

中文图书分类号: TP393

密 级: 公开

UDC: 004

学 校 代 码: 10005



博 士 学 位 论 文

DOCTORAL DISSERTATION

论 文 题 目: 软件定义无线网络下的计算卸载策略研究

论 文 作 者: 李飞翔

学 科: 计算机科学与技术

指 导 教 师: 刘韵洁教授

论 文 提 交 日 期: 2020 年 05 月

UDC: 004
中文图书分类号: TP393

学校代码: 10005
学号: B201641001
密级: 公开

北京工业大学工学博士学位论文

题目: 软件定义无线网络下的计算卸载策略研究

英文题目: Research on Computation Offloading Strategy
in Software Defined Wireless Networks

论文作者: 李飞翔

学科专业: 计算机科学与技术

研究方向: 移动边缘计算

申请学位: 工学博士

指导教师: 刘韵洁教授

所在单位: 北京未来网络科技创新中心

答辩日期: 2020年5月

授予学位单位: 北京工业大学

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的
研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他
人已经发表或撰写过的研究成果，也不包含为获得北京工业大学或其它教育机构
的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均
已在论文中作了明确的说明并表示了谢意。

签 名：李飞翔
日 期： 2020 年 05 月 23 日

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权
保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部
分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签 名：李飞翔
日 期： 2020 年 05 月 23 日

导师签名：张明虎
日 期： 2020 年 05 月 23 日

摘 要

近年来随着新颖移动应用场景的发展,越来越多的计算密集型和数据密集型任务对时延敏感,移动边缘计算已经成为 5G 网络应对该挑战的核心技术之一。该技术通过将计算能力从中心网络下沉至网络边缘的方式,有效缩短了传输时延。在移动边缘计算环境下,用户设备可以将计算密集型或者数据密集型的任务以超低的传输时延卸载到边缘云,由边缘云为其分配计算资源并辅助执行任务计算,从而达到加速用户设备任务处理和节约用户设备能耗的目的。然而,边缘云中的通信资源、计算资源和存储资源有限,使得如何提高用户设备卸载任务的效率成为研究热点之一。本文利用软件定义网络解耦数据平面和控制平面的思想,对整个无线网络进行全局管理,并针对软件定义无线网络计算卸载问题进行了深入的研究,主要工作如下:

(1) 在软件定义无线网络场景下,提出了面向低能耗的多控制器部署策略。软件定义无线网络中,单个控制器自身管控能力有限,单个控制器一旦发生故障,会造成整个网络的瘫痪。针对多个控制器难以合理部署问题,基于数据场理论,提出了一种自适应的多控制器部署策略。与粒子群算法等启发式算法相比,该策略可以有效减小多控制器的平均最小传输功率,降低计算时间复杂度。同时,该策略有效提高了无线网络全局管理能力,为计算卸载服务的实现提供了高效决策平台。

(2) 在软件定义无线网络场景下,提出了面向低时延的多用户计算卸载服务策略。鉴于大多数无线网络中的基站都在多信道环境中工作,因此存在多个用户设备之间如何实现有效的无线接入协调以应对计算卸载的挑战。针对计算卸载的资源受限的问题,提出了面向用户设备能耗、传输时延和计算时延多约束下的资源优化分配模型及一种离散布谷鸟搜索算法。与其他启发式算法相比,该算法可以有效缩短任务卸载的平均时延。

(3) 在软件定义无线网络场景下,提出了面向经济效益的多用户计算卸载服务策略。针对边缘云合理定价实现利润最大化的问题,基于社会信息交互的边缘云与用户设备间效用函数,提出一种斯坦科尔伯格博弈模型,并利用动态迭代算法求出最优的纳什均衡解。仿真实验表明,该方法在完全信息和不完全信息下,可使总需求值和总效用值达到近似最优。

(4) 在软件定义无线网络场景下,提出了面向智能化的多用户计算卸载服务策略。边缘服务器难以准确地对具有不同指标的任务问题建模,又无法满足边缘卸载任务的动态变化。针对边缘服务器的资源管理问题,考虑边缘云的计算、

通信和存储资源的环境状态，提出了一种基于深度强化学习的最优决策算法。仿真实验表明，该算法可以快速收敛到满意解。

综上，本文研究了软件定义无线网络场景下的计算卸载策略，从控制平面通过合理部署多控制器提高整个网络的全局管理能力，从数据平面从低时延、经济效益和智能化三个方面提供可行的计算卸载策略，为未来软件定义无线网络场景下实现经济高效智能的任务卸载服务提供了可行的解决思路。

关键词：软件定义无线网络；移动边缘计算；计算卸载；斯坦科尔伯格博弈；深度强化学习

Abstract

In recent years, with the development of new mobile application scenarios, more and more computing-intensive and data-intensive tasks are sensitive to delay. Mobile edge computing (MEC) has become one of the key technologies to cope with this challenge in the 5G networking. To be specific, this technology effectively reduces the transmission delay by transferring the computing capability from the centre to the edge. In the MEC environment, the user equipment (UE) can offload computing-intensive or data-intensive tasks to the edge cloud with ultra-low transmission delay. Furthermore, the edge cloud allocates computing resources for them and helps computing tasks, so as to accelerate task processing and save energy consumption of UE. However, the communication resources, computing resources and storage resources in the edge cloud are limited, how to improve the efficiency of resources allocation for offloading tasks has become one of the research hotspots. In this paper, we utilize the idea of decoupling data plane and control plane in software defined networking (SDN) to manage the whole wireless network globally, and make a comprehensive study on the computation offloading problem in the software defined wireless networks (SDWN). The main work is summarized as follows:

(1) In the SDWN scenario, a multi-controller deployment strategy for low energy consumption is proposed. In SDWN, it is difficult for an SDWN controller to control the whole elements due to the limitation of its ability. Meanwhile, once a single controller fails, the whole network will be paralyzed. To solve the problem that it is difficult to deploy multiple controllers reasonably, an adaptive multi-controller deployment strategy relying on the data field method is proposed. Compared with other heuristic algorithms, this strategy can effectively save the average minimal transmission power of multi-controllers and reduce the computation complexity. At the same time, the strategy effectively improves the global management ability of wireless networking, as well as provides an efficient decision-making platform for realizing computing offloading service.

(2) In the SDWN scenario, a multi-user computation offloading strategy for low latency service is proposed. Since most of the base stations in wireless networking are working in a multi-channel environment, which leads to a challenge of how to achieve effectively wireless access coordination among multiple devices for computation offloading. Hence, to solve the problem of limited communication resources, a resource allocation model is established with the energy consumption, transmission delay and computation delay of multiple UEs, and the discrete cuckoo search algorithm is utilized to allocate communication resources appropriately. Compared

with other heuristic algorithms, this algorithm can effectively reduce the average delay of task offloading.

(3) In the SDWN scenario, a multi-user computation offloading strategy for economic benefit service is proposed. Aiming at the goal of maximizing profit by reasonable pricing of edge cloud, social interaction information is introduced for establishing the utility functions of edge cloud and devices respectively. A Stackelberg game model is proposed and the Nash equilibrium solution is obtained by employing a dynamic iterative algorithm. The simulation results show that the total requirement value and total utility value can be approximately optimal in the cases of complete information and incomplete information.

(4) In the SDWN scenario, a multi-user computation offloading strategy for intelligent service is proposed. In the MEC, the edge server does not have enough computing and storage capacity. The edge cloud server can not accurately model these problems with different indexes. In addition, the edge cloud can not meet the dynamic changes of the offloaded task. Aiming at managing the resource allocation problem of edge cloud, a deep reinforcement learning algorithm is employed to make the optimal decision, considering the environmental states of computing, communication and storage resources. Simulation results show that the algorithm can converge to the satisfying solution quickly.

