，从而提高所有用户的学习效率。理论分析表明，本章所提算法可获得对数形式的遗憾函数。仿真结果证明，本章所提算法与其他现有算法相比，更能适应复杂时变的车辆边缘计算网络环境。

4.2 系统模型（System Model）

本章考虑一个多用户的 VEC 场景，该场景由多个车辆以及一个路边单元组成，系统模型如图 4-1 所示。车辆与车辆间通过 V2V 的方式进行通信，车辆与路边单元通过 V2I 的方式进行通信。为了保障 TaV 与 SeV 之间的连接质量，TaV 会在其通信范围内选择行驶方向相同的 SeV 作为任务卸载的候选节点。车辆间的驾驶状态信息（如驾驶方向、驾驶速度等）可以通过 DSRC（Dedicated short-range communication）技术^[77]等车载通信协议进行交换。在 VEC 网络中，车辆的高机动性将导致 SeV 在不同 TaV 的通信范围内切换，从而降低了 TaV 的学习效率。为此，本章提出基于路边单元辅助的用户协作机制。具体来说，TaV 在每次任务卸载结束后都会将学习到的关于 SeV 的知识通过 V2I 的方式上传到 RSU，RSU 将知识汇总后再广播给通信范围内的所有 TaV。本章采用的是分布式的任务卸载策略，即每个 TaV 只考虑自身的情况进行任务卸载。系统模型中的 TaV 采用部分任务卸载方式，优化目标为减少 TaV 的任务时延。整个任务卸载过程分为四个部分：任务分割，任务传输，任务计算以及结果回传。任务卸载的流程如图 4-2 所示，下面将介绍这四个部分的数学模型。

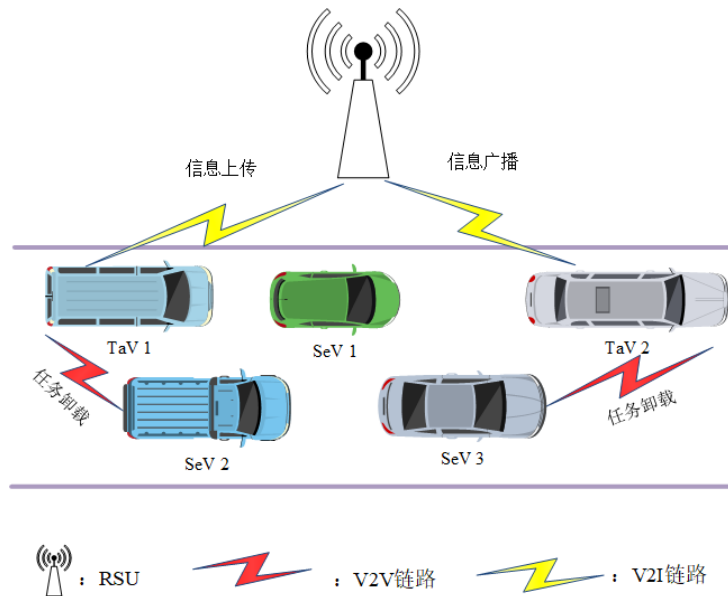


图 4-1 车辆边缘计算网络车辆间任务卸载模型图

Figure 4-1 Architecture of V2V Task Offloading Model in VEC Network

4.2.1 任务卸载模型

(1) 任务分割

为了体现 VEC 系统的时变特性，本章将系统中的时间划分为 T 个离散的时隙。记 TaV 的 SeV 候选集为 $N(t)$ ，TaV 会在每个时隙开始时从 $N(t)$ 中选择一个 SeV 进行任务卸载，记被选择的 SeV 为 SeV_n ， $n \in N(t)$ 。TaV 的计算任务用 $\varphi(t)$ 表示， $\varphi(t) = [L_t, C_t]$ ，其中 L_t 表示任务量的大小，单位为比特， C_t 为任务的计算复杂度，指计算 1 比特任务所需的 CPU 周期数。由于采用的是部分任务卸载的方式，因此 TaV 将任务分成两个子任务，分别在本地与 SeV 进行运算。记 $\lambda_{n,t} \in [0,1]$ 为任务卸载比例，表示卸载任务量占总任务量的比重。所以卸载给 SeV 的任务数据量可表示为 $\lambda_{n,t}L_t$ ，而本地计算的任务数据量表示为 $(1-\lambda_{n,t})L_t$ 。

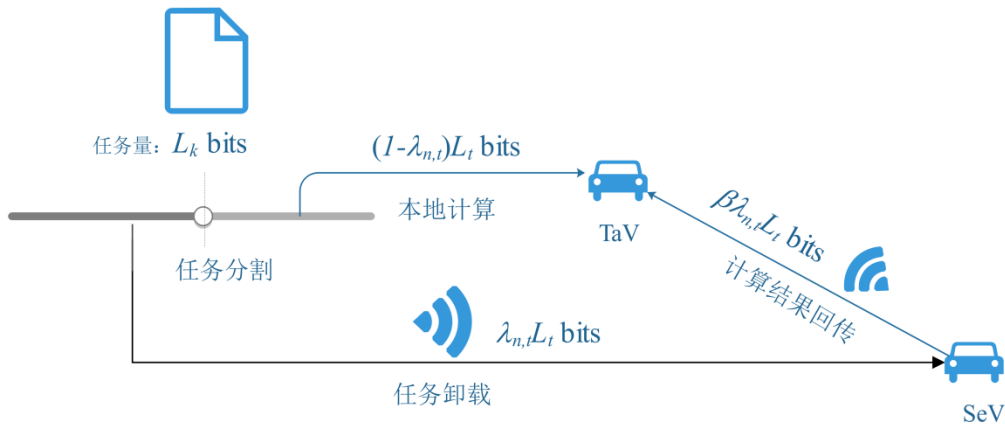


图 4-2 车辆间任务卸载流程图

Figure 4-2 Diagram of V2V Task Offloading

(2) 任务传输

当 TaV 完成任务分割之后，就会在进行本地计算的同时将需要 SeV 计算的子任务进行上传。记 TaV 的发射功率为 P_t ，信道带宽为 B ，TaV 与 SeV_n 间的上行链路信道增益为 $h_{n,t}^o$ ，则上行传输速率 $r_{n,t}^o$ 可表示为：

$$r_{n,t}^o = B \log_2 \left(1 + \frac{P_t h_{n,t}^o}{\sigma^2} \right) \quad (4-1)$$

其中 σ^2 为高斯白噪声的噪声功率。由于模型采用 OFDM 技术，因此不考虑同信道噪声干扰。

传输时延可表示为：

$$d_o(n,t) = \frac{\lambda_{n,t}L_t}{r_{n,t}^o} \quad (4-2)$$

(3) 任务计算

计算任务被分割成了两个子任务分别在 TaV 与 SeV 中进行运算。记 TaV 的 CPU 工作频率为 f_t' ，则 TaV 进行本地计算的时延 $d_{lc}(n,t)$ 可表示为：

$$d_{lc}(n, t) = \frac{C_t(1 - \lambda_{n,t})L_t}{f_t^l} \quad (4-3)$$

记 SeV 计算卸载任务的 CPU 工作频率为 $f_{n,t}^s$ ，则 SeV 完成计算任务时延 $d_{sc}(n, t)$ 可表示为

$$d_{sc}(n, t) = \frac{C_t \lambda_{n,t} L_t}{f_{n,t}^s} \quad (4-4)$$

(4) 计算结果回传

SeV 处理完子任务后，将计算结果回传给 TaV。记 $h_{n,t}^b$ 为 SeV n 到 TaV 的下行链路信道增益，SeV 的发射功率为 P_s ，则下行传输速率 $r_{n,t}^b$ 可表示为：

$$r_{n,t}^b = B \log_2 \left(1 + \frac{P_s h_{n,t}^b}{\sigma^2} \right) \quad (4-5)$$

用 $\beta \in [0, 1)$ 表示计算结果占计算任务大小的比例，则计算结果的传输时延可表示为：

$$d_b(n, t) = \frac{\beta \lambda_{n,t} L_t}{r_{n,t}^b} \quad (4-6)$$

由于本地子任务的运算和卸载任务的传输是并行执行的，所以整个任务的时延 $d_{sum}(n, t)$ 可表示为：

$$d_{sum}(n, t) = \max(d_{lc}(n, t), d_{off}(n, t)) \quad (4-7)$$

其中 $d_{off}(n, t)$ 表示处理卸载任务所需时延， $d_{off}(n, t) = d_o(n, t) + d_{sc}(n, t) + d_b(n, t)$ 。

4.2.2 优化问题描述

本章的优化目标为最小化 T 个时隙内的平均时延。这个优化问题可以表述为以下形式：

$$[\mathbf{P2}] \quad \min_{\lambda_{n,t}, a_1, \dots, a_T} \frac{1}{T} \sum_{t=1}^T d_{sum}(n, t) \quad (4-8)$$

其中， a_t 表示 TaV 在 t 时隙的卸载决策， $a_t \in \mathcal{N}(t)$ 。

本章采取分布式的任务卸载策略，故 TaV 只需要选择服务能力最强的 SeV 进行任务卸载。由于本章优化目标是时延，所以将 SeV 的服务能力定义为处理 1 比特任务所消耗的时间，SeV n 的服务能力 $u_{n,t}$ 的表达式如下：

$$u_{n,t} = \frac{d_{off}(n, t)}{\lambda_{n,t} L_t} \quad (4-9)$$

从式(4-9)可以看出, SeV 的服务能力与网络中的信道条件 $h_{n,t}^o$, $h_{n,t}^b$ 以及其 CPU 工作频率 $f_{n,t}^s$ 有关。如果这些信息对于 TaV 来说都是已知的话, 则 TaV 就能够轻易地选出最佳的 SeV 进行任务卸载, 它的卸载选择表达式如下:

$$a_t = \arg \min_{n \in N(t)} u_{n,t} \quad (4-10)$$

然而在 VEC 网络中, 车辆的移动性使网络内的拓扑结构与信道状况始终处于快速变化的状态, 且 SeV 能够提供的计算能力也是浮动的。想要准确地获取上述信息, TaV 需要十分频繁地与周围的 SeV 交换信息, 从而产生大量的信令开销, 得不偿失。为了应对这个问题, 本章将 V2V 任务卸载问题建模成一个多臂赌博机问题, 并提出了基于多臂赌博机的分布式在线学习任务卸载策略, 使 TaV 能够自适应地对网络内的信息进行学习, 从而做出合理的任务卸载决策。

4.3 基于多臂赌博机的分布式在线学习任务卸载算法 (Multi-armed Bandit based Distributed Online Learning Task Offloading Algorithm)

4.3.1 算法描述

对于本章提出的模型, 将 TaV 当作智能体, 将服务能力未知的 SeV 当作摇臂, 将 SeV 的服务能力可视为摇臂的奖励, TaV 选择 SeV 进行任务卸载的过程即是智能体选择摇臂的过程。这样一来, 动态 VEC 网络中的任务卸载问题就可以抽象成一个多臂赌博机问题进行求解。在传统的多臂赌博机问题中, 摇臂通常是保持不变的, 然而本章研究问题中的 SeV 候选集 $N(t)$ 却是变化的量, 这使得传统的多臂赌博机算法 (如 ϵ -贪婪算法, UCB1 算法) 不适用于本问题。为此, 本章对传统的 UCB 算法进行了改进, 提出了基于多臂赌博机的分布式在线学习任务卸载算法, 使 TaV 在进行任务卸载的时候能够自适应地对 SeV 的服务能力进行学习, 从而做出合理的任务卸载策略, 降低平均任务时延。车辆的移动性使 SeV 在不同 TaV 的通信范围内切入切出, 影响了 TaV 的学习效率。针对这个问题, 本算法提出了一种基于路边单元辅助的用户协作机制, 它能够帮助通信范围内 TaV 共享关于 SeV 的服务能力信息, 从而提高系统中 TaV 的学习效率。具体来说, 每个 TaV 在当前时隙结束时会将自己学习到的关于 SeV 的服务能力信息上传至路边单元, 路边单元进行汇总后在下一个时隙开始时将信息广播给通信范围内的所有 TaV。下面将对本章所提出的算法进行详细介绍。

与 UCB 算法相同, 本算法需要一个初始化的过程, 即保证 TaV 对每个 SeV 都至少选择过一次^[78]。因此, 在每个时隙开始时, TaV 首先会检查自身的 SeV 候选集为 $N(t)$ 中是否出现了未知的 SeV。若出现了, TaV 就会选择这个新的 SeV 进行任务卸载。由于用户对于这个新的 SeV 没有任何先验知识, 因此任务卸载比