In conclusion, this paper investigates the offloading strategy in the SDWN scenario. The global management ability of the whole network is improved from the control plane through deploy multiple controllers reasonably. In addition, feasible computing offloading strategies are provided in the data plane in three aspects: real-time, economic benefit and intelligence. It can provide feasible solutions for economic, efficient and intelligent task offloading service in the future SDWN scenario.

Keywords: software defined wireless networks, mobile edge computing, computing offloading, Stackelberg game, deep reinforcement learning

摘 要.....	I
Abstract.....	III
第 1 章 绪论.....	1
1.1 论文的研究背景.....	1
1.2 论文的选题意义.....	2
1.3 论文的主要研究内容及创新点.....	3
1.3.1 论文的主要研究内容.....	3
1.3.2 论文的创新点.....	4
1.4 论文的组织结构.....	6
第 2 章 软件定义无线网络下的计算卸载策略相关研究概述.....	9
2.1 无线网络技术概述.....	9
2.1.1 无线网络技术的演进.....	9
2.1.2 超密集蜂窝网络技术.....	11
2.2 软件定义无线网络相关研究概述.....	13
2.2.1 软件定义无线网络的基本概念及背景.....	13
2.2.2 软件定义无线网络的研究进展.....	15
2.3 移动边缘计算研究现状.....	17
2.3.1 移动边缘计算的基本概念及背景.....	17
2.3.2 计算卸载问题研究现状.....	23
2.3.3 资源管理问题研究现状.....	27
2.4 本章小结.....	30
第 3 章 面向低能耗的多控制器部署策略.....	31
3.1 引言.....	31
3.2 网络场景和问题模型.....	32
3.2.1 软件定义无线网络架构.....	32
3.2.2 控制器部署问题模型.....	33
3.3 自适应控制器部署算法.....	34
3.3.1 控制器数量确定.....	34
3.3.2 控制器部署位置确定.....	36
3.4 仿真结果.....	36
3.4.1 影响因子的讨论.....	37
3.4.2 控制器部署结果.....	40
3.4.3 仿真结果.....	40

3.5 本章小结.....	43
第4章 面向低时延的多用户计算卸载服务策略.....	45
4.1 引言.....	45
4.2 系统模型.....	46
4.2.1 通信模型.....	46
4.2.2 计算模型.....	48
4.2.3 计算卸载问题模型.....	48
4.3 基于布谷鸟算法的计算卸载策略.....	49
4.3.1 布谷鸟搜索算法.....	49
4.3.2 基于布谷鸟搜索算法的计算卸载策略.....	52
4.3.3 布谷鸟搜索算法的收敛性分析.....	54
4.4 仿真结果.....	56
4.4.1 仿真实验参数设置.....	57
4.4.2 数值仿真结果.....	58
4.5 本章小结.....	61
第5章 面向经济效益的多用户计算卸载服务策略.....	63
5.1 引言.....	63
5.2 系统模型.....	65
5.2.1 通信模型.....	65
5.2.2 计算模型.....	66
5.2.3 斯坦科尔伯格模型.....	66
5.3 基于完全信息的斯坦科尔伯格分析.....	68
5.3.1 完全社会交互信息的问题模型.....	68
5.3.2 斯坦科尔伯格博弈纳什均衡分析.....	69
5.4 基于不完全信息的斯坦科尔伯格分析.....	72
5.4.1 不完全社会交互信息的问题模型.....	73
5.4.2 贝叶斯斯坦科尔伯格博弈纳什均衡分析.....	74
5.5 仿真结果.....	78
5.5.1 仿真实验参数设置.....	78
5.5.2 具有完全社会交互信息的斯坦科尔伯格博弈仿真结果.....	79
5.5.3 具有不完全社会交互信息的贝叶斯斯坦科尔伯格博弈仿真结果.....	82
5.5.4 与其他算法仿真结果的比较.....	85
5.6 本章小结.....	86
第6章 面向智能化的多用户计算卸载服务策略.....	87

6.1 引言.....	87
6.2 系统模型.....	88
6.2.1 超密集蜂窝网络架构.....	88
6.2.2 通信模型.....	89
6.2.3 计算模型.....	91
6.2.4 缓存模型.....	92
6.3 深度强化学习算法.....	93
6.3.1 强化学习.....	93
6.3.2 深度 Q 学习算法.....	94
6.4 基于深度强化学习算法的计算卸载和缓存策略.....	95
6.5 仿真结果.....	97
6.5.1 仿真实验参数设置.....	97
6.5.2 仿真实验结果.....	98
6.6 本章小结.....	102
第 7 章 总结与展望.....	103
参 考 文 献.....	105
附录 中英文缩略词对照表.....	115
攻读博士学位期间所发表的成果.....	117

第1章 绪论

1.1 论文的研究背景

随着无线通信技术的发展,接入无线网络的用户设备数量呈爆炸式增长。截止2020年底,全球设备总量将达到750亿台,移动通信量超过24.3EB/月^[1]。此外,用户设备(User Equipment, UE)的高度智能化为其应用提供了更为广阔的前景,如可穿戴虚拟现实^[2]、移动社交媒体^[3]和车载系统^[4]等。这些应用往往要求大量的计算能力和持久的数据处理。然而,由于受到用户设备计算能力和电池寿命的限制,这些新兴应用只能依靠先进的计算卸载策略和改进的通信基础设施来实现。因此,在移动医疗^[6]、智能电网^[7]、车联网^[8]、智能服务^[9]等应用领域,未来的通信网络不仅需要支持无线内容接入,还需要为用户设备提供计算卸载^[5]。

超密集网络(Ultra Dense Network, UDN)是本文重点研究的一种新颖的无线网络体系架构^[10]。它由小蜂窝基站(Small-cell Base Station, SBS)和宏蜂窝基站(Macro-cell Base Stations, MBS)组成。由于小蜂窝基站的密集部署,该技术为第五代移动通信技术(5th Generation Mobile Networks, 5G)网络中的用户设备提供巨大的接入容量^[11]。与此同时,为了应对实时卸载计算密集型任务的挑战,移动边缘计算应运而生^[12]。它通过部署在网络边缘的边缘云或雾节点为用户提供短延迟、高性能的计算服务,以满足延迟敏感任务的计算需求。移动边缘计算中边缘云^[13]的引入具有两大优势:(1)与本地计算^[14]相比,边缘云计算可以克服移动终端计算能力有限的限制;(2)与向远程云的计算卸载^[15]相比,边缘云计算可以避免在远程云上卸载任务导致的大延迟。因此,移动边缘计算能够很好的为延迟敏感型任务提供一定的计算能力^[16]。

通过将边缘云部署在超密集网络中,用户可以将计算强度较大的任务卸载到边缘云^[17]。然而,目前在超密集网络和移动边缘计算方面的工作大多是分开的。移动边缘计算在超密集网络中的有效应用面临以下挑战:(1)在超密集网络中,小蜂窝基站的密集部署增加了网络环境的复杂性,并且边缘云计算资源有限。因此,如何控制这些分布式计算资源是一个具有挑战性的问题;(2)当用户卸载计算任务时,他们对要访问的无线网络信息包括被访问网络的流量负载和边缘云的计算负载几乎一无所知。因此,如何根据用户设备的剩余电池容量和网络状态进行任务卸载,如何保证计算资源的合理利用以及任务的有效分配,合理的部署控制器就显得至关重要且极具挑战性。

作为一种新的网络模式,软件定义网络(Software Defined Networking, SDN)^[18]可以实现对分布式网络节点和用户设备的逻辑集中控制。软件定义网络在软件定义网络控制器的部署^[19]、网络功能的配置^[20]和软件定义网络的应用(如负载均衡^[21]和流量分配^[22])等方面得到了广泛的研究。目前利用SDN技术管理超密集网络的计算卸载工作刚刚起步,主要是在控制平面和数据平面分离的基础上,将主要的计算和控制功能从分布式的小蜂窝基站和宏蜂窝基站整合到软件定义无线网络的控制器中。软件定义无线网络的控制器通过感知网络状态,为用户设备任务卸载生成合理决策,这为移动边缘计算的实际应用提供了一个很重要的研究方向。