例设定为一个预定值 λ' 。对于已经具备有相关的先验信息的 SeV，可以计算出最佳的任务卸载比例。在 **P2** 中，本地计算的时延消耗 $d_{lc}(n,t)$ 随任务卸载比例 $\lambda_{n,t}$ 单调递减，而任务卸载的总时延 $d_{off}(n,t)$ 随任务卸载比例单调递增。结合优化目标 $d_{sum}(n,t)$ 的表达式(4-7)，可知最佳的任务卸载比例 $\lambda_{n,t}^*$ 在 $d_{lc}(n,t)$ 与 $d_{off}(n,t)$ 相等时取得。结合式(4-3)与式(4-9)可得 $\lambda_{n,t}^*$ 的表达式：

$$\lambda_{n,t}^* = \frac{C_t}{C_t + f_l' \bar{u}_{n,t-1}} \quad (4-11)$$

其中 $\bar{u}_{n,t-1}$ 为 SeV n 截止 $t-1$ 时隙的平均服务能力。

与经典的 UCB 算法一样，为了解决探索-利用问题，本节通过定义一个效用函数来衡量各 SeV 的服务能力，从而实现利用与探索策略的均衡。效用函数表达式如下：

$$\hat{u}_{n,t} = \bar{u}_{n,t-1} - \sqrt{\frac{\eta(1-\lambda_{n,t})\tilde{L}_t v_{n,t} \ln(t-t_n)}{k_{n,t-1}}} \quad (4-12)$$

其中 η 为权重参数， $k_{n,t-1}$ 指 SeV n 截止 $t-1$ 时隙被选择的次数， t_n 指 SeV n 第一次被选择的时隙， \tilde{L}_t 为归一化的任务量， $v_{n,t}$ 值 SeV n 服务能力的方差。 \tilde{L}_t 与 $v_{n,t}$ 的表达式会在稍后进行介绍。

与 2.2.2 节中介绍的式(2-4)相同，效用函数的第一项 $\bar{u}_{n,t-1}$ 指 SeV n 的期望服务能力，而第二项表示该期望值的置信度。当第一项的值较大时，算法会趋向于“利用”策略，而当第二项的值较大时，算法会趋向于“探索”策略。受 VUCB 算法^[79]与 UCB1-Tuned^[80]算法的启发，期望值的置信度由卸载任务量的归一化值 $(1-\lambda_{n,t})\tilde{L}_t$ 、SeV n 第一次被选择的时隙 t_n 、SeV 服务能力的方差 $v_{n,t}$ 以及其被选择的次数 $k_{n,t-1}$ 构成。置信度表达式中的权重参数 η 用于调整算法对探索策略的倾向程度， η 越大，算法越倾向于进行探索。归一化的任务量也具有类似的作用，当任务量较小时，算法会倾向于进行“探索”，因为此时就算选择了服务能力较差的 SeV 也不会产生较大的时延。而当任务量较大时，算法就会倾向于选择已知的最佳节点，以避免时延过大。归一化的任务量 \tilde{L}_t 的表达式如下：

$$\tilde{L}_t = \frac{L_t - L_{\min}}{L_{\max} - L_{\min}} \quad (4-13)$$

其中 L_{\max} 与 L_{\min} 分别指任务负载的最大值与最小值。

式(4-12)中第二项的其余部分 $\sqrt{\frac{v_{n,t} \ln(t-t_n)}{k_{n,t-1}}}$ 表示对该 SeV 的确信程度，若方

差 $v_{n,t}$ 较大，或者被选择的次数 $k_{n,t-1}$ 较小，则意味着该 SeV 还没有被充分探索。

效用函数中的置信度表达式使得当前期望服务能力较小的 SeV 也会有被选中的机会，保证了算法对未知 SeV 的探索能力。根据 UCB1-tuned 算法^[80]，方差 $v_{n,t}$ 的表达式如下：

$$v_{n,t} = \min\left(\frac{1}{4}, \sum_{\tau=1}^{t-1} \frac{1}{k_{n,t-1}} k_{n,\tau} u_{n,\tau}^2 - u_{n,t-1}^2 + \sqrt{\frac{\eta(1-\lambda_{n,t})\tilde{L}_t v_{n,t} \ln(t-t_n)}{k_{n,t-1}}}\right) \quad (4-14)$$

其中， $u_{n,\tau}$ 为 SeV 在 τ 时隙的服务能力。

算法 4.1：基于多臂赌博机的分布式在线学习任务卸载算法

输入： λ' ， \tilde{L}_t ， f_t^l

输出： a_1, \dots, a_T ， $\lambda_{a_t,t}^*$

1. **for** $t=1, \dots, T$ **do**

2. **if** $N(t)$ 中出现了新的 SeV n

3. **if** 该 SeV n 的服务能力 $\bar{u}_{n,t-1}$ 能够从 RSU 广播的信息获得

4. $k_{n,t} = 1$ ， $t_n = t$ ， 根据式(4-11)计算相应的 $\lambda_{n,t}^*$

5. **else** $a_t = n$ ， $\lambda_{n,t} = \lambda'$ ， TaV 选择 SeV n 进行任务卸载， 然后根据式(4-9)计算 $u_{n,t}$ ， $k_{n,t} = 1$ ， $t_n = t$

6. **end if**

7. **end if**

8. **if** $a_t = 0$

9. 根据式(4-11)计算每个 SeV $n \in N(t)$ 的最佳卸载比 $\lambda_{n,t}^*$

10. 根据式(4-12)计算每个 SeV $n \in N(t)$ 的效用函数值 $\hat{u}_{n,t}$

11. 根据式(4-15)选择最佳的 SeV a_t 并进行任务卸载

12. 根据式(4-9)计算 SeV a_t 的服务能力 $u_{a_t,t}$

13. **end if**

14. 更新参数 $\bar{u}_{a_t,t} \leftarrow \frac{\bar{u}_{a_t,t-1} k_{a_t,t-1} + u_{a_t,t}}{k_{a_t,t-1} + 1}$ ， $k_{a_t,t} \leftarrow k_{a_t,t-1} + 1$

15. 将 $\bar{u}_{a_t,t}$ 上传至 RSU

16. **end for**

在获取了 $N(t)$ 中所有 SeV 的效用函数后, TaV 将会选择效用函数最小的 SeV 进行任务卸载, TaV 的卸载决策表达式如下:

$$a_t = \arg \min_{n \in N(t)} \hat{u}_{n,t} \quad (4-15)$$

做完卸载决策, TaV 按照 4.2 小节中介绍的流程进行任务卸载, 并在完成任务后得到 a_t 的服务能力, 对参数 $\bar{u}_{a_t,t}$ 与 $k_{a_t,t}$ 进行更新, 接着将更新后的 $\bar{u}_{a_t,t}$ 上传至路边单元。算法整体的流程如算法 4.1 所示。

4.3.2 算法性能分析

为了分析所提算法的性能, 本小节采用在 MAB 算法中广泛使用的累积遗憾作为衡量算法的性能指标。累积遗憾具体表示为所提算法带来的收益与全局最佳收益之间的差值, 描述了算法所作的决策与最优解之间的距离。为了进行理论分析, 将 T 个时隙划分成 B 个时段, 每个时段由若干个时隙组成, 并假设每个时段的 SeV 集合 $N(t)$ 保持不变。记时段 b 的起始时隙为 t_b , 末尾时隙为 t_b' , $u_n = E[u_{n,t}]$ 为 SeV n 的期望服务能力, $a_b^* = \arg \min_{n \in N(t)} u_n$ 为时段 b 的最佳决策, 其服务能力为 u_b^* 。则算法在 T 个时隙的累积遗憾 R_T 表达式如下:

$$R_T = \sum_{b=1}^B E \left[\sum_{t=t_b}^{t_b'} x_t (u_{n,t} - u_b^*) \right] \quad (4-16)$$

其中 x_t 表示 SeV n 在时隙 t 处理的计算负载量。在现实中, x_t 的值在每个时隙都不相同, 为了进行理论分析, 本节假设 x_t 在一个时段内保持不变。研究[81]表明, 上述假设对结论不会产生影响。

记 $\Delta_{n,b} = u_{n,t} - u_b^*$ 表示选择 SeV n 产生的遗憾值, 则式(4-16)可以进一步表示为以下形式:

$$R_T = \sum_{b=1}^B E \left[x_t \sum_{n \neq a_b^*} k_{n,t} \Delta_{n,b} \right] = x_t \Delta_{n,b} \sum_{b=1}^B E \left[\sum_{n \neq a_b^*} k_{n,t} \right] \quad (4-17)$$

从式(4-17)可以看出, 累积遗憾值 R_T 主要由 $E[k_{n,t}]$ 来决定。对于一个任意的表达式 ς , 若 ς 为真, 则记 $\{\varsigma\} = 1$, 反之记 $\{\varsigma\} = 0$ 。设 l 为一个任意大小的正整数, 则对于 $k_{n,t}$ 有以下不等式:

$$\begin{aligned}
k_{n,t} &\leq 1 + \sum_{t=2}^T \{a_t = n\} \\
&\leq l + \sum_{t=2}^T \{a_t = n, k_{n,t-1} \geq l\} \\
&\leq l + \sum_{t=2}^T \{\bar{u}_{n,t-1} - I_{n,t-1} \leq u_{t-1}^* - I_{t-1}^*, k_{n,t-1} \geq l\} \\
&\leq l + \sum_{t=2}^T \{\min_{l < s_n < t} \bar{u}_{n,s_n} - I_{n,s_n} \leq \max_{0 < s < t} u_s^* - I_s\} \\
&\leq l + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_n=l}^{t-1} \{\bar{u}_{n,s_n} - I_{n,s_n} \leq u_s^* - I_s\}
\end{aligned} \tag{4-18}$$

其中 $I_{n,t-1}$ 表示 $\hat{u}_{n,t}$ 的置信度，即式(4-12)中的第二项。

按照本章所提出的算法，若 SeV n 在时隙 t 被选中了，则意味着它的效用函数值比最优 SeV 的效用函数值小，即 $\hat{u}_{n,t} \leq \hat{u}_t^*$ 。将该不等式展开如下：

$$\bar{u}_{n,s_n} - I_{n,s_n} \leq u_s^* - I_s \tag{4-19}$$

由 UCB 理论可知，式(4-19)要成立，必须至少满足以下三个不等式中的一个：

$$u_n - u^* < 2I_{n,s_n} \tag{4-20}$$

$$u^* + I_{n,s_n} \leq \bar{u}_n^* \tag{4-21}$$

$$\bar{u}_n + I_{n,s_n} \leq u_n \tag{4-22}$$

令 $\eta = 2$ ，结合式(4-20)与式(4-12)，可得：

$$k_{n,t-1} \leq \frac{8\ln(t-t_n)}{\Delta_n^2} \tag{4-23}$$

其中 $\Delta_n = u_n - u^*$ 。

令 $l = \left\lceil \frac{8\ln(t-t_n)}{\Delta_n^2} \right\rceil$ ，在式(4-20)中，对于 $\forall s_n > l$ ，有：

$$u_n - u^* - 2I_{n,s_n} \geq u_n - u^* - \Delta_n = 0 \tag{4-24}$$

式(4-24)与式(4-20)矛盾，故式(4-20)不成立。根据 Chernoff-Hoeffding 不等式^[82]，式(4-21)与(4-22)成立的概率如下：

$$P(\bar{u}_n^* - u^* \leq I_{n,s_n}) \leq e^{-4\ln(t-t_n)} \tag{4-25}$$

$$P(u_n - \bar{u}_n \geq I_{n,s_n}) \leq e^{-4\ln(t-t_n)} \tag{4-26}$$

结合式(4-18)与式(4-25), (4-26)可得

$$\begin{aligned}
 E[k_{n,T}] &\leq \left\lfloor \frac{8\ln(t-t_n)}{\Delta_n^2} \right\rfloor + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_n=l}^{t-1} \{P(\bar{u}_{n,t-1}^* - u^* \leq I_{n,s_n}) + P(u_n - \bar{u}_n \geq I_{n,s_n})\} \\
 &\leq \frac{8\ln(t-t_n)}{\Delta_n^2} + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_n=\left\lfloor \frac{8\ln(t-t_n)}{\Delta_n^2} \right\rfloor}^{t-1} \{2t^{-4}\} \\
 &\leq \frac{8\ln(t-t_n)}{\Delta_n^2} + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_n=1}^{t-1} \{2t^{-4}\} \\
 &\leq \frac{8\ln(t-t_n)}{\Delta_n^2} + 1 + \frac{\pi^2}{3}
 \end{aligned} \tag{4-27}$$

将式(4-27)带入式(4-17), 可得本章所提算法在 T 个时隙内的累积遗憾上限:

$$R_T \leq x_t \sum_{b=1}^B \left[\sum_{n \neq a_b^*} \left(\frac{8\ln T}{\Delta_{n,b}} + \left(1 + \frac{\pi^2}{3}\right) \Delta_{n,b} \right) \right] \tag{4-28}$$

式(4-28)表明, 本章所提算法的累积遗憾与运行时间 T 的对数呈正比, 说明本章所提算法与全局最优解的偏差呈次线性。

4.4 仿真结果与分析 (Simulation Results and Analysis)