本文针对软件定义无线网络下的计算卸载问题,以超密集网络和移动边缘计算为基础,对多控制器部署问题、用户设备的计算卸载时延敏感问题、边缘云和用户设备资源博弈问题以及计算卸载资源智能分配问题四个关键问题进行建模,采用了数据场方法、布谷鸟搜索算法、动态迭代方法以及深度强化学习算法,对上述关键问题进行求解,实现了多控制器合理部署,提高了网络体系架构的控制能力,并从低时延、经济效益以及智能化三个方面提高了计算卸载策略在实际应用中的可行性。

1.2 论文的选题意义

移动网络的迅速发展极大地推动了社会发展,同时也对当前网络产生了严峻的挑战。网络数据的爆炸式增长延长了用户设备的传输时延,同时造成网络拥塞。因此,如何有效感知网络状态,及时有效地提供合理的计算卸载策略,是软件定义无线网络下的计算卸载策略面临的重要挑战。

首先,本课题引入了软件定义无线网络架构。软件定义无线网络控制器可以收集用户设备和边缘云的信息,从全局角度感知网络状态,从而根据数学模型计算合理决策。根据软件定义无线网络控制器的决策,用户设备选择在本地执行任务或将任务卸载到边缘云进行处理。随着网络规模扩大,单个控制器难以及时、有效的感知网络状态,因此需要部署多个控制器提高控制平面的计算能力。因此,合理部署软件定义无线网络中多个控制器,有利于提高网络架构的控制能力,及时下发合理的计算卸载策略。

其次,本课题针对该无线网络体系架构下的计算卸载场景,根据用户设备的需求对计算卸载问题进行建模。然后利用新颖的启发式算法,为卸载方案找到满意解,进而实现场景中多个用户任务快速卸载和节省终端能耗。此外,提供边缘计算服务的运营商(即边缘云)希望通过合理收费来达到自身收益最大化,同时不同用户设备也根据运营商价格的变化动态调整卸载任务量的需求。本课题将两

者的资源博弈问题建模，以求达到用户和服务运营商的斯坦科尔伯格纳什均衡。针对计算卸载资源的智能管理问题，本课题利用深度强化学习算法，考虑计算资源、通信资源和缓存资源的环境状态，针对不同状态做出最优的计算卸载决策，为实时分配任务资源提供了一种可行的方法。

1.3 论文的主要研究内容及创新点

1.3.1 论文的主要研究内容

如图 1-1 所示，本课题针对软件定义无线网络下的高效计算卸载策略进行研究，主要包括四个部分，一是面向低能耗的多控制器部署策略研究。该问题的研究旨在节省控制器部署的传输能耗，有效提高软件定义无线网络的控制能力，为高效计算卸载策略提供控制层服务；二是面向经济效益的多用户计算卸载服务策略研究。该问题的解决将有效缩短多用户任务卸载的平均时延，提高该网络场景下的计算卸载效率；三是面向经济效益的多用户计算卸载服务策略研究。该问题的解决可以为用户设备和服务运营商之间博弈达到斯坦科尔伯格纳什均衡；四是面向智能化的多用户计算卸载服务策略研究。该问题的解决有助于实现计算卸载策略的实时分配，实现网络边缘的智能计算。

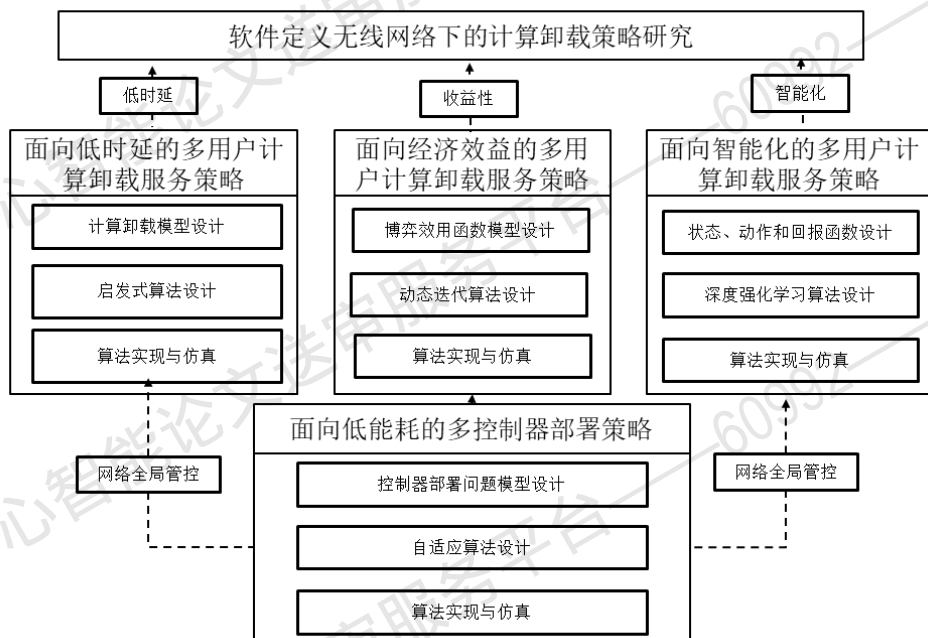


图 1-1 论文研究内容

Fig. 1-1 Research content of thesis

(1) 面向低能耗的多控制器部署策略。软件定义无线网络的网络状态主要由控制器负责，随着软件定义无线网络元素的增长，单个软件定义无线网络控制

器由于自身能力的限制，难以控制所有网元。同时，单个控制器一旦发生故障，会造成整个网络的瘫痪。主流方式是在软件定义无线网络中分布式部署多个控制器，提高大规模网络的控制能力。但在大规模网络部署中，控制器的数量和位置决定了网络的成本，因此如何有效合理地部署控制器是本课题拟研究的第一个问题。

(2) 面向低时延的多用户计算卸载服务策略。影响计算卸载性能的一个关键因素是无线接入效率。鉴于大多数无线网络中的基站都在多信道环境中工作，因此一个关键的挑战是如何在多个用户设备用户之间实现有效的无线接入协调以用于计算卸载。如果太多的用户设备选择相同的无线信道来同时将计算卸载到边缘云中，那么它们彼此间会造成严重干扰，这将降低计算卸载的数据速率。在这种情况下，用户设备将计算卸载到边缘云中是不利的。为了实现移动边缘云计算的高效计算卸载，用户设备应该如何在不同边缘云计算（通过计算卸载）之间进行选择，提高计算卸载性能是本课题拟研究的第二个问题。

(3) 面向经济效益的多用户计算卸载服务策略。在计算卸载中，除了无线接入技术的多样性，另一个趋势是网络基础设施已从一个运营商（即边缘云）的垄断转变为多个运营商并存的局面。不同运营商的竞争以及各类无线接入技术的共存，为用户设备提供了更广泛的覆盖范围、更灵活的接入模式和更好的服务质量。用户设备可以任意地选择不同运营商以及不同的无线接入方式，以获得丰富的应用服务。用户设备之间存在竞争关系，它们会更偏向于接入到服务质量更好、价格更低的接入网络。同时多个网络服务运营商为吸引顾客，相互之间也存在着竞争关系。运营商希望通过收费来支付投资成本，获得自身收益。运营商在制定价格时，必须考虑到竞争对手的价格策略，因为用户能够自主选择一个服务质量和价格等更优的运营商网络接入。运营商、用户设备彼此之间的这种交互特性是一个典型的斯坦科尔伯格博弈问题。如何达到运营商和用户设备彼此之间的斯坦科尔伯格纳什均衡是本课题拟研究的第三个问题。

(4) 面向智能化的多用户计算卸载服务策略。在计算卸载中，边缘云往往计算能力和存储能力有限，资源管理问题存在于边缘云服务器中有以下原因：第一，卸载到边缘云的任务具有不同特性、延迟要求、计算要求和能源消耗等，传统的方法不可能准确地对该问题建模。第二，边缘任务的实时性要求边缘云的能够适应性的做动态在线决策。因此，如何设计一个有效的资源管理框架，帮助边缘云实时有效地为不同任务分配资源是本课题拟研究的第四个问题。

1.3.2 论文的创新点

论文以软件定义无线网络计算卸载作为研究对象，针对多控制器合理部署问

题、计算卸载的时延敏感问题、计算卸载的经济效益问题以及计算卸载资源的智能分配问题等问题开展了研究。

论文的主要创新点如下：

(1) 针对多控制器部署问题，基于数据场理论，提出了一种自适应的多控制器部署策略。该算法自适应地对网络节点进行分簇，合理部署多个控制器。仿真实验表明，与传统启发式算法相比，该策略能够有效提升多控制器的传输能效。

(2) 针对计算卸载资源受限问题，提出了面向用户设备能耗、传输时延和计算时延多约束下的资源优化分配模型及一种离散布谷鸟搜索算法。仿真实验表明，该算法可以有效缩短任务卸载的平均时延。

(3) 针对边缘云合理定价实现利润最大化的问题，基于社会信息交互的边缘云与用户设备间效用函数，提出一种斯坦科尔伯格博弈模型，并利用动态迭代算法求出最优的纳什均衡解。仿真实验表明，该方法在完全信息和不完全信息下，可使总需求值和总效用值达到近似最优。

(4) 针对边缘服务器的资源管理问题，提出了一种基于深度强化学习的最优决策算法。考虑系统中的通信资源、计算资源和缓存资源，针对不同的环境状态做出最优的决策。仿真实验表明，该算法可以快速收敛到满意解。

如图 1-2 所示面向低能耗的多控制器部署策略提高了网络全局管控能力，为计算卸载策略实现了控制层的可行方案。面向低时延的多用户计算卸载服务策略有效解决计算卸载的时延敏感问题，面向经济效益的多用户计算卸载服务策略主要用于解决计算卸载的经济效益问题，而面向智能化的多用户计算卸载服务策略为计算卸载资源的智能分配问题提供了解决方案。

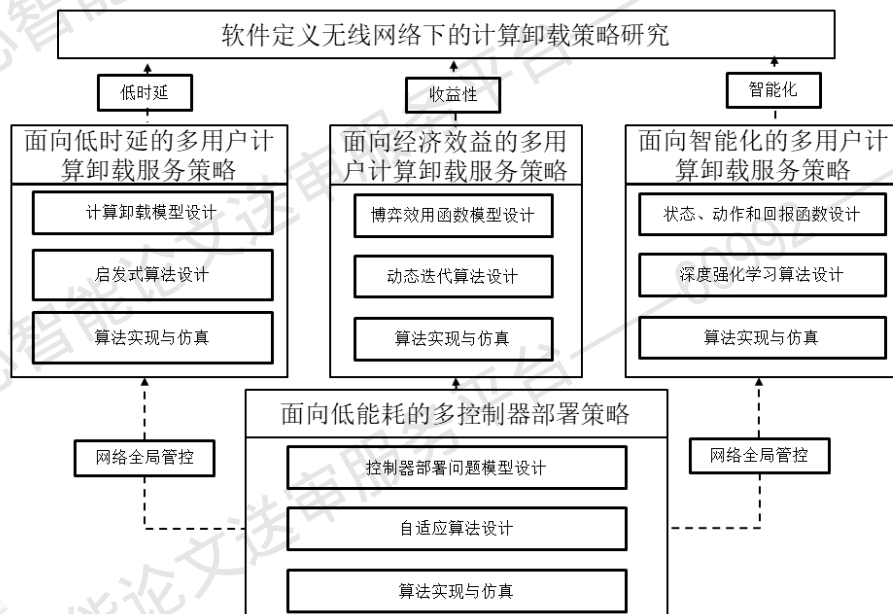


图 1-2 创新点之间的关系图

Fig. 1-2 Relationship among innovations

1.4 论文的组织结构

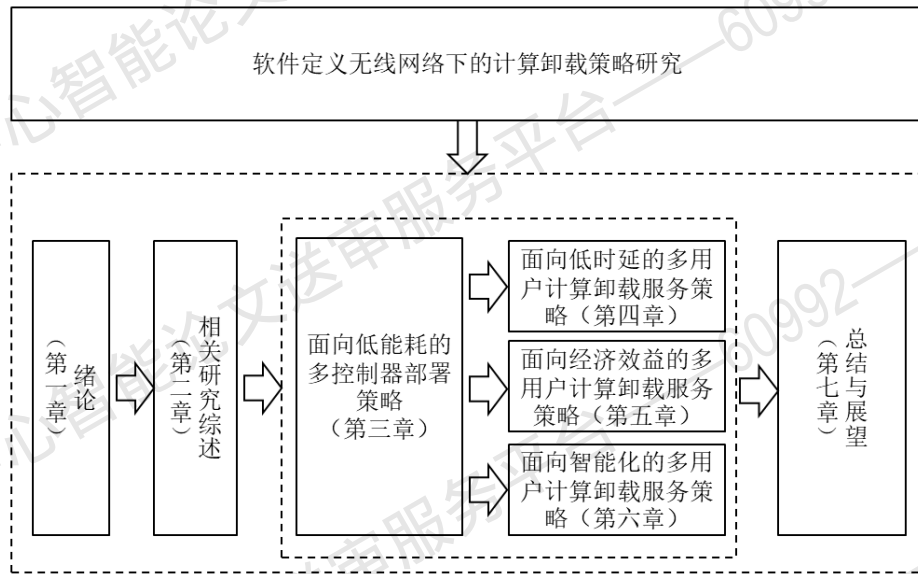


图 1-3 论文组织架构图

Fig. 1-3 Architecture of thesis

本论文基于软件定义无线网络的体系架构,针对边缘计算卸载关键问题进行研究和设计,主要内容包括软件定义无线网络的控制器部署策略、基于启发式算法的低时延计算卸载策略、基于斯坦科尔伯格的计算卸载策略,以及基于深度强化学习算法的计算卸载策略的研究。

第二章给出了当前软件定义网络下的计算卸载策略相关研究概述。包括无线网络技术概述,软件定义无线网络的基本概念、背景以及当前进展,移动边缘计算的基本概念、背景以及关键问题的研究现状。

第三章详细研究了面向低能耗的多控制器部署策略。首先介绍了当前软件定义无线网络单个控制器的瓶颈,然后给出多控制器部署的网络场景和问题模型,接下来提出了自适应控制器部署算法,最后通过仿真结果验证算法的有效性。

第四章详细研究了面向低时延的多用户计算卸载服务策略。首先描述了以低时延为目标函数的系统模型。为了求解该问题,提出基于布谷鸟算法的计算卸载策略。最后通过仿真实验验证该方法的可行性,并验证了不同参数对算法性能的影响。

第五章研究了面向经济效益的多用户计算卸载服务策略。该方案分为两个阶段,在第一阶段,边缘云确定计算服务的适当价格。然后,用户设备在第二阶段根据边缘云的价格来决定它们的计算需求。考虑到其他用户设备的社会交互信息,提出了用户设备和边缘云的效用函数模型。此外,还分析了信息完整与否的主从对策,证明了两种情形下斯坦科尔伯格均衡的存在性和唯一性。仿真结果验

证了该方法能够在计算卸载场景中提供合适的解决方案。

第六章描述了面向智能化的多用户计算卸载服务策略。该方案在超密集网络的网络架构中，所提出的算法能够根据系统的通信资源、计算资源和缓存资源智能提供有效的计算卸载策略。仿真结果验证算法可以收敛到满意解，并讨论了算法参数对实验结果的影响。