本节将对本章所提出的算法使用 MATLAB 进行仿真, 并对仿真结果进行分析。为了验证算法性能, 本仿真考虑系统中有 3 个 TaV 与 30 个 SeV, SeV 随机分布在 TaV 周围 50m 的范围内, 且 TaV 与 SeV 的距离在每个时隙内随机变化, 变化范围为-5m 到 5m。SeV 的空闲计算资源占自身总资源的比重随机分布在 20%~50%之间, 其余的仿真参数如表 4-1 所示。

表 4-1 动态 VEC 系统参数配置表

Table 4-1 Parameters Assignment Table of Dynamic VEC System

参数	值
传输功率 P	0.1W
高斯白噪声功率 σ^2	10^{-13}W
带宽 B	10MHz
计算复杂度 C_u	10^3 cycles/bit
权重参数 η	0.2
任务卸载比例预定值 λ'	0.5
SeV 的 CPU 最大工作频率 f_{\max}^s	5GHz
TaV 的 CPU 最大工作频率 f_{\max}^l	3GHz
TaV 的通信距离	50m
任务量 L_i	0.5~0.8Mbits

参照文献[83]中对城市交通的 V2V 通信链路研究, 仿真中的信道增益建模如下:

$$h_{n,t}^o = h_{n,t}^b = A_0 l^{-\varphi} \quad (4-29)$$

其中 $A_0 = -17.8\text{dB}$, l 为 TaV 与 SeV 的距离, φ 为路径损耗因子, 随机分布于 $[2, 4]$ 。

为了详细分析本算法的性能, 本节将基于多臂赌博机的分布式在线学习算法简写为 MDOLTO 算法, 并与 UCB 算法、VUCB 算法^[79]、Optimal 算法与 Random 算法进行对照, 对照算法介绍如下:

(1) UCB 算法, 即置信区间上界算法, 是 MAB 问题中的经典算法, 本文在 2.2.2 小节对其进行了详细介绍。参照算法定义, 该算法中的置信度表达式在

仿真中设置为 $\sqrt{\frac{\beta \ln t}{k_{t-1,n}}}$ 。

(2) VUCB 算法, 主要研究了 MAB 问题中摇臂可能发生变化的问题, 解决方案是在已有 UCB 算法的基础上将摇臂出现的时隙纳入到对其置信度的衡量

中, 该算法中的置信度表达式在仿真中设置为 $\sqrt{\frac{\beta \ln(t - t_n)}{k_{t-1,n}}}$ 。

(3) Optimal 算法, 即最优决策算法, 该算法拥有网络的实时全局信息, 并按照式(4-10)做出最佳的任务卸载决策。此算法具有最优的性能, 主要用于评估其他算法的优劣。

(4) Random 算法, 此算法要求 TaV 随机地对 $N(t)$ 中的 SeV 进行选择, 用于与其他算法进行对照。

各算法的平均时延随时间的变化如图 4-3 所示。从图中可以看出, 本章所提算法具备除 Optimal 算法外的最低时延, 证明了其对动态的 VEC 网络环境具有良好的适应能力。由于 UCB 类的算法都需要进行初始化, 因此算法在初始阶段的平均延迟较大。而 UCB 类的算法性能都远高于随机算法, 这表明 MAB 算法能够有效地适应动态的 VEC 网络环境。此外, VUCB 算法比 UCB 算法具有更低的时延, 证明了在 VEC 网络中考虑动态 SeV 集的影响有利于提升算法性能。

各算法的累积遗憾随时间的变化如图 4-4 所示。从图中可以看出, 本章所提算法与 VUCB 算法相比, 累积遗憾降低了 85%。此外, 随着时间的推移, 本章所提算法的累积遗憾呈次线性增加, 与理论分析相符。这意味着算法能够渐进地收敛于最优策略。

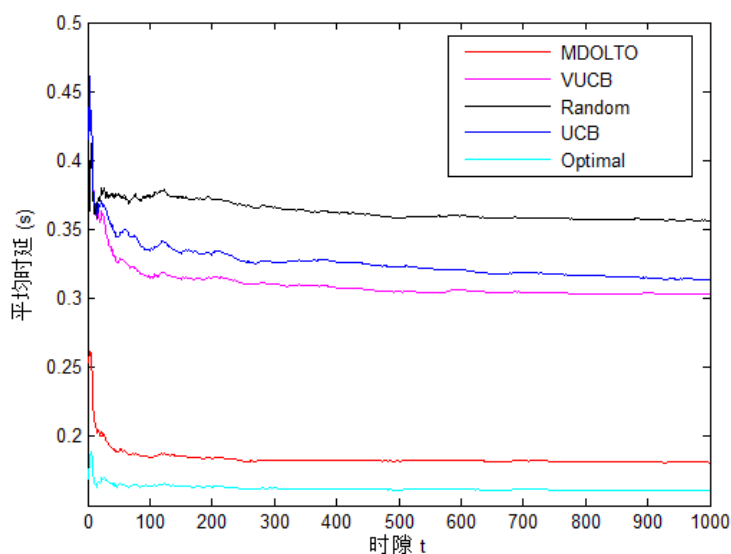


图 4-3 各算法的平均时延
Figure 4-3 Average Delay for Each Algorithm

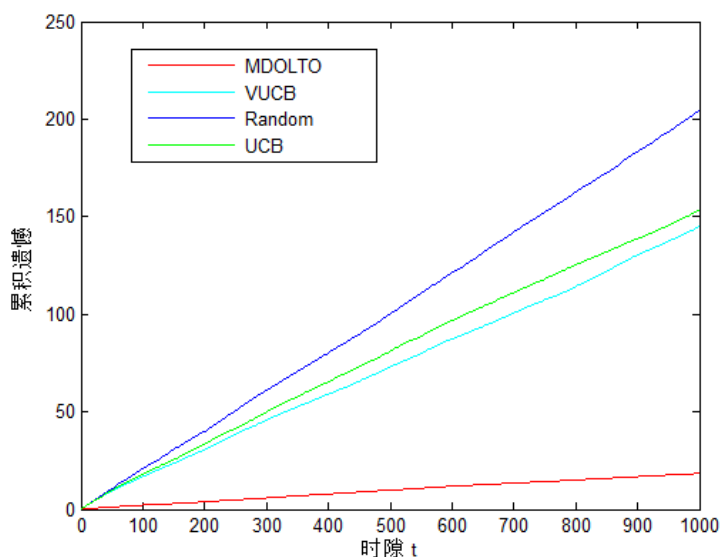


图 4-4 各算法的累积遗憾
Figure 4-4 Cumulated Regret of Each Algorithm

图 4-5 展示了各算法的最佳决策率，即算法做出最佳决策的次数占总决策数的比例。从图中可以看出，与其他算法相比，本章所提算法的最佳决策率最高。然而，由于仿真中的实验环境处于高度变化的状态，所有算法的最佳决策率都没有超过 50%。结合图 4-3 与图 4-4 进行分析，可做出如下判断：本章所提算法在最佳决策率不足 45% 的情况下依旧具备较好的性能，是因为选择了与最佳决策相差无几的次优解。

为了探究 MDOLTO 算法中的部分卸载策略以及基于路边单元辅助的用户协作机制的有效性，本节分别仿真了使用二进制卸载的 MDOLTO 算法与取消基于

路边单元辅助的用户协作机制的算法进行对照, 分别简称为 B-MDOLTO 算法与 NR-MDOLTO 算法。仿真结果如图 4-6 所示。

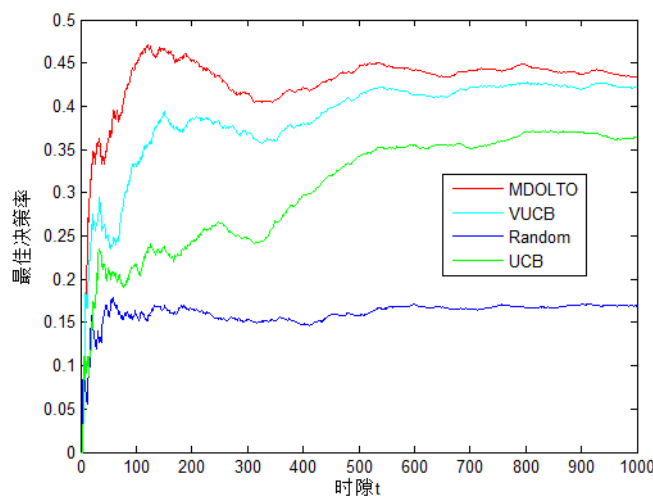


图 4-5 各算法的最佳决策率

Figure 4-5 Best Decision Ratio of Each Algorithm

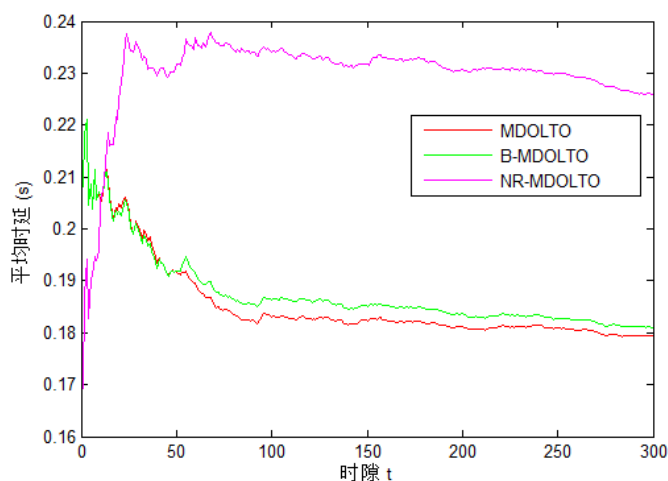


图 4-6 不同机制下 MDOLTO 算法的平均时延

Figure 4-6 The Average Delay of MDOLTO Algorithm with Different Mechanism

从图 4-6 中可以看出, 部分卸载策略对降低时延约束起较为关键的作用, 这是因为部分卸载能够将任务进行分割, 然后在 TaV 与 SeV 中进行并行运算。为了详细验证基于路边单元辅助的用户协作机制对动态环境的适应能力, 在仿真中采用了 4.3.2 小节中关于时段的设定, 其中一个时段由 50 个时隙构成。从图中可以看出, 在第二个时段开始时, 具备基于路边单元辅助的用户协作机制比不具备该机制的算法具有更低的平均任务时延, 这说明了该机制有助于提高 RSU 通信范围内 TaV 的学习效率, 从而提高对动态环境的适应性。

图 4-7 验证了置信度表达式中 η 的有效性。 η 是一个用于调整算法对探索策略倾向程度的权重参数。从图中可以看出, 在初始的 180 个时隙内, $\eta = 0$ 的算

法与 $\eta = 0.2$ 的算法具有相同的性能，而在接下来的 800 个时隙里， $\eta = 0$ 的算法的累积遗憾增长得很快。这是因为 $\eta = 0$ 意味着算法不会进行探索，而是采取贪婪策略，每次都选取已知回报值最大的 SeV 进行任务卸载，从而陷入了一个次优解，与理论分析的情况一致。

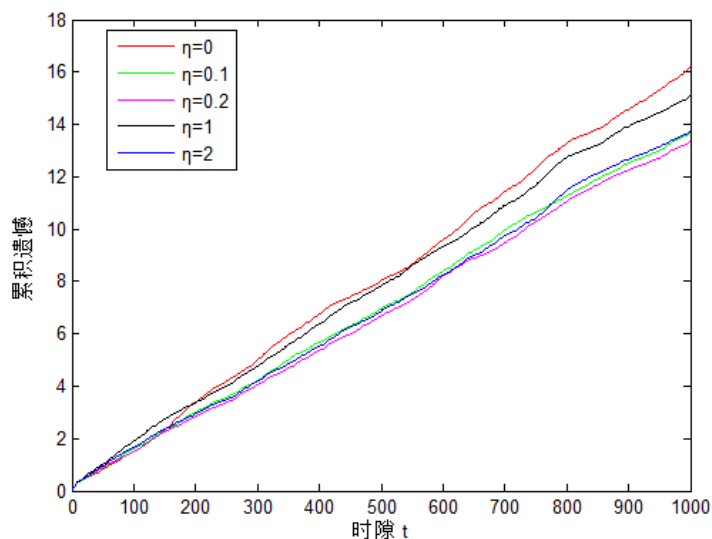


图 4-7 不同权重参数下 MDOLTO 算法的性能

Figure 4-7 The Performance of MDOLTO Algorithm with Different Weight Factor

4.5 本章小结 (Conclusions of the Chapter)

针对车辆移动性导致的 VEC 网络中链路稳定性差与网络拓扑结构快速变化的问题，本章将动态 VEC 网络中的任务卸载问题建模成了多臂赌博机问题，并提出了基于多臂赌博机的分布式在线学习任务卸载算法，对网络内用户的平均时延进行优化。该算法采用部分任务卸载，充分利用了本地设备的计算资源，并通过并行计算大大降低了任务时延。此外，算法根据任务卸载的特性与动态的网络环境设计一个效用函数，有效地实现了探索与利用策略的平衡。为了增强算法对动态网络环境的适应性，本章还提出了基于路边单元辅助的用户协作机制，使用户间可以共享学习信息，提升了算法的效率。

5 基于公平匹配 MAB 的动态 VEC 任务卸载

5 Task Offloading of Dynamic Vehicular Edge Computing Network based on Fair Matching MAB

5.1 引言 (Introduction)

第四章针对 VEC 网络状态信息快速变化的问题,提出了一种基于多臂赌博机的分布式在线学习任务卸载算法,使车辆能够自适应地学习网络状态信息,从而不断优化自身卸载决策。这样的分布式任务卸载方案不需要用户间进行沟通,每个用户都选择使自身收益最大化的策略,从而减少了用户的信令开销。但这也带来了两个问题:一是用户间可能出现资源竞争的情况,二是网络中用户的公平性难以保证,即有些用户占用了超出自身需求的资源,而有些用户则无法获得足够的资源。

针对 VEC 网络中分布式任务卸载策略产生的资源竞争与用户公平性难以保证的问题,本章提出了基于公平匹配 MAB^[84]的任务卸载方案。在现实网络中,满足所有用户的 QoS 需求有时比最大化某项指标更为重要。因此,本文提出的基于公平匹配 MAB 的任务卸载方案不再像传统的 MAB 算法那样追求收益的最大化,而是以满足用户的 QoS 需求为目标进行任务卸载,从而保障用户的公平性。为了解决资源竞争问题,方案建立了一个 TaV 与 SeV 间的匹配对, TaV 将按照匹配结果选择对应的 SeV 进行任务卸载。本章以用户的时延约束为 QoS 需求,构建了一个长期时延约束下用户能耗最小化的优化问题,并提出了一种基于李雅普诺夫优化的动态任务卸载算法。该算法利用李雅普诺夫优化理论,将能耗优化问题分解成了任务卸载决策子问题与卸载功率分配子问题,并分别通过基于公平匹配 MAB 的任务卸载算法与第三章提出的基于 DC 规划的功率优化算法进行求解。仿真结果表明,本章所提算法能够有效的解决系统中的资源竞争与用户公平性问题,提高了系统的资源利用率。

5.2 系统模型 (System Model)

如图 5-1 所示,本章考虑一个多用户的 VEC 应用场景。在系统模型中, TaV 通过 V2V 链路向 SeV 进行任务卸载。为了体现 VEC 系统的时变特性,本章将系统中的时间划分为 T 个离散的时隙,每个时隙的编号用 t 表示, $t = \{1, 2, \dots, T\}$ 。设 SeV 的数量为 N , 每个 SeV 的编号用 n 表示, SeV 的集合用 \mathbf{N} 表示, $n \in \mathbf{N} = \{1, 2, \dots, N\}$, TaV 的数量为 M , 每个 SeV 的编号用 m 表示, TaV 的集合用 \mathbf{M} 表示, $m \in \mathbf{M} = \{1, 2, \dots, M\}$ 。考虑到 VEC 系统中存在一些大型任务, TaV 将其

拆分为多个子任务进行运算，每个时隙处理一个子任务。子任务不可再划分，因此对每个子任务采用二进制任务卸载技术，即 TaV 将整个子任务卸载至 SeV 进行计算。每个时隙的子任务用 $\varphi_m(t)$ 表示， $\varphi_m(t)$ 由三个变量构成： $\varphi_m(t) = \{L_m(t), C_m(t), T_m(t)\}$ ，其中 $L_m(t)$ 表示计算任务的大小，单位为 bit， $C_m(t)$ 表示计算任务的复杂度，指计算 1 比特任务所需的 CPU 周期数，而 $T_m(t)$ 表示任务的时延约束。本模型假设任务的计算复杂度 $C_m(t)$ 对于 TaV m 来说是已知的^[86]，因此计算任务需消耗的 CPU 周期数可表示为 $L_m(t)C_m(t)$ 。由于计算任务无法分割，下面将从 SeV 选择、任务传输、任务计算以及结果回传四个方面对能耗与时延进行建模。

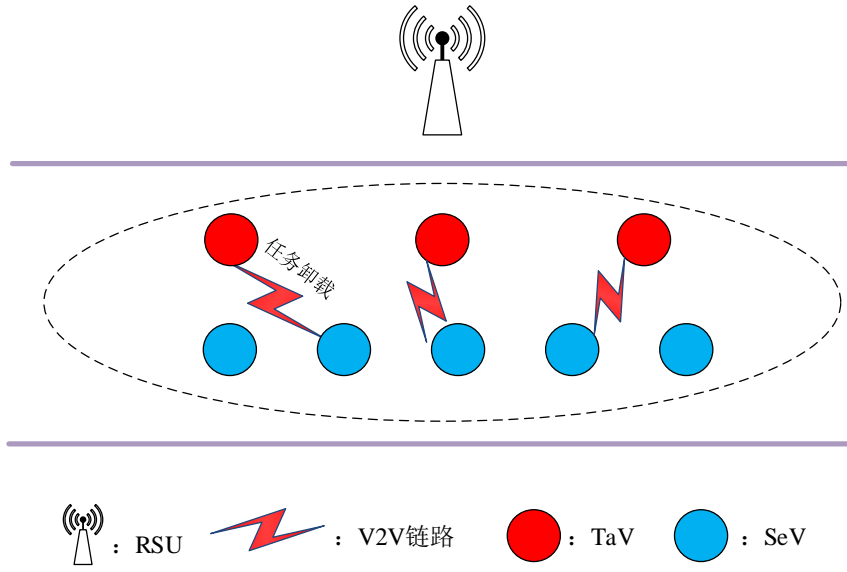


图 5-1 车辆边缘计算网络多用户任务卸载场景

Figure 5-1 Task Offloading Scenario of Multi-user VEC Network

5.2.1 任务卸载模型

(1) SeV 选择

因为本模型采用的是二进制任务卸载，所以 TaV 需要选择相应的 SeV 进行任务传输与计算。用二进制变量 $\beta_{m,n}(t) \in \{0,1\}$ 表示 TaV m 对 SeV n 的选择情况，如果 TaV m 在时隙 t 选择了 SeV n ，则 $\beta_{m,n}(t) = 1$ ，反之 $\beta_{m,n}(t) = 0$ 。由于 TaV 在一个时隙只能选择一个 SeV 进行任务卸载，因此对 $\beta_{m,n}(t)$ 有以下约束：

$$\sum_{n \in N} \beta_{m,n}(t) = 1 \quad (5-1)$$

(2) 任务传输

完成 SeV 选择后，TaV 进行任务传输。由于 VEC 网络的高度动态性，模型中假设信道增益在每个时隙保持不变，在不同时隙持续发生变化。记 TaV m 与 SeV n 间的上行链路信道增益为 $h_{m,n}^o(t)$ ，TaV m 的发射功率为 $P_m(t)$ ，信道带宽为 B ，则任务传输速率 $r_{m,n}^o(t)$ 可表示为：

$$r_{m,n}^o(t) = B \log_2 \left(1 + \frac{P_m(t) h_{m,n}^o(t)}{\sigma^2} \right) \quad (5-2)$$

其中 σ^2 为高斯白噪声的噪声功率。由于模型采用 OFDM 技术，因此不考虑同信道噪声干扰。

则 TaV m 在 t 时隙的任务传输时延 $d_{m,n}^o(t)$ 可表示为：

$$d_{m,n}^o(t) = \beta_{m,n}(t) \frac{L_m(t)}{r_{m,n}^o(t)} \quad (5-3)$$

TaV m 在 t 时隙的任务传输能耗 $E_{m,n}(t)$ 可表示为：

$$E_{m,n}(t) = P_m(t) d_{m,n}^o(t) \quad (5-4)$$

(3) 任务计算

任务传输至相应的 SeV 后，由 SeV 进行计算。记 SeV n 在时隙 t 的 CPU 工作频率为 $f_n(t)$ ，任务的计算时延 $d_{m,n}^c(t)$ 可表示为：

$$d_{m,n}^c(t) = \sum_{n \in N} \beta_{m,n}(t) \frac{L_m(t) C_m(t)}{f_n(t)} \quad (5-5)$$

(4) 结果回传

SeV 完成计算任务后，将计算结果回传至 TaV。记 $h_{n,m}^b(t)$ 为 SeV n 到 TaV m 的下行链路信道增益，SeV 的发射功率为 P_m ，则下行传输速率 $r_{n,m}^b(t)$ 可表示为：

$$r_{n,m}^b(t) = B \log_2 \left(1 + \frac{P_n h_{n,m}^b(t)}{\sigma^2} \right) \quad (5-6)$$

设计算结果的数据量为 $y_m(t)$ ，则计算结果的传输时延 $d_{n,m}^b(t)$ 可表示为：

$$d_{n,m}^b(t) = \frac{y_m(t)}{r_{n,m}^b(t)} \quad (5-7)$$

5.2.2 优化问题描述

通过上述建模，TaV m 完成任务的总时延 $d_{sum}(m,t)$ 可表示为：

$$d_{sum}(m,t) = d_{m,n}^o(t) + d_m^c(t) + d_{n,m}^b(t) \quad (5-8)$$

在动态的 VEC 环境中，SeV 的计算资源 $f_n(t)$ 处于动态变化的状态，这给任务卸载造成了困难。本章研究在任务的时延约束下最小化 T 个时隙内平均能耗的优化问题，优化变量为 SeV 选择函数 $\beta_{m,n}(t)$ 与发射功率 $P_m(t)$ 。优化问题可以表述为如下形式：

$$[\text{P3}] \quad \min_{\beta_{m,n}(t), P_m(t)} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \{ E_{m,n}(t) \} \quad (5-9a)$$

$$s.t. \quad \frac{1}{T} \sum_{t=1}^T \mathbb{E}\{d_{sum}(m,t)\} \leq \bar{T} \quad (5-9b)$$

$$\beta_{m,n}(t) \in \{0,1\} \quad (5-9c)$$

$$\sum_{n \in N} \beta_{m,n}(t) = 1 \quad (5-9d)$$

$$0 \leq P_m(t) \leq P_{\max} \quad (5-9e)$$

其中 $\mathbb{E}\{E_{m,n}(t)\}$ 表示为对 $E_{m,n}(t)$ 求期望。约束式(5-9b)为时延约束，由于 TaV 在每个时隙都需要完成一个计算任务，所以用 $\frac{1}{T} \sum_{t=1}^T \mathbb{E}\{d_{sum}(m,t)\}$ 来表示完成任务的

平均时延，而 $\bar{T} = \frac{1}{T} \sum_{t=1}^T T_m(t)$ 则表示任务的平均时延约束。约束式(5-9c)与(5-9d)为

SeV 选择约束，保证了 TaV 每个时隙只能选择一个 SeV 进行任务卸载。

优化问题 **P3** 是一个动态优化问题，动态变化的网络状态信息使(5-9a)与(5-9b)中的期望项难以求解。在解决动态优化问题时，系统的稳定通常十分重要，而李雅普诺夫优化方法则是保障系统稳定性的较好方法^[87]。因此，本章利用李雅普诺夫优化理论，将优化问题 **P3** 分解为任务卸载决策子问题与功率优化子问题，并分别对其进行求解。

5.3 基于公平匹配 MAB 的动态任务卸载算法 (Fair Matching MAB based Dynamic Task Offloading Algorithm)

5.3.1 基于 Lyapunov 优化的优化问题分解

根据李雅普诺夫理论，首先引入一个虚拟队列 $Q(t)$ 来重新构建长期时延约束 (5-9b)。 $Q(t)$ 表示为每一个时隙的时延队列积压，其更新过程表示如下：

$$Q(t+1) = \max\{Q(t) + d_{sum}(m,t) - \bar{T}, 0\} \quad (5-10)$$

定义一个二次 Lyapunov 函数：

$$L(\theta(t)) \triangleq \frac{1}{2} Q(t)^2 \quad (5-11)$$

为了保证队列的稳定性，接下来引入 Lyapunov 漂移函数^[88]。该函数表示 Lyapunov 函数从当前时隙到下一时隙的增长量，表达式如下：

$$\Delta(\theta(t)) \triangleq L(\theta(t+1)) - L(\theta(t)) \quad (5-12)$$

通过最小化漂移函数 $\Delta(\theta(t))$ ，就能够保持队列 $Q(t)$ 的稳定。根据 Lyapunov

漂移惩罚理论^[89], 在队列的稳定性中引入延迟性, 即定义一个漂移惩罚函数来同时求解漂移函数与目标函数的最小值, 就能够将问题 **P1** 化解成单时隙的优化问题。最小化漂移惩罚函数的表达式如下:

$$\min (\Delta(\theta(t)) + VE_{m,n}(t)) \quad (5-13)$$

其中 $V \geq 0$ 为控制权重, 用来调节对队列稳定性与目标函数的重视程度。 $\Delta(\theta(t))$ 中包含了 $L(\theta(t+1))$ 项, 根据 Lyapunov 漂移惩罚理论, 通过对式(5-9b)、(5-11)、(5-12)与(5-13)进行放缩变换, 可得以下不等式:

$$\Delta(\theta(t)) + VE_{m,n}(t) \leq H + Q(t)[d_{sum}(m,t) - \bar{T}] + VE_{m,n}(t) \quad (5-14)$$