最后第七章给出了本论文全部工作的总结，并对未来的研究方向给出了展望。

第2章 软件定义无线网络下的计算卸载策略相关研究概述

2.1 无线网络技术概述

2.1.1 无线网络技术的演进

多年来,由于网络的不断变化和技术创新,不同代的无线网络(1G-4G)都在不断发展。然而,移动数据流量和持续需求呈指数级增长的问题给移动网络运营商带来了更大的压力,要求他们向最终用户提供更高的数据速率和更低的延迟服务。因此,5G网络已经于2020年开始部署。此外,在未来的6G网络计划中,卫星将与无线网络相结合,实现全球覆盖。提出的四个全球系统覆盖范围分别是美国、中国、欧盟和俄罗斯开发的全球定位系统、指南针、伽利略和全球导航卫星系统。本节描述了移动系统从1G到5G的演进过程,并在表2-1中对它们之间的比较进行了总结,以一目了然地展示1G-5G的特点^[23]。

(一) 第一代移动通信网络(1G)

1G系统部署于20世纪80年代,是模拟蜂窝系统,例如,高级移动电话服务、北欧移动电话和总接入通信系统。这些系统的数据传输速率高达2.4kbps,但也有许多缺点。此外,1G系统包括称为7号信令系统(SS7)的信令协议。

(二) 第二代移动通信网络(2G)

2G移动通信系统出现于20世纪90年代,它们是数字蜂窝系统,主要用于语音和功率控制方法,提供大约64kbps的固定数据速率,用于可靠的服务质量(Quality of Service, QoS)维护。此外,2G系统还提供电子邮件和短讯等服务。主要的2G标准是全球移动通信系统(Global System for Mobile Communications, GSM)、国际标准95(IS-95)和国际标准136(IS-136),移动和设置的显著优点是更长的电池寿命和低功率无线电信号。此外,2G有2.5G和2.75G两种传输方式。主要的2.5G技术是带有数据的通用分组无线业务,速率为50kbps,演进的GSM数据速率为200kbps,码分多址。

(三) 第三代移动通信网络(3G)

3G系统出现于2000年代,它提供高达2Mbps的传输速率,提高了服务质量。3G标准是时分多址、宽带码分多址,称为国际移动通信2000,由国际标准化电信联盟和通用移动通信系统采用3GPP标准,演进数据优化标准为3GPP2,高速上行分组接入和高速下行分组接入。第三代移动通信提供的服务包括提高语

音质量和全球漫游。此外, 3G 系统有三个过渡, 即 3.5G 和 3.75G 以及 3.9G^[24]。3G 移动系统的主要缺点是它们比大多数 2G 系统需要更多的功率。

(四) 第四代移动通信网络 (4G)

4G 无线系统旨在满足 IMT-advanced (IMT-A) 对所有服务使用 Internet 协议的要求。在 4G 系统中, 先进的数据接口技术可以使用诸如多输入多输出 (Multi Input Multi Output, MIMO)、正交频分复用和链路自适应等技术。现有的 4G 系统 (例如, LTE 和 LTE-A 系统) 能够支持在高移动性情况下高达 100Mbps 的数据速率, 在低移动性情况下高达 1Gbps。目前, 4G 网络将无法充分适应未来的预期流量, 因为网络已达到其理论极限^[25]。4G 系统的主要缺点是, 由于使用了小蜂窝专用信号, 降低了整个网络的负载, 因此会产生过多的开销。此外, 4G 还存在一些无法应对的特殊挑战, 例如巨大的能源消耗和频谱受限。因此, 新一代的移动通信 5G 成为必要。

表 2-1 不同无线通信技术比较

Table 2-1 Comparison among mobile system generations

特征	1G	2G	3G	4G	5G
部署时间	1980	1990	2000	2010	2020
过渡	-	2.5G, 2.75G	3.5G, 3.75G, 3.9G	-	-
频带	800MHz	850/900/1800/ 1900MHz	850/900/1800/ 1900/2100MHz	1.8GHz, 2.6GHz	30-300GHz
速度	2kbps	64kbps	2Mbps	1Gbps	>1Gbps
延迟	N/A	629ms	212ms	60-98ms	<1ms
功率强度 (Watt/m ²)	4.0	4.5-9.0	4.5-10	10	10
带宽	30kHz	200kHz	5MHz	1.4-20MHz	60GHz
技术	类蜂窝	数字蜂窝	CDMA, UMTS	LTE-A, WIFI	multi-RATs, Wi-Gig
应用	语音	数字语音, 低速率数据	高质量视频/音 频电话等	可穿戴设备等	增强虚拟现实等

(五) 第五代移动通信网络 (5G)

5G 无线网络将包括设备密度、天线数量空前的基站、带宽巨大的超高载波频率。5G 将确保整个世界的连接, 实现人与人、人与机器以及机器与机器之间的普遍和无缝通信。此外, 5G 网络目前有望满足新业务的不同需求, 同时支持新兴领域, 如医疗、能源、汽车、农业等。

目前, 移动产业界和学术界的注意力正朝着 5G 无线宽带系统转移, 以解决日益增长的用户需求问题。5G 的最终目标是支持 10 到 100 倍高的串行数据速率 (在密集的城市地区为 1 到 10GB/s)、10 到 100 倍高的连接设备数量、10 倍以

上的低功耗电池寿命,以及在每个区域增加 1000 倍的移动数据量。5G 技术包括 multi-RATs、超高速无线数据传输前端系统 Wi-Gig、滤波器组多载波、波束分割多址和非正交多址。总之,5G 有望实现前几代无线网络无法相比的巨大功能。

2.1.2 超密集蜂窝网络技术

每一代移动通信技术的出现都引入了新的无线技术和服务。同样,人们对移动通信网络中的高数据速率、高质量服务和可用频率资源稀缺性的兴趣日益增长,这促使人们开发新技术,以促进新兴 5G 网络频谱和能源效率的显著提高。未来的蜂窝网络需要融合大量新技术,以满足前所未有的流量需求,并提供新构想的服务,从而使 5G 系统可行。超密集蜂窝网络是 5G 网络中一种新颖的技术,也是本论文要重点研究的无线网络架构。接下来将详细介绍超密集蜂窝网络技术。

超密集蜂窝网络按蜂窝范围大小可以分为宏蜂窝和小蜂窝。宏蜂窝是传统的由运营商安装的基站,用于提供开放式接入和广域覆盖,通常是指在最大可容忍延迟和中断限制下,提供最小数据速率的宏蜂窝。宏蜂窝通常传输功率高达 46 dBm,为众多用户提供服务,并使用专用的回程链路。不同于传统的蜂窝网络,在 UDN 中通常是根据流量需求部署基站。因此,UDN 往往是一个随机网络,基站数量不固定。UDN 中的用户设备可能由于基站的随机部署而遭受来自相邻基站的蜂窝内干扰,人们对蜂窝内干扰管理进行了大量的研究,其中一种有效的方案是干扰协调,如宏蜂窝网络的频率复用技术。但是,此技术不能直接应用于 UDN 基于以下两点原因:首先,在 UDN 中,网络拓扑是不规则的,基站具有不同数量的干扰基站。因此,在 UDN 中基站在宏蜂窝网络中复用频率是困难的;其次,UDN 的宏蜂窝网络覆盖不规则,用户对服务质量的要求也不尽相同,使得 UDN 中的蜂窝负载极不均衡。显然,新的宏蜂窝网络协调方案仍在研究之中。

接下来将介绍小蜂窝的概念,小蜂窝的类型,小蜂窝的访问模式,小蜂窝中的频谱分配技术以及部署小蜂窝的优势。

小蜂窝的概念:小蜂窝也称为小型蜂窝基站,被定义为一种低成本、短距离、低功耗的蜂窝基站。该基站在有许可证和无许可证的频谱中运行,通过有线宽带连接或无线将标准移动设备连接到移动运营商的网络。小蜂窝可以安装在家庭、小型商业环境、灯柱或路灯杆上,以增强信号接收。小蜂窝部署的目的是在用户位于基站覆盖范围边界或室内等弱信号区域时提高通信质量。