其中 H 是一个与 $T_m(t)$ 有关的常数。

利用 Lyapunov 漂移惩罚理论得到漂移惩罚函数的上界之后, 原问题 **P3** 就可以被近似转化为求解该上界的最小值问题, 转化后表达式如下:

$$[\mathbf{P3.1}] \quad \min_{\beta_{m,n}(t), P_m(t)} H + Q(t)[d_{sum}(m,t) - \bar{T}] + VE_{m,n}(t) \quad (5-15a)$$

$$s.t. \quad \frac{1}{T} \sum_{t=1}^T \mathbb{E}\{d_{sum}(m,t)\} \leq \bar{T} \quad (5-15b)$$

$$\beta_{m,n}(t) \in \{0,1\} \quad (5-15c)$$

$$\sum_{n \in N} \beta_{m,n}(t) = 1 \quad (5-15d)$$

$$0 \leq P_m(t) \leq P_{\max} \quad (5-15e)$$

接下来对问题 **P3.1** 进行分解。根据文献[73]提出的分解理论, 可知:

$$\inf_{x,y} f(x,y) = \inf_x f^*(x) \quad (5-16)$$

其中 $f^*(x) = \inf_y f(x,y)$ 。

因此, 可以将问题 **P3.1** 进行分解为任务卸载决策子问题与功率优化子问题, 分别对优化变量 $\beta_{m,n}(t)$ 与 $P_m(t)$ 进行求解。任务卸载决策子问题表示如下:

$$[\mathbf{P3.2}] \quad \min_{\beta_{m,n}(t)} Q(t)(d_{m,n}^c(t) + d_{n,m}^b(t)) \quad (5-17a)$$

$$s.t. \quad \beta_{m,n}(t) \in \{0,1\} \quad (5-17b)$$

$$\sum_{n \in N} \beta_{m,n}(t) = 1 \quad (5-17c)$$

设 **P3.2** 的最优解表示为 U_β^* , 则功率优化子问题可表示为:

$$[\mathbf{P3.3}] \quad \min_{P_m(t)} (U_\beta^* + Q(t)d_{m,n}^o(t) + VE_{m,n}(t)) \quad (5-18a)$$

$$s.t. \quad 0 \leq P_m(t) \leq P_{\max} \quad (5-18b)$$

接下来分别对 **P3.2** 与 **P3.3** 进行求解。

5.3.2 基于公平匹配 MAB 的任务卸载

在动态的 VEC 网络环境中, SeV 的计算资源 $f_n(t)$ 处于时刻变化的状态, 这使得 TaV 难以进行任务卸载。MAB 算法可用于解决这个问题, TaV 通过在进行任务卸载的过程中学习附近 SeV 的服务能力, 能够不断地优化自身决策。然而在常规的 MAB 算法中, 用户往往追求自身的最大收益, 这在 VEC 网络中将会产生资源竞争与用户公平性难以保障的问题。为此, 本小节提出基于公平匹配 MAB 的任务卸载算法, 通过建立满足用户 QoS 需求的匹配对, 解决了上述问题。

在拓扑结构快速变化的 VEC 网络中, 集中式的调度难以实现, 因此本算法采用了一个完全分布式的卸载策略, 即用户节点无需与其他节点进行信令交互。为了使 TaV 在彼此不沟通的情况下察觉到资源竞争的情况出现, 算法规定当 SeV 被多个 TaV 选中时, SeV 会拒绝为所有 TaV 提供服务。整个算法分为探索, 匹配, 利用三个步骤, 具体表述如下:

(1) 探索

与传统的 MAB 算法相同, 算法首先进行探索 (explore)。在给定的时隙 T_e 内, TaV 会在每个时隙开始时随机选择一个 SeV n 进行任务卸载, 并记录任务完成的时间 $d_{m,n}^c(t)$ 与其他相关信息。为了选择能够满足 TaV 时延 QoS 需求的 SeV, 算法将 SeV 的服务能力 $u_{n,t}$ 定义为处理 1 比特任务所消耗的时间, $u_{n,t}$ 表达式如下:

$$u_{n,t} = \frac{d_{m,n}^c(t)}{L_m(t)} \quad (5-19)$$

为了在动态的 VEC 网络环境中实现探索与利用策略的均衡, 算法沿用了第四章中所提出的效用函数(4-12), 效用函数表达式如下:

$$\hat{u}_{n,t} = \bar{u}_{n,t-1} - \sqrt{\frac{\eta \tilde{L}_t v_{n,t} \ln(t - t_n)}{k_{n,t-1}}} \quad (5-20)$$

其中 $\bar{u}_{n,t-1}$ 为 SeV 截止 $t-1$ 时隙的平均服务能力, η 为权重参数, $k_{n,t-1}$ 指 SeV n 截止 $t-1$ 时隙被选择的次数, t_n 指 SeV n 第一次被选择的时隙, \tilde{L}_t 为任务量 L_t 的归一化值, $v_{n,t}$ 值 SeV n 服务能力的方差。 \tilde{L}_t 的表达式由式(4-13)给出, $v_{n,t}$ 的表达式如下:

$$v_{n,t} = \min\left(\frac{1}{4}, \sum_{\tau=1}^{t-1} \frac{1}{k_{n,t-1}} k_{n,\tau} u_{n,\tau}^2 - u_{n,t-1}^2 + \sqrt{\frac{\eta \tilde{L}_t \ln(t-t_n)}{k_{n,t-1}}}\right) \quad (5-21)$$

其中, $u_{n,\tau}$ 为 SeV 在 τ 时隙的服务能力。

用 $\eta_{n,m}(t)$ 来表示 SeV n 是否发生了碰撞, $\eta_{n,m}(t)=1$ 表示没有发生碰撞, $\eta_{n,m}(t)=0$ 表示发生了碰撞。在 T_e 结束时, TaV 通过式(5-20)评估 SeV 的服务能力。探索的具体步骤如算法 5.1 所示。

算法 5.1: 探索算法

输入: 探索持续的时隙 T_e , 起始时隙 T_0

输出: 各 SeV 的效用函数 $\hat{u}_{m,t}$

1. **for** $t = T_0, \dots, T_0 + T_e$ **do**
 2. 在 N 个 SeV 中随机选取一个 SeV n 进行任务卸载
 3. 获得该 SeV n 的时延 $d_{m,n}^c(t)$ 与碰撞信息 $\eta_{n,m}(t)$
 4. **if** $\eta_{n,m}(t) = 1$
 5. $k_{n,t} = k_{n,t} + 1$
 6. **if** $k_{n,t} = 1$
 7. $t_n = t$
 6. **end if**
 7. **end if**
 9. 根据式(5-19)计算每个 SeV n 的服务能力 $u_{n,t}$
 10. **end for**
 14. 根据式(5-20)计算每个 SeV n 的效用函数 $\hat{u}_{n,t}$
-

(2) 匹配

为了避免资源竞争并保证用户的公平性, 本算法根据 TaV 的 QoS 需求对 TaV 与 SeV 进行匹配, TaV 根据匹配结果选择 SeV 进行任务卸载。给定匹配持续的时隙 T_m , TaV 首先以自身时延约束 $T_n(t)$ 为 QoS 需求, 将能够满足自身需求的 SeV 纳入可选的 SeV 集 ε_m 。模型中假设车辆具备无线信道检测的能力^[90], 且 TaV 在判断时假设车辆通信都采用最大功率 P_{\max} , 因此任务传输与结果回传的时间 $d_{m,n}^o(t)$ 与 $d_{n,m}^b(t)$ 已知。根据任务量大小 $L_m(t)$ 与时延约束 $T_m(t)$, 可知 ε_m 中的 SeV 需满足下述不等式:

$$\hat{u}_{n,t} \leq \frac{T_m(t) - d_{m,n}^o(t) - d_{n,m}^b(t)}{L_m(t)} \quad (5-21)$$

在每个时隙开始时, TaV m 在 ε_m 中随机选取一个 SeV $a_n(t)$ 进行任务卸载。如果发生了碰撞, 则 TaV m 会再选择 ε_m 中的其他 SeV。若没有发生碰撞, 则在接下来的时隙内, TaV m 都会保持自身选择不变。用 $\tilde{\rho}$ 来表示所有 TaV 中是否存在碰撞, $\tilde{\rho} = 1$ 表示此时匹配中不存在碰撞, $\tilde{\rho} = 0$ 表示此时匹配中发生了碰撞。在 T_m 结束时, 若 $\tilde{\rho} = 1$, 则说明匹配已成功建立, 记录此时的匹配对 \tilde{a}_m , TaV 将按照 \tilde{a}_m 进行任务卸载, 并按照式(5-20)更新 $\hat{u}_{n,t}$ 。若 $\tilde{\rho} = 0$, 则说明发生了碰撞, 存在 TaV 没能找到合适的 SeV。此时, 该 TaV 会将所有的 SeV 依次选择一遍, 与所有的 TaV 发生碰撞, 以促使系统重新进行探索与匹配步骤。这种做法能够保证系统中每个 TaV 的服务质量, 提升系统的公平性。由于匹配的过程可以表示为一个吸收马尔可夫链^[90], 因此只要匹配存在, 在匹配持续时隙 T_m 较长的情况下就能收敛到该状态。匹配的具体步骤如下所示:

算法 5.2: 匹配算法

输入: 匹配持续的时隙 T_m , 起始时隙 T_0 , 各 SeV 的效用函数 $\hat{u}_{n,t}$

输出: 匹配对 \tilde{a}_m

1. TaV m 根据式(5-21)建立可选的 SeV 集 ε_m
 2. TaV m 在 ε_m 中随机选择一个 SeV $a_m(t)$
 3. **for** $t = T_0, \dots, T_0 + T_m$ **do**
 4. **if** $\beta_{n,m}(t) = 1$
 5. TaV 维持自身选择不变, 即 $a_m(t+1) = a_m(t)$, 更新 $\hat{u}_{a_m(t),t}$
 6. **else**
 7. 在 ε_m 中随机选择另一个 SeV $a_m(t+1)$, 更新 $\hat{u}_{a_m(t),t}$
 8. **end if**
 9. **end for**
 10. **if** $\tilde{\rho} = 0$
 11. 选择 $a_m(t) = 1, 2, \dots, N$ 进行任务卸载
 12. **end if**
 13. 记录匹配状态 $\tilde{\rho}$ 与匹配对 \tilde{a}_m
-

(3) 利用

若匹配步骤成功建立了匹配对, 则利用步骤就会按照构建好的匹配对 \tilde{a}_m 进行任务卸载。若匹配对没有建立成功, 则需要重新进行探索与匹配步骤。由于 VEC 网络中的拓扑结构与 SeV 的计算资源处于快速变化的状态, 因此构建好的匹配对 \tilde{a}_m 可能无法一直维持下去。当网络状态发生变化致使原有的匹配对无法维持时, 就需要重新构建匹配对。为了适应网络的动态性, 算法将 T 个时隙分成 K 个周期, 当原先的匹配对失效时, 算法会进入到下一个周期, 如算法 5.3 所示。

算法 5.3: 基于公平匹配 MAB 的任务卸载算法

输入: 各 TaV 的时延约束 T_e

输出: 匹配对 \tilde{a}_m

```

1. for  $k = 1, \dots, K$ 
2.    $T_e = \lceil N \log(k+1) \rceil$ , 依据算法 5.1 进行探索, 得到各 SeV 的效用函数  $\hat{u}_{n,t}$ 
3.    $T_m = \lceil N \log(k+1) \rceil$ , 依据算法 5.2 进行匹配, 得到匹配对  $\tilde{a}_m$ 
4.   令  $d \leftarrow run$ 
5.   while  $d = run$  do
6.     if  $\eta_{m,\tilde{a}_m}(t) = 1$ 
7.       按照匹配对  $\tilde{a}_m$  进行任务卸载, 更新  $\hat{u}_{\tilde{a}_m,t}$ 
8.     else
9.        $d \leftarrow terminate$ 
10.    end if
11.    if  $\hat{u}_{n,t} > \frac{T_m(t) - d_{m,n}^o(t) - d_{n,m}^b(t)}{L_m(t)}$  且  $d = run$ 
12.       $d \leftarrow signal$ 
13.    end if
14.    if  $d = signal$ 
15.      依次选择  $a_m(t) = 1, 2, \dots, N$  进行任务卸载
16.    end if
17.  end while
18.end for

```

在算法 5.3 中，匹配对目前的状态， d 一共有三个状态，分别是 **run**，**signal** 与 **terminate**。当 $d = \text{run}$ 时，表示该匹配对正常运行，每个 TaV 都对自己的匹配对象满意。而在利用步骤执行的过程中，TaV 发现式(5-21)不再成立时，意味着它发现自己的匹配对象不再能够满足 QoS 需求，就会将状态切换至 $d = \text{signal}$ ，要求重新匹配。为了重新进行匹配，此 TaV 将依次对所有 SeV 进行任务卸载，与所有 TaV 产生碰撞。而当 TaV 感受到碰撞时，会将状态切换至 $d = \text{terminate}$ 。此时该周期结束，开始下一个周期。

5.3.3 基于凸差规划的功率优化

由于 SeV 的发射功率是定值，且车辆具备无线信道检测的能力，因此 **P3.3** 中的 U_β^* 项为常数。**P3.3** 的优化目标(5-18a)可详细写为以下形式：

$$\min_{P_n(t)} \left(Q(t) \frac{L_n(t)}{r_{n,m}^o(t)} + VP_n(t) \frac{L_n(t)}{r_{n,m}^o(t)} \right) \quad (5-22)$$

式(5-22)与第三章中的式(3-12)具有相同的形式，即存在同样的 $P_n(t)$ 与 $r_{n,m}^o(t)$ 耦合的情况，因此可使用第三章提出的基于凸差规划的优化算法将 **P3.3** 转化为凸函数进行求解。具体步骤如下：

首先引入辅助变量 $Z_{m,n}$ ，根据凸优化理论^[62]， $\frac{y^2}{x}$ 的形式是凸的，而且由于

P3.3 为一个最小化的优化问题，所以用 $Z_{m,n}^2$ 表示优化目标的上限，表达式如下：

$$L_m(t)(Q(t) + VP_m(t)) \frac{1}{r_{m,n}^o(t)} \leq Z_{m,n}^2 \quad (5-23)$$

再将式(5-23)进一步转化成分式的形式：

$$\frac{1}{r_{n,m}^o(t)} - \frac{Z_{m,n}^2}{L_n(t)(Q(t) + VP_n(t))} \leq 0 \quad (5-24)$$

式(5-24)为 DC 约束。接下来将 DC 约束中的第二项线性化，以建立凸近似的可行解。给定起始点 $Z_{m,n}^0$ 与 $P_n^0(t)$ ，对 DC 约束的第二项进行泰勒展开：

$$\frac{1}{r_{n,m}^o(t)} - \frac{1}{L_n(t)} \left(\frac{(Z_{m,n}^0)^2}{Q(t) + VP_n^0(t)} + \frac{2Z_{m,n}^0(Z_{m,n} - Z_{m,n}^0)}{Q(t) + VP_n^0(t)} - \frac{(Z_{m,n}^0)^2(P_n(t) - P_n^0(t))}{(Q(t) + VP_n^0(t))^2} \right) \leq 0 \quad (5-25)$$

此时，式(5-25)为凸约束。由于 $Z_{m,n}^2$ 表示优化目标式（5-22）的上限，因此 **P3.3** 可进一步转化为如下形式：

$$[\text{P3.4}] \quad \min_{P_n(t), Z_{m,n}} (Z_{m,n})^2 \quad (5-26a)$$

$$\text{s.t. } 0 \leq P_n(t) \leq P_{\max} \quad (5-26b)$$

(5-25)

由凸优化理论可知, **P3.4** 是一个凸问题。因此, **P3.4** 可以用拉格朗日对偶分解法进行求解。具体解法在本文 3.3 节给出, 在此不做赘述。求解完 **P3.4** 后得到起始点 $Z_{m,n}^0$ 与 $P_m^0(t)$ 下的最优解 $Z_{m,n}^*$ 与 $P_m(t)^*$ 。接下来, 将这些变量作为新的起始点 $Z_{m,n}^1$ 与 $P_m^1(t)$ 再带入 **P3.4** 进行求解。当最优解 $Z_{m,n}^*$ 与 $P_m(t)^*$ 收敛时, 就得到了 **P3.3** 关于给定 U_β^* 的最优解。

综上所述, 本章所提算法首先利用李雅普诺夫优化理论将优化问题 **P3** 转化成了单时隙优化问题 **P3.1**。然后对每个时隙的 **P3.1** 进行求解, 就能够得到 **P3** 的近似最优解。**P3.1** 的求解步骤为: 首先利用分解理论将其分解为任务卸载子问题 **P3.2** 与功率优化子问题 **P3.3**, 然后利用基于公平匹配 MAB 的任务算法与基于凸差规划的功率优化算法分别对 **P3.2** 与 **P3.3** 进行求解。求解完成后, 按照式(5-10)更新虚拟队列 $Q(t)$, 代入下一个时隙的优化问题 **P3.1** 进行求解。

5.3.4 算法性能分析

本节主要对基于公平匹配 MAB 的动态任务卸载算法的表现与计算复杂度进行分析。

首先, 文献[84]指出, 在公平匹配 MAB 算法中, 由于用户综合利用了探索阶段与开发阶段学习到的信息, 用户对摇臂的估计会变得更准确。这使得算法建立的匹配对较为牢固, 即算法的利用阶段持续时间会变得足够长, 最终使基于公平匹配 MAB 的动态任务卸载算法的遗憾达到了 $O(1)$ 。算法的遗憾为 $O(1)$, 意味着长期时延约束(5-9b)总是能被满足, 说明基于公平匹配 MAB 的动态任务卸载算法能够有效地解决优化问题 **P3**。

然后对算法的计算复杂度进行分析。在基于公平匹配 MAB 的任务卸载算法中, 算法在每个时隙都需要计算一次效用函数 $\hat{u}_{m,t}$, 其计算复杂度为 $O(1)$ 。在基于凸差规划的功率优化算法中, 首先需要给定初始点, 将 **P3.3** 转化为 **P3.4**, 然后利用拉格朗日对偶分解法求解 **P3.4** 的最优解。接着将最优解作为新的初始点代入 **P3.3** 进行循环迭代, 直到最优解收敛为止。设拉格朗日对偶分解法求解 **P3.4** 需要的迭代次数为 I_1 , 每次求解计算复杂度为 $O(X)$, 求解 **P3.3** 需要的迭代次数为 I_2 , 则基于凸差规划的功率优化算法的计算复杂度为 $O(I_1 I_2 X)$ 。整个算法需要执行 T 个时隙, 因此基于公平匹配 MAB 的动态任务卸载算法的计算复杂度为 $O(I_1 I_2 XT)$ 。

5.4 仿真结果与分析 (Simulation Results and Analysis)

为了验证算法的性能, 本仿真考虑系统中有 5 个 TaV 与 10 个 SeV, SeV 随机分布在 TaV 周围 50m 的范围内, 且 TaV 与 SeV 的距离在每个时隙内随机变

化，变化范围为-5m 到 5m。SeV 的空闲计算资源占自身总资源的比重随机分布在 30%~50%之间，计算结果的任务量 $y_m(t) = 0.2L_m(t)$ ，其余的仿真参数如表 5-1 所示。

表 5-1 多用户动态 VEC 系统参数配置表

Table 5-1 Parameters Assignment Table of Multi-user Dynamic VEC System

参数	值
最大传输功率 P_{\max}	0.1W
高斯白噪声功率 σ^2	10^{-13}W
带宽 B	10MHz
计算复杂度 $C_m(t)$	10^3 cycles/bit
权重参数 η	0.2
SeV 的 CPU 最大工作频率 f_{\max}^s	5GHz
TaV 的通信距离	50m
任务量 $L_m(t)$	0.5~0.8Mbits
平均时延约束 \bar{T}	0.3s
控制权重 V	100

参照文献[81]中对城市交通的 V2V 通信链路研究，仿真中的信道增益建模如下：

$$h_{n,t}^o = h_{n,t}^b = A_0 l^{-\varphi} \quad (5-27)$$

其中 $A_0 = -17.8\text{dB}$ ， l 为 TaV 与 SeV 的距离， φ 为路径损耗因子，随机分布于 $[2, 4]$ 。

为了验证本章算法对网络公平性与资源竞争状况的改善，本节引入了 UCB 算法，拥有网络全局信息的 Optimal 算法与 Random 算法进行对照。对仿真中的对照算法具体介绍如下：

(1) LUCB 算法，指先利用 Lyapunov 优化将优化目标分解，然后用传统的 UCB 算法求解任务卸载决策子问题的算法。传统的 UCB 算法会驱使 TaV 选择服务能力最强的 SeV，而不考虑网络中的公平性与资源竞争的问题。

(2) Optimal 算法，即最佳决策算法。该算法拥有网络的实时全局信息，能够建立严格遵守时延约束的匹配对，并最大程度地优化 TaV 的能耗，其性能是所有算法性能的上界。

(3) DTOAFMM 算法，即本章在 5.3 节提出基于公平匹配 MAB 的动态任务卸载算法。

(4) Random 算法，即随机算法。该算法中的 TaV 随机选取 SeV 进行任务卸载。

为了验证算法对网络中用户公平性与资源竞争现象的改善,图 5-2 与图 5-3 分别展示了各算法中 TaV 的平均时延与平均能耗随时间的变化。从图中可以看出,本文所提的算法相较于 LUCB 算法具有更低的时延与能耗,且能够渐进收敛到最优解。这说明本章所提算法建立的匹配随时间的推移会趋于稳定,且能够满足所有 TaV 的 QoS 需求,改善网络中用户公平性与资源竞争现象。LUCB 算法相较于随机算法具有较低的时延与能耗,说明 MAB 算法能够有效地适应动态的 VEC 网络环境。LUCB 算法的性能不及本文提出的 DTOAFMM 算法,这是因为 LUCB 算法驱使 TaV 选择服务能力最强的 SeV,从而产生了资源竞争的情况。资源竞争引发的碰撞使部分 TaV 无法寻找符合自身 QoS 需求的 SeV,因此造成了系统平均时延与能耗的增加。

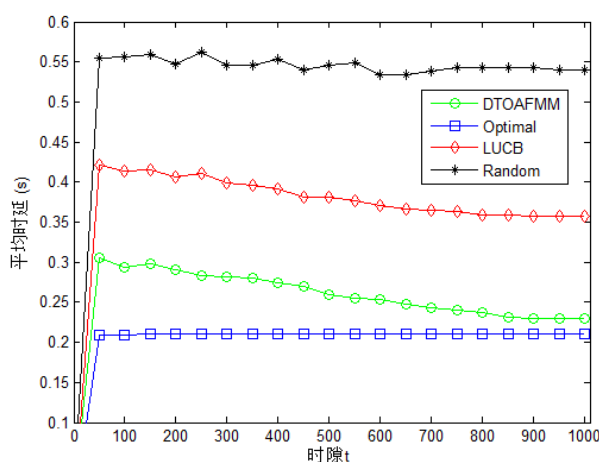


图 5-2 各算法的平均时延

Figure 5-2 Average Delay of Each Algorithm

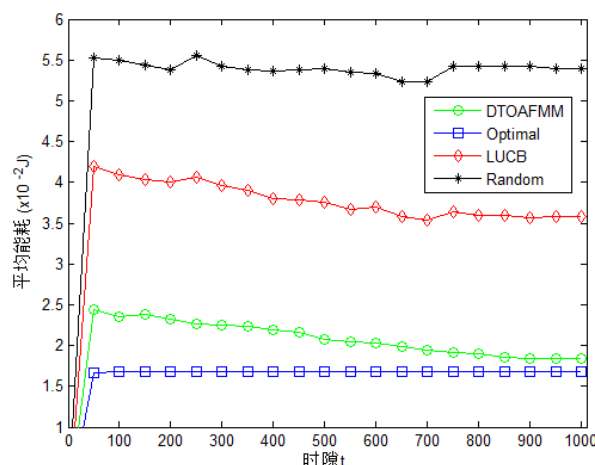


图 5-3 各算法的平均能耗

Figure 5-3 Average Energy Consumption of Each Algorithm

为了探究算法中基于凸差规划的功率优化在能耗优化中的作用,本节仿真了不同时延约束下 1000 个时隙的平均能耗,如图 5-4 所示。从图中可以看出,随

随着时延约束逐渐宽松，算法的平均能耗有所降低。这是因为宽松的时延约束减小了虚拟队列 $Q(t)$ 的累积，算法不再需要提高传输功率来减少时延，而是选择合适传输功率，从而减少了能耗。