小蜂窝类型:在宏蜂窝下存在不同类型的小蜂窝,即 Microcell, Picocell, Femtocell,中继节点和射频拉远头(Remote Radio Head, RRH)。这些小蜂窝通过其部署位置、覆盖范围、传输功率等进行区分。表 2-2 给出了小蜂窝之间相关

数据的详细比较。

表 2-2 小蜂窝的比较

Table 2-2 Comparison among small cells

类型	Microcell	Picocell	Femtocell	中继节点	RRH
部署位置	室外	室外/室内	室内	室外/室内	室外
覆盖范围	250 米-1 千米	<100 米-300 米	<10 米-30 米	300 米	几千米
安装者	运营商	运营商	用户	运营商	运营商
访问模式	开放访问 模式	开放访问模式	开放/封闭/混合 访问模式	开放访问 模式	开放访问 模式
传输功率	30-43 dBm	23-30 dBm	<23 dBm	30 dBm	46 dBm

小蜂窝存在三种不同的访问模式，即：封闭接入模式、开放接入模式和混合接入模式。

(1) 封闭访问模式：只有注册用户才有权在封闭访问模式下访问一个小蜂窝。这些用户被称为闭合用户组 (Closed Subscriber Group, CSG)。此访问模式为 CSG 中的小蜂窝用户提供专用服务，但如果附近有导致交叉使用的宏蜂窝用户，则性能可能会受到影响。

(2) 开放访问模式：所有附近的用户都可以在开放访问模式下访问一个小蜂窝。这种访问模式提供了最高级别的网络容量，但当附近未注册的宏蜂窝用户数量迅速增加时，就会出现由于正在运行占用带宽的应用程序而导致的 QoS 下降正在运行占用带宽的应用程序。此外，这种访问模式通常由移动运营商部署，其中具有开放访问的小蜂窝部署在公共区域。

(3) 混合接入模式：混合接入模式，即在封闭接入模式和开放接入模式之间进行权衡，为 CSG 中注册用户和小区覆盖范围内附近未注册的宏蜂窝用户提供可区分的服务。CSG 中为注册用户预留了一定数量的资源，而附近所有未注册的用户都可以使用有限数量的资源。

小蜂窝和宏蜂窝的频谱分配技术有三种：专用信道分配、共信道分配和部分信道共享分配。

(1) 专用信道分配：在专用信道分配中，小蜂窝和宏蜂窝使用不同的频带。这种分配是避免小蜂窝网络与宏蜂窝网络相互干扰的有效方法，但由于带宽分割，频谱利用率不高。因此，专用信道分配对于小蜂窝的超密集部署是不可行的，因为这既会导致 UDN 中的资源利用不足，又会导致网络流量负载发生变化时，基站可能无法充分利用其无线资源。

(2) 部分信道共享分配：在部分信道共享分配中，将总带宽分成两部分。一部分专门分配给宏蜂窝用户，另一部分由宏蜂窝和小蜂窝共享。宏蜂窝用户受益于在专用载波频率和部分覆盖（小蜂窝覆盖范围之外）或共享载波频率。这种

分配是有效的,不会造成很大的带宽损失和相互干扰,但需要一部分频谱和高成本的载波聚合终端。

(3) 共信道分配: 频谱利用率很高,因为小蜂窝和宏蜂窝共享相同的频带,而没有带宽分割,但是在这种共信道分配技术中会出现严重的干扰。无线网络运营商的专用信道分配和部分信道共享分配优于同信道分配,这是因为许可频谱成本高且稀缺,但需要强大的干扰抑制方案。

部署小蜂窝具有以下优势:

(1) 提高覆盖率: 由于发送者(小蜂窝)和接收者(小蜂窝用户)非常接近,在室外或室内环境中部署小蜂窝将覆盖宏蜂窝的覆盖漏洞。

(2) 增强容量: 部署小蜂窝将增强蜂窝网络的网络容量,因为小蜂窝以较低的功率传输,从而减少干扰并实现较高的信噪比。因此,可以将更多的用户分组到一个区域以运行同一频谱。

(3) 更高的宏蜂窝可靠性: 通过将室内/热点流量加载到小蜂窝,宏蜂窝上的流量拥塞将减少,从而使宏蜂窝更可靠。此外,从宏蜂窝到小蜂窝的流量加载提高了整体网络性能和服务质量。

(4) 延长系统电池寿命: 由于小蜂窝需要较低的传输功率,因此延长系统电池的寿命。

(5) 无缝连接: 一个小蜂窝可以执行到附近宏蜂窝的切换。因此,即使小蜂窝用户离开服务小蜂窝的覆盖区域,也有不间断的服务。

(6) 更好的 QoS: 将用户从宏蜂窝加载到小蜂窝,提高了用户的满意度,因为小蜂窝服务于少量用户,每个用户都可以获得更多的资源来满足其 QoS 要求,从而提高 QoS。

(7) 资本支出和运营支出减少: 资本支出和运营支出的很大一部分归因于场地收购、蜂窝塔安装和维护、能源账单和宏蜂窝回程。小蜂窝的部署降低了移动运营商的基础设施、维护和运营成本。

2.2 软件定义无线网络相关研究概述

2.2.1 软件定义无线网络的基本概念及背景

软件定义网络已被视为一种可能的未来网络体系架构,可在不降低性能的情况下提高网络的灵活性^[26]。软件定义网络可以调整网络配置以支持高级网络服务和应用程序。它将数据平面和控制平面解耦,其中控制平面由逻辑上集中的控制器组成。最初,软件定义网络是为管理数据中心的流量而提出的,它能够在不影

响整体性能的情况下极大地节省数据中心的运营费用^[27]。现在,软件定义网络已经成功应用到不同领域,包括云计算和网络虚拟化^[28],并且由于其具有诸多优势而受到欢迎,如灵活性强,具有网络可见性,能效高,可伸缩性强和易于管理。

随着无线网络领域中快速变化的应用程序,用户设备需求呈指数增长,同时由于管理不当将导致不同无线信号之间的干扰,从而降低用户的体验质量,因此管理无线流量是最大的挑战。无线网络正在迅速发展,迫切需要提高能源效率、无缝连接、高数据速率和更好的性能。尽管当前的网络体系结构能够支持这些应用程序,但是从这些网络获得的性能和灵活性是有限的。传统网络是特定于供应商的,由于所有网络设备形成的程序包都是由供应商捆绑在一起,因此缺乏更改网络的灵活性。此外,庞大的应用程序应用于网络会增加成本开销,因为需要升级整个网络才能支持这些应用程序。

上述原因促使形成新的无线网络体系架构——软件定义无线网络 (Software Defined Wireless Networks, SDWN)。使用软件定义无线网络的主要优势是它能够支持网络上新应用程序的部署。此功能有助于软件定义无线网络解决传统无线网络体系结构中存在的挑战,并促进网络快速创新。软件定义无线网络最初部署在蜂窝和无线局域网 (Wireless Local Area Network, WLAN) 体系结构中,后来扩展到其他网络类型,例如无线网状网络和认知网络^[29]。无线网络中的流量呈指数增长,通过这种灵活的体系结构可以有效管理网络中的大量流量,同时又不增加成本开销。软件定义无线网络中的集中式网络控制有利于在无线网络中使用流量工程技术进行有效的流量管理^[30],如在交互式网络电视 (Internet Protocol Television, IPTV), 三维电视 (3 Dimension Television, 3DTV), 高清晰度电视 (High Definition Television, HDTV) 和基于 IP 的语音传输 (Voice over Internet Protocol, VoIP) 等应用中。

(一) 软件定义网络的概述

软件定义网络来自传统网络体系结构和新研究的思想的结合^[31],这些新思想包括主动网络中所提出的网络可编程性,例如可编程交换机^[32]、网络虚拟化^[33];也包括路由控制平台模型^[34]中所提出的集中控制框架。文献[35]中提出的 4D 分布式架构将网络分为四个平面,即决策平面,传输平面,发现平面和数据平面。数据平面和决策平面分别执行转发分组和管理网络的操作。发现平面监视链路连接,而传输平面则使这些层之间的通信畅通。其中, Ethane^[36]和 OpenFlow^[37]是支持软件定义网络的著名架构。

(二) 软件定义网络架构

软件定义网络体系结构由三个不同的层组成,即数据层、控制层和应用层。软件定义网络通过解耦数据层和控制层来简化网络结构。

控制层：控制平面由管理整个网络的逻辑集中式控制器组成。控制器的主要功能包括为 OpenFlow 交换机分配流表规则、配置和更新策略。FloodLight 和 NOX 是软件定义网络广泛使用的控制器。

数据层：数据层由硬件和软件交换机组成，负责转发数据包。数据包的包头中包含要执行的动作、源地址和目的地址。OpenFlow 交换机将此操作与控制器提供的流表条目相匹配，并执行该操作。Open vSwitch、Pica8、HP 和 brocade 是数据平面交换机的一些代表示例。

应用层：应用层管理所有与业务相关的应用程序。它有一个可编程的应用程序接口，管理员可以通过它根据需要执行修改网络的操作。

（三）软件定义无线网络架构

软件定义无线网络被认为是一种新兴的技术，可以增强网络容量、覆盖范围和安全性，以支持呈指数级增长的需求和移动技术的使用。软件定义无线网络（软件定义网络和无线网络的组合）通过在网络中采用集中控制，可以简化无线网络体系结构中的许多任务。

同样地，软件定义无线网络体系结构由三层组成：数据层、控制层和应用层。软件定义无线网络的数据层由 WLAN、蜂窝网络、无线传感器网络和无线 Mesh 网络等无线网络组成。此层中的所有设备都充当 OpenFlow 交换机，并执行将数据包转发到控制器的任务。软件定义无线网络的控制层可以完成对整个网络的管理、功率和资源的分配、QoS 的优化等任务。应用层负责安全监控、视频等任务。

2.2.2 软件定义无线网络的研究进展

随着每一代通信网络的发展，蜂窝网络的技术正在迅速变化。蜂窝大小最终从微蜂窝缩小到微小蜂窝。蜂窝技术正向 5G 发展^[38]。5G 蜂窝网络的数据传输速率有望超过 1Gbps，低延迟可小于 1ms。在传统网络架构上部署 5G 会增加成本，无法提供高性能^[39]。蜂窝体系结构不断增长的需求要求实现一些经济高效的网络体系结构，如软件定义无线网络等。

软件定义无线网络能够在智能网络管理方面提供多种好处，如实现网络负载均衡和网络流量高效管理。文献[40]中提出的 SoftAir 是一种基于 SDN 的 5G 架构，其主要目的是通过选择流量工程技术来实现网络中的高效管理和提高资源利用率。尽管这些负载均衡技术提高了向最终用户提供的服务质量，但它消耗了相当大的一部分网络资源。文献[41]提出了一种基于 SDN 的 5G 蜂窝结构负载均衡方案，在不影响用户服务质量的前提下解决了高资源消耗的问题。在文献[42]中提出了一种基于软件定义无线网络的 5G 蜂窝网络终端直通（Device-to-Device, D2D）通信框架。该体系结构主要集中在当蜂窝体系结构不起作用时，在自然或

人为灾害期间启用通信。为了降低 LTE (Long Term Evolution) 蜂窝网络的成本和复杂性, 文献[43]中提出的 Cell-SDN 架构对软件定义网络体系结构做出了一些扩展, 这有助于将不灵活的传统蜂窝 LTE 网络转换成易于管理的平台, 通过软件定义无线网络支持大量的服务订阅。为了支持可编程性并提高网络的弹性和耐用性, NoStack^[44]借鉴了 SDN 体系结构的三个主要思想: a) 数据平面和控制平面的分离; b) 使用全局网络视图存储网络的当前状态; c) 将规则存储在流表中, 并将这些规则与用户流匹配。

文献[45]提出的 SWAN 架构是校园网中常用的软件定义 WLAN 架构之一。这种体系结构取代了原来昂贵和不灵活的体系结构。该体系结构还通过使用软件访问点消除了用户与访问点之间的复杂连接, 从而实现更好的连接和高度的可扩展性。VirtueMAN 是文献[46]中提出的基于 SDN 的城域 Wi-Fi 网络。它使用 SDN 来实现四个主要目标: 高服务质量、在没有虚拟专用网络 (Virtual Private Network, VPN) 的情况下启用单个安全连接、增加移动性以提供更好的连接和流量过滤。文献[47]中提出的一种大接入点架构使用 SDN 来实现 WLAN 中的合理利用和减少切换过程中的延迟。它将网络中的所有访问点抽象为一个逻辑访问点, 并使用客户端透明的访问点切换来实现高性能和负载平衡。

文献[48]提出了一种基于 SDN 的无线传感器网络 (Wireless Sensor Network, WSN) 模型, 用于解决 WSN 中的负载均衡和能量消耗等问题。在这种结构中, 网络中的主节点作为控制器, 允许应用不同的程序逻辑来有效地管理整个传感器网络。基于 SDN 的 WSN 也被用于智能电网中的监控应用。该应用遵循主节点结构, 其中所有节点都由主节点管理。主节点具有完整的网络视图, 并动态地适应环境^[49]。在文献[50]中提出了一种类似的体系结构, 其中整个网络由位于下沉节点的 SDN 控制器管理, 并且网络中的所有传感器节点都被分配了用于执行/完成操作的流表规则。这种方法有助于解决传统无线传感器网络中与网络管理相关的问题, 其网络可见性功能可以在不影响 WSN 性能的情况下实现智能网络管理。

软件定义网络中的控制器部署问题是由 Heller 等人首次提出的^[51]。该文章主要描述了 SDN 中控制器数目如何确定和控制器如何部署两个问题。作者提出了基本的评价指标, 但是没有提供一个有效的解决方案来最小化延迟。除了考虑控制器和交换机之间的传输延迟外, GAO 等人^[52]针对不同控制器之间的延迟问题, 首先采用粒子群算法求解。文献[53]考虑多目标优化, 提出了基于帕累托最优的控制器部署策略。Huque 等人^[54]提出了一种考虑交换机动态业务负载的控制器部署平台 (LiDy)。此外, Hu 等人^[55]提出了 LiDy+ 平台来降低时间复杂度。文献[56]提出了一种基于密度的控制器部署方法, 这种方法将网络分成若干个子网, 并在

每个子网中放置控制器。

针对软件定义网络中的控制器部署问题,国内学者也已经提出了一些解决方案。文献[57]给出了控制器部署问题的研究综述,主要围绕控制器部署问题的性能改善以及搜索算法两个方面,为该问题做了系统性的总结和归纳。针对控制器部署影响 SDN 中流表项安装请求响应率的问题,文献[58]提出了在传输链路存在故障的情况下,控制器部署的设计指标以平均流表项安装请求丢失数为基础,并提出了两种控制器部署算法, SROC (Flow Set-up Request quantity Oriented Controller Placement) 和 SROC-Fast。实验结果证明了该方法不仅可以优化请求的可靠性,还能有效降低平均通信延迟。Wang 等人^[59]针对软件定义无线网络的节能问题,提出了基于量子熵的禁忌搜索算法选择要切换的基站,该算法扩大搜索范围,保证了收敛速度。