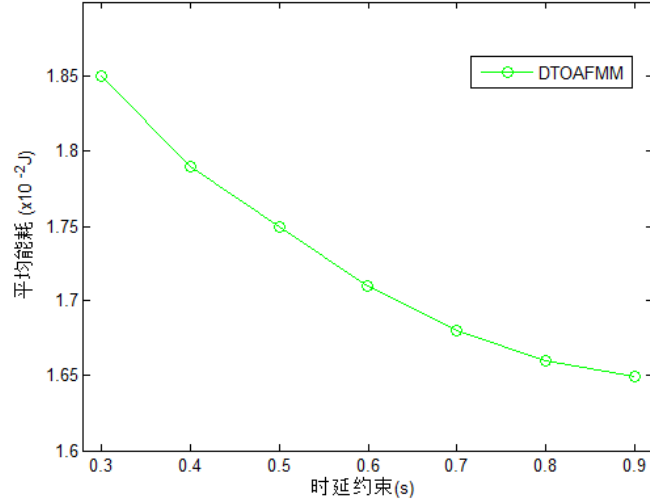


图 5-4 不同时延约束下的平均能耗

Figure 5-4 Average Energy Consumption with Different Energy Consumption

为了探究控制权重 V 对算法的影响，本节仿真了不同 V 值下所提算法的平均时延与能耗，分别如图 5-5 与图 5-6 所示。从图中可以看出，随着 V 值的增加，算法的平均能耗有所降低，平均时延有所增加。这说明 V 值越大，算法就越重视能耗的优化，代价则是时延的增加。通过调节 V 值，能够实现时延与能耗间的均衡。此外，从图 5-6 中可以发现，随着 V 值的增加，算法的收敛速度降低。这意味着控制权重 V 还能够实现优化目标函数值与收敛速度的均衡。

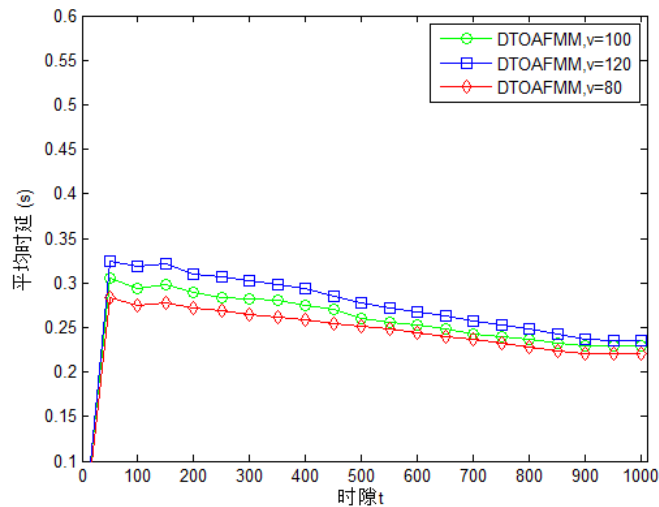


图 5-5 不同控制权重参数下的平均时延

Figure 5-5 Average Delay with Different Weight Parameter

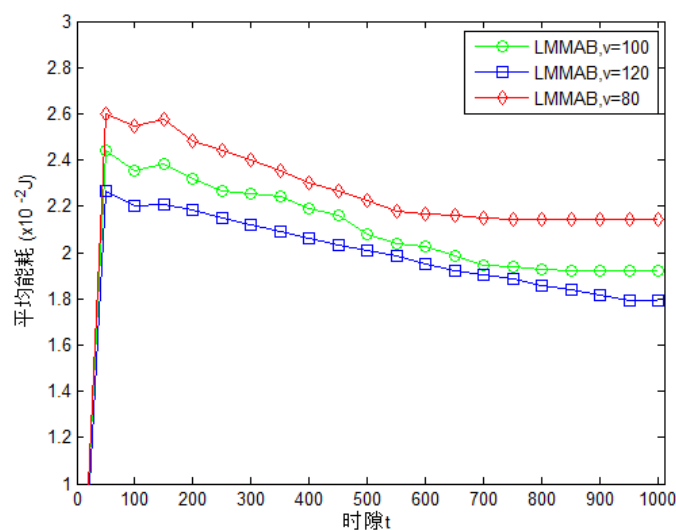


图 5-6 不同控制权重参数下的平均能耗

Figure 5-6 Average Energy Consumption with Different Weight Parameter

5.5 本章小结 (Conclusions of the Chapter)

本章针对 VEC 网络中分布式任务卸载策略存在的资源竞争与用户公平性难以保证的问题, 提出了基于公平匹配 MAB 的动态任务卸载方案。区别于传统的 MAB 算法追求最高收益, 公平匹配 MAB 算法以满足用户 QoS 需求为目标进行任务卸载, 从而能够避免用户间进行资源竞争的情况。本章以用户的时延约束为 QoS 需求, 构建了用户平均能耗最小化的优化问题。为了解决动态 VEC 网络下的长期优化问题, 本章通过李雅普诺夫优化理论将原问题解耦为任务卸载决策子问题与功率分配子问题进行求解。针对任务卸载子问题, 本章采用了基于公平匹配 MAB 的算法进行求解。针对功率分配子问题, 本章采用了在第三章提出的基于凸差规划的优化算法进行求解。最后, 本章通过仿真验证了算法的有效性。

6 总结与展望

6 Conclusions and Prospects

6.1 总结 (Conclusions)

随着车辆智能化水平的不断提高,一系列计算密集型与时延敏感型的应用得到了快速的发展。面对诸如自动驾驶之类的应用,仅靠车辆自身的有限资源无法满足其 QoS 需求。MEC 通过将云端的服务器下放至网络的边缘,从而能够在近距离,低延迟地为用户提供计算服务。因此,将 MEC 技术引入车联网能极大地缓解上述应用带来的压力。然而,边缘服务器相较于云服务器来说,资源同样是有限的,在交通流量较大时可能无法满足所有用户的计算资源。此外,车辆节点固有的移动性使网络的拓扑结构与通信链路处于快速变化的状态,这给任务卸载决策造成了额外的挑战。本文针对上述问题,研究了车辆边缘计算网络中的资源分配与任务卸载问题,主要工作内容如下:

(1) 针对车联网中交通流量较大时产生的系统资源受限问题,本文提出了基于通信与计算资源联合优化的资源分配策略。通过引入动态电压调节技术,使系统能够根据用户任务的时延约束以及信道状况,对车载计算单元工作频率、通信发射功率、车辆的任务卸载量以及边缘服务器的计算资源分配进行联合优化,从而最小化系统内用户的能耗。优化问题最终建模成了一个多变量耦合的非凸问题,本文采用凸差规划将其转化为两个凸的子问题,随后采用拉格朗日乘法法进行联合迭代求解。最后通过仿真实验验证了算法在能耗优化方面的有效性。

(2) 针对动态车联网中用户难以进行节点感知导致的任务卸载效率下降的问题,本文提出了基于多臂赌博机的分布式在线学习任务卸载策略。本文将任务卸载过程建模为一个多臂赌博机问题,使用户在进行任务卸载的同时能够学习到周围环境以及计算服务提供者的相关信息,从而利用学到的信息不断改善自身的决策。车辆高速移动使计算服务提供者处于动态变化的状态,这使得传统多臂赌博机算法失效。为了解决这个问题,本文提出了基于多臂赌博机的分布式在线学习算法,同时设计了基于路边单元辅助通信的用户协作机制,提高了系统内用户的学习效率。最后通过理论分析与仿真验证了所提算法能够更好地适应动态的 VEC 网络环境。

(3) 针对 VEC 网络中分布式任务卸载策略产生的资源竞争与用户公平性难以保证的问题,本章提出了基于公平匹配 MAB 的动态任务卸载方案。该方案以满足用户的 QoS 需求为目标进行任务卸载,从而保障了用户的公平性,避免了资源竞争情况的发生。优化问题建模为一个时延约束下的平均能耗最小化优化问

题。本文首先利用李雅普诺夫优化理论,将能耗优化问题分解成了任务卸载决策子问题与卸载功率分配子问题,然后分别通过基于公平匹配 MAB 的任务卸载算法与第三章提出的基于凸差规划的功率优化算法进行求解。仿真结果表明,本章所提算法能够有效地解决系统中的资源竞争与用户公平性问题,提高了系统的资源利用率。

6.2 展望 (Prospects)

本文针对车辆边缘计算网络中部分场景的资源分配与任务卸载问题进行了研究,存在一定的局限性以及不足之处,今后还可以从以下方向进行深入研究:

(1) 在对资源受限的 VEC 系统中进行资源分配问题的研究时,本文只考虑了具备一个边缘服务器的系统。而在现实中,可以考虑多个边缘服务器系统,引入负载均衡技术,在单个服务器复杂过大时允许其将计算负载分给周围的服务器,从而提升整个系统的服务能力。此外,考虑到网络内的智能设备越来越多,通过用户间协作的机制来充分利用网络中的空闲资源也是提高资源利用率的方法之一。

(2) 在对动态 VEC 系统中任务卸载问题的研究中,本文只考虑了用户间进行的 V2V 任务卸载,而未考虑用户向服务器进行任务卸载的情况。车辆节点高速移动使其在多个边缘服务器的服务范围内频繁切入切出,从而影响了任务卸载的效率。因此将车辆的移动性管理与移动预测纳入到任务卸载策略的制定中,将会有利于提高任务卸载的效率,是一个值得研究的方向。

参考文献

- [1] Kolosz B W, Grant-Muller S M. Appraisal and evaluation of interurban ITS: A European survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 16(3): 1070-1087.
- [2] Ullah A, Yao X, Shaheen S, et al. Advances in position based routing towards ITS Enabled FoG-Oriented VANET—A Survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(2): 828-840.
- [3] 赵娜, 袁家斌, 徐晗. 智能交通系统综述 [J]. 计算机科学, 2014, 41(11): 7-11.
- [4] Ji B, Zhang X, Mumtaz S, et al. Survey on the internet of vehicles: Network architectures and applications[J]. IEEE Communications Standards Magazine, 2020, 4(1): 34-41.
- [5] 肖瑶, 刘会衡, 程晓红. 车联网关键技术及其发展趋势与挑战 [J]. 通信技术, 2021, 54(1): 1-8.
- [6] Zhang J, Letaief K B. Mobile edge intelligence and computing for the internet of vehicles[J]. Proceedings of the IEEE, 2019, 108(2): 246-261.
- [7] 吕品, 许嘉, 李陶深, 徐文彪. 面向自动驾驶的边缘计算技术研究综述 [J]. 通信学报, 2021, 42(03): 190-208.
- [8] 周忠, 周颐, 肖江剑. 虚拟现实增强技术综述[J]. 中国科学: 信息科学, 2015, 45(02): 157-180.
- [9] 余韵, 连晓灿, 朱宇航, 谭国平. 增强现实场景下移动边缘计算资源分配优化方法[J]. 计算机应用, 2019, 39(01): 22-25.
- [10] Abbas N, Zhang Y, Taherkordi A, et al. Mobile edge computing: A survey[J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.
- [11] Olariu S. A survey of vehicular cloud research: Trends, applications and challenges[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(6): 2648-2663.
- [12] 邓聃婷, 滕飞, 杨燕. 云计算平台中面向车联网应用的能耗感知调度算法[J]. 计算机科学, 2016, 43(03): 44-48.
- [13] Lin C C, Deng D J, Yao C C. Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units[J]. IEEE Internet of Things Journal, 2017, 5(5): 3692-3700.
- [14] Taleb T, Samdanis K, Mada B, et al. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681.
- [15] 赵明. 边缘计算技术及应用综述[J]. 计算机科学, 2020, 47(S1): 268-272.
- [16] 黄永明, 郑冲, 张征明, 尤肖虎. 大规模无线通信网络移动边缘计算和缓存研究[J]. 通信学报, 2021, 42(04): 44-61.
- [17] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation

- offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [18] Liu L, Chen C, Pei Q, et al. Vehicular edge computing and networking: A survey[J]. Mobile networks and applications, 2021, 26(3): 1145-1168.
- [19] 李智勇,王琦,陈一凡,谢国琪,李仁发.车辆边缘计算环境下任务卸载研究综述[J].计算机学报,2021,44(05):963-982.
- [20] 张海波,荆昆仑,刘开健,贺晓帆.车联网中一种基于软件定义网络与移动边缘计算的卸载策略[J].电子与信息学报,2020,42(03):645-652.
- [21] Hao Y, Chen M, Hu L, et al. Energy efficient task caching and offloading for mobile edge computing[J]. IEEE Access, 2018, 6: 11365-11373.
- [22] Chen W, Wang D, Li K. Multi-user multi-task computation offloading in green mobile edge cloud computing[J]. IEEE Transactions on Services Computing, 2018, 12(5): 726-738.
- [23] You C, Huang K, Chae H, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2016, 16(3): 1397-1411.
- [24] 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, 2017, 17(3): 1784-1797.
- [25] Wang Y, Sheng M, Wang X, et al. Cooperative dynamic voltage scaling and radio resource allocation for energy-efficient multiuser mobile edge computing[C]// 2018 IEEE International Conference on Communications (ICC). IEEE, 2018: 1-6.
- [26] 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.
- [27] Darwish T S J, Bakar K A. Fog based intelligent transportation big data analytics in the internet of vehicles environment: motivations, architecture, challenges, and critical issues[J]. IEEE Access, 2018, 6: 15679-15701.
- [28] Luo G, Yuan Q, Zhou H, et al. Cooperative vehicular content distribution in edge computing assisted 5G-VANET[J]. China Communications, 2018, 15(7): 1-17.
- [29] Soua A, Tohme S. Multi-level SDN with vehicles as fog computing infrastructures: A new integrated architecture for 5G-VANETs[C]// 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). IEEE, 2018: 1-8.
- [30] Kwak J, Kim Y, Lee J, et al. DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems[J]. IEEE Journal on Selected Areas in Communications, 2015, 33(12): 2510-2523.
- [31] Shi L, Zhao L, Zheng G, et al. Incentive design for cache-enabled D2D underlaid cellular

- networks using stackelberg game[J]. IEEE Transactions on Vehicular Technology, 2018, 68(1): 765-779.
- [32] Zhang Y, Zhang H, Long K, et al. Software-defined and fog-computing-based next generation vehicular networks[J]. IEEE Communications Magazine, 2018, 56(9): 34-41.
- [33] Deng D J, Lien S Y, Lin C C, et al. Latency control in software-defined mobile-edge vehicular networking[J]. IEEE Communications Magazine, 2017, 55(8): 87-93.
- [34] 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.
- [35] 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 (GLOBECOM). IEEE, 2018: 1-7.
- [36] Zhang J, Guo H, Liu J, et al. Task offloading in vehicular edge computing networks: A load-balancing solution[J]. IEEE Transactions on Vehicular Technology, 2019, 69(2): 2092-2104.
- [37] Sun J, Gu Q, Zheng T, et al. Joint optimization of computation offloading and task scheduling in vehicular edge computing networks[J]. Ieee Access, 2020, 8: 10466-10477.
- [38] Zhang K, Mao Y, Leng S, et al. Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks[C]// 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). IEEE, 2016: 288-294.
- [39] Zhang K, Mao Y, Leng S, et al. Optimal delay constrained offloading for vehicular edge computing networks[C]// 2017 IEEE International Conference on Communications (ICC). IEEE, 2017: 1-6.
- [40] 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.
- [41] Li C, Wang S, Huang X, et al. Parked vehicular computing for energy-efficient Internet of vehicles: A contract theoretic approach[J]. IEEE Internet of Things Journal, 2018, 6(4): 6079-6088.
- [42] 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.
- [43] Wang H, Li X, Ji H, et al. Dynamic offloading scheduling scheme for MEC-enabled vehicular networks[C]// 2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops). IEEE, 2018: 206-210.
- [44] Fan W, Liu Y, Tang B, et al. Computation offloading based on cooperations of mobile edge

- computing-enabled base stations[J]. IEEE Access, 2017, 6: 22622-22633.
- [45] Abdelhamid S, Hassanein H S, Takahara G. Vehicle as a resource (VaaR)[J]. IEEE Network, 2015, 29(1): 12-17.
- [46] 李瑞轩,董新华,辜希武,周湾湾,王聪.移动云服务的数据安全与隐私保护综述[J].通信学报,2013,34(12):158-166.
- [47] Othman M, Madani S A, Khan S U. A survey of mobile cloud computing application models[J]. IEEE communications surveys & tutorials, 2013, 16(1): 393-413.
- [48] Sanaei Z, Abolfazli S, Gani A, et al. Heterogeneity in mobile cloud computing: taxonomy and open challenges[J]. IEEE Communications Surveys & Tutorials, 2013, 16(1): 369-392.
- [49] Dinh H T, Lee C, Niyato D, et al. A survey of mobile cloud computing: architecture, applications, and approaches[J]. Wireless communications and mobile computing, 2013, 13(18): 1587-1611.
- [50] 崔勇,宋健,缪葱葱,唐俊.移动云计算研究进展与趋势[J].计算机学报,2017,40(02):273-295.
- [51] Hu Y C, Patel M, Sabella D, et al. Mobile edge computing—A key technology towards 5G[J]. ETSI white paper, 2015, 11(11): 1-16.
- [52] Yi B, Wang X, Li K, et al. A comprehensive survey of network function virtualization[J]. Computer Networks, 2018, 133: 212-262.
- [53] Bari M F, Chowdhury S R, Ahmed R, et al. A survey of naming and routing in information-centric networks[J]. IEEE Communications Magazine, 2012, 50(12): 44-53.
- [54] Hu F, Hao Q, Bao K. A survey on software-defined network and openflow: From concept to implementation[J]. IEEE Communications Surveys & Tutorials, 2014, 16(4): 2181-2206.
- [55] Kemp R, Palmer N, Kielmann T, et al. eyedentify: Multimedia cyber foraging from a smartphone[C]// 2009 11th IEEE International Symposium on Multimedia. IEEE, 2009: 392-399.
- [56] Chung J M, Park Y S, Park J H, et al. Adaptive cloud offloading of augmented reality applications on smart devices for minimum energy consumption[J]. KSII Transactions on Internet and Information Systems (TIIS), 2015, 9(8): 3090-3102.
- [57] 董思岐,李海龙,屈毓铨,张钊,胡磊.移动边缘计算中的计算卸载策略研究综述[J].计算机科学,2019,46(11):32-40.
- [58] 谢人超,廉晓飞,贾庆民,黄韬,刘韵洁.移动边缘计算卸载技术综述[J].通信学报,2018,39(11):138-155.
- [59] 刘通,方璐,高洪皓.边缘计算中任务卸载研究综述[J].计算机科学,2021,48(01):11-15.
- [60] Dorigo M, Birattari M, Stutzle T. Ant colony optimization[J]. IEEE computational intelligence magazine, 2006, 1(4): 28-39.

- [61] 高阳, 陈世福, 陆鑫. 强化学习研究综述 [J]. 自动化学报, 2004, 30(1): 15.
- [62] Thrun S, Littman M L. Reinforcement learning: an introduction[J]. AI Magazine, 2000, 21(1): 103-103.
- [63] 杜威, 丁世飞. 多智能体强化学习综述[J]. 计算机科学, 2019, 46(08): 1-8.
- [64] Robbins H. Some aspects of the sequential design of experiments[J]. Bulletin of the American Mathematical Society, 1952, 58(5): 527-535.
- [65] Vermorel J, Mohri M. Multi-armed bandit algorithms and empirical evaluation[C]// European conference on machine learning. Springer, Berlin, Heidelberg, 2005: 437-448.
- [66] Kulkarni S R, Lugosi G. Finite-time lower bounds for the two-armed bandit problem[J]. IEEE Transactions on Automatic Control, 2000, 45(4): 711-714.
- [67] Min-Allah N, Wang Y, Xing J, et al. Towards dynamic voltage scaling in real-time systems-a survey[J]. International Journal of Computer Sciences and Engineering Systems, 2007, 1(2): 93-103.
- [68] Wu Y, Zou W Y. Orthogonal frequency division multiplexing: A multi-carrier modulation scheme[J]. IEEE Transactions on Consumer Electronics, 1995, 41(3): 392-399.
- [69] Vogeleer K D, Memmi G, Jouvelot P, et al. The energy/frequency convexity rule: Modeling and experimental validation on mobile devices[C]// International Conference on Parallel Processing and Applied Mathematics. Springer, Berlin, Heidelberg, 2013: 793-803.
- [70] Burd T D, Brodersen R W. Processor design for portable systems[J]. Journal of VLSI signal processing systems for signal, image and video technology, 1996, 13(2): 203-221.
- [71] Tao P D, Souad E B. Duality in DC (difference of convex functions) Optimization. Subgradient methods[J]. Trends in Mathematical Optimization, 1988, 11(3): 277-293.
- [72] Mao Y, You C, Zhang J, et al. A survey on mobile edge computing: The communication perspective[J]. IEEE communications surveys & tutorials, 2017, 19(4): 2322-2358.
- [73] Palomar D P, Chiang M. A tutorial on decomposition methods for network utility maximization[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(8): 1439-1451.
- [74] Boyd S, Boyd S P, Vandenberghe L. Convex optimization[M]. Cambridge university press, 2004.
- [75] Nesterov Y. Primal-dual subgradient methods for convex problems[J]. Mathematical programming, 2009, 120(1): 221-259.
- [76] Chen M, Hao Y, Li Y, et al. On the computation offloading at ad hoc cloudlet: architecture and service modes[J]. IEEE Communications Magazine, 2015, 53(6): 18-24.
- [77] Kenney J B. Dedicated short-range communications (DSRC) standards in the United States[J]. Proceedings of the IEEE, 2011, 99(7): 1162-1182.

- [78] Garivier A, Moulines E. On upper-confidence bound policies for switching bandit problems[C]// International Conference on Algorithmic Learning Theory. Springer, Berlin, Heidelberg, 2011: 174-188.
- [79] Bnaya Z, Puzis R, Stern R, et al. Bandit algorithms for social network queries[C]// 2013 international conference on social computing. IEEE, 2013: 148-153.
- [80] Bouneffouf D. Finite-time analysis of the multi-armed bandit problem with known trend[C]// 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2016: 2543-2549.
- [81] 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.
- [82] Chernoff H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations[J]. The Annals of Mathematical Statistics, 1952, 23(4): 493-507.
- [83] Abdulla M, Steinmetz E, Wymeersch H. Vehicle-to-vehicle communications with urban intersection path loss models[C]// 2016 IEEE Globecom Workshops (GC Wkshps). IEEE, 2016: 1-6.
- [84] Bistriz I, Baharav T Z, Leshem A, et al. One for all and all for one: Distributed learning of fair allocations with multi-player bandits[J]. IEEE Journal on Selected Areas in Information Theory, 2021, 2(2): 584-598.
- [85] Zhu Z, Liu T, Yang Y, et al. BLOT: Bandit learning-based offloading of tasks in fog-enabled networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(12): 2636-2649.
- [86] Liu C F, Bennis M, Debbah M, et al. Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing[J]. IEEE Transactions on Communications, 2019, 67(6): 4132-4150.
- [87] Tassiulas L, Ephremides A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks[C]//29th IEEE Conference on Decision and Control. IEEE, 1990: 2130-2132.
- [88] Georgiadis L, Neely M J, Tassiulas L. Resource allocation and cross-layer control in wireless networks[M]. Now Publishers Inc, 2006.
- [89] Bracciale L, Loreti P. Lyapunov drift-plus-penalty optimization for queues with finite capacity[J]. IEEE Communications Letters, 2020, 24(11): 2555-2558.
- [90] Yang Y, Fei D, Dang S. Inter-vehicle cooperation channel estimation for IEEE 802.11 p V2I communications[J]. Journal of Communications and Networks, 2017, 19(3): 227-238.

作者简历

一、基本情况

姓名：孙尉斌 性别：男 民族：汉 出生年月：1998-09-10 籍贯：湖南省长沙市

2015-09—2019-06 中国矿业大学信息与控制工程学院学士；

2019-09—2022-06 中国矿业大学信息与控制工程学院攻读硕士学位

二、学术论文

1. Li S, **Sun W**, Sun Y, et al. Energy-Efficient Task Offloading Using Dynamic Voltage Scaling in Mobile Edge Computing[J]. IEEE Transactions on Network Science and Engineering, 2020, 8(1): 588-598.

三、获奖情况

1.2019-2020 年度 一等学业奖学金

2.2020-2021 年度 二等学业奖学金

1.2021-2022 年度 一等学业奖学金

四、研究项目

1. 业务属性驱动的矿山动态协同作业场景无线通信关键技术研究，编号：62071472，项目参与人员；

学位论文原创性声明

本人郑重声明：所呈交的学位论文《车联网边缘计算资源分配与任务卸载方法研究》，是本人在导师指导下，在中国矿业大学攻读学位期间进行的研究工作所取得的成果。据我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

年 月 日

学位论文数据集

关键词*	密级*	中图分类号*	UDC	论文资助
移动边缘计算； 车联网；任务卸 载；凸优化；强 化学习	公开	TN929. 5	645	无
学位授予单位名称*	学位授予单位代码*	学位类别*	学位级别*	
中国矿业大学	10290	工学学位	硕士	
论文题名*		并列题名*	论文语种*	
车联网边缘计算资源分配与任务卸载 方法研究		Research on Resource Allocation and Task Offloading for Vehicular Edge Computing Networks	中文	
作者姓名*	孙尉斌	学号*	TS19060042A31LD	
培养单位名称*	培养单位代码*	培养单位地址	邮编	
中国矿业大学	10290	江苏省徐州市	221116	
学科专业*	研究方向*	学制*	学位授予年*	
信息与通信工程	车联网资源分配	3	2022	
论文提交日期*		2022. 4. 10		
导师姓名*	孙彦景	职称*	教授	
评阅人		答辩委员会主席*	答辩委员会成员	
盲评		刘辉	郝国生、刘卫东、荆胜羽、宋贤芳	
电子版论文提交格式 文本 (√) 图像 () 视频 () 音频 () 多媒体 () 其他 ()				
推荐格式: application/msword; application/pdf				
电子版论文出版 (发布) 者	电子版论文出版 (发布) 地		权限声明	
论文总页数*		85		
注: 共 33 项, 其中带*为必填数据, 共 22 项。				