2.3 移动边缘计算研究现状

2.3.1 移动边缘计算的基本概念及背景

由思科预测,在 2020 年底大约有 500 亿个物联网设备(例如,传感器和可穿戴设备)将被添加到移动互联网,这些设备将产生海量数据,对第五代移动通信^[60-62]造成挑战。此外,5G 网络^[63]新增的业务场景,如车联网^{[64][65]}、智能视频分析^[66]、增强虚拟现实^{[67][68]}、工业物联网^[69-72]等,对时延、能效等指标也提出了更高的要求。传统的移动云计算(Mobile Cloud Computing, MCC)技术^[73-75]通过无线网络中高速可靠的空口技术,为用户设备提供计算服务,但是,从用户终端到远程云中心的距离通常较长,移动应用程序的等待时间也随之增长,无法满足 5G 网络中的用户需求^[76-77]。

为了改善移动云计算的局限性,满足 5G 所需的低时延、低能耗、高可靠的业务需求,有学者提出通过让网络边缘^[78]具有计算能力,以提高终端用户的体验质量,该技术称为移动边缘计算(Mobile Edge Computing, MEC)^[79-80],并被移动通信国际标准化组织 3GPP 认可为 5G 移动网络架构的关键技术之一。与移动云计算相比,移动边缘计算具有以下两个优势:(1)有效缩短终端用户和远程云之间的数据传输距离,缩短任务执行时延;(2)避免大量数据交换造成网络拥塞,提高网络回传能力。在移动边缘计算领域中,如何合理选取终端任务卸载到边缘云执行、边缘云如何为任务分配合理的计算资源、如何将移动边缘计算合理有效地应用于现在的业务场景中都是亟待解决的关键问题。

为了有效解决以上几个关键问题,将移动边缘计算应用于实际业务场景中,

近年来研究者们将计算卸载、资源管理，以及应用场景等作为核心问题，并提出了大量基于这些问题模型的求解方法。如图 2-1 所示，本章节将系统性的概述和介绍移动边缘计算。首先介绍移动边缘的基本概念、参考框架以及它的以及应用场景。然后对移动边缘计算中计算卸载问题、资源分配问题的研究现状进行总结。

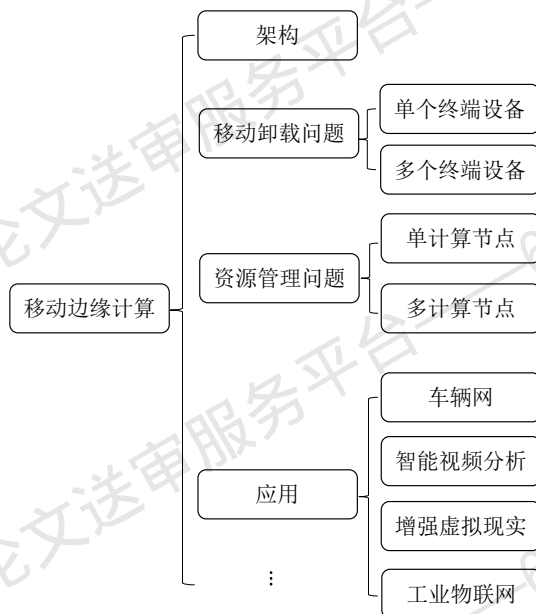


图 2-1 移动边缘计算分类图

Fig. 2-1 Classification of Mobile Edge Computing

表 2-3 三种边缘计算方案

Table 2-3 Three schemes of edge computing

方案	雾计算	微云	移动边缘计算
提出者	思科公司	卡耐基梅隆大学	欧洲电信标准化协会
是否允许边缘应用多重占用	是	是	是
是否需要云计算的扩展	是	通常是	可以是，也可以不需要
是否需要业务需求的驱动	是	较少	较少，提升体验质量
是否需要无线接入	是	不一定	是
使用灵活性	差	最差	最好

欧洲电信标准化协会在 2014 年首先提出了移动边缘计算的概念^[81]。它的定义为在靠近物或数据源头的网络边缘侧，融合网络、计算、存储、应用核心能力的分布式开放平台，就近提供边缘智能服务，满足行业数字化在敏捷联接、实时业务、数据优化、应用智能、安全与隐私保护等方面的关键需求。最近，思科公司提出雾计算的概念，将边缘设备的定义变得更广泛^[82]，这引出了一个新的研究

领域——雾计算^[83-84]。然而，雾计算和移动边缘计算的领域是重叠的，术语经常互换使用。除此之外，卡耐基梅隆大学也提出了微云^[85]的概念，这三种方案的区别和联系如表 2-3 所示。本文专注于移动边缘计算，但讨论的许多技术也同样适用于雾计算。

移动边缘计算可以认为是对移动云计算功能的补充和完善，二者相互依存，相辅相成。完全依赖移动云计算来进行数据传输和处理将会造成巨大的网络延迟和抖动，移动边缘计算将数据在边缘节点进行处理能够有效减少数据的传输和处理，但其计算能力和存储能力受限，因此通过云计算的远程存储仍然至关重要。移动云计算和移动边缘计算的区别如表 2-4 所示。

表 2-4 移动云计算和移动边缘计算的区别

Table 2-4 Difference between MCC and MEC

类型	移动云计算	移动边缘计算
部署方式	集中式	分布式
与用户设备的距离	远	近
延迟	长	短
抖动	高	低
计算能力	强大	受限制
存储能力	强大	受限制

（一）架构

图 2-2 展示移动边缘计算的架构。架构中有三个基本组件：（1）终端用户/设备，包括连接到网络上的不同应用场景的设备，应用场景有车辆网、监控视频流分析、增强虚拟现实和工业物联网等。为了节省能耗，缩短任务时延，终端用户/设备将任务卸载到边缘云或者中心云计算；（2）边缘云，部署在移动基站的计算设备。边缘云接收远程传输的任务并为其分配相应的计算资源，将运算结果返回给终端用户/设备；（3）中心云，是在互联网上的云基础设施。根据任务复杂程度不同，中心云接收边缘云无法处理的任务并为其分配相应的计算资源，将运算结果返回给终端用户/设备。

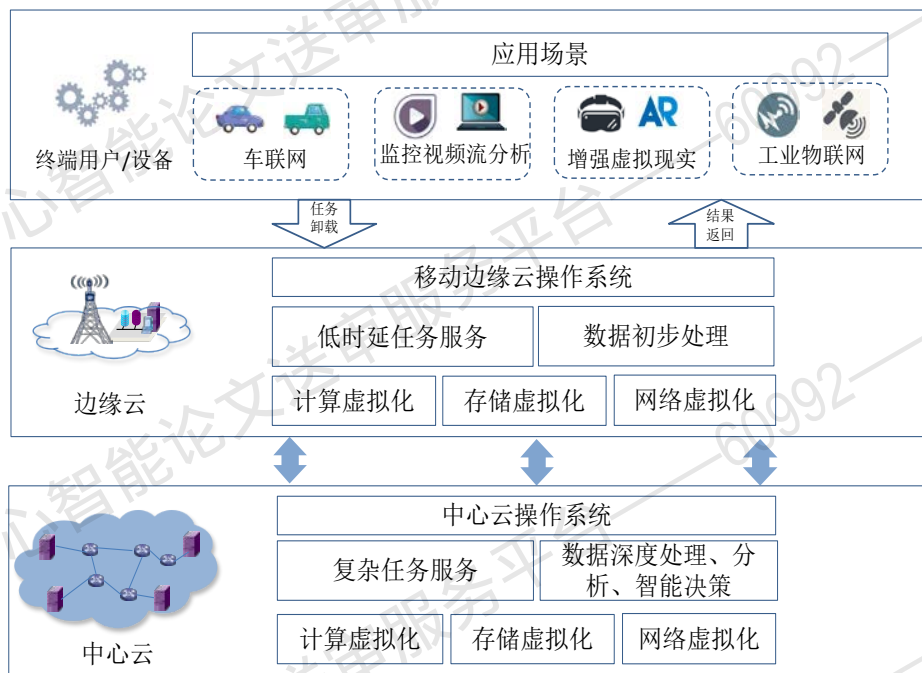


图 2-2 移动边缘计算架构图

Fig. 2-2 Architecture of mobile edge computing

移动边缘计算的主要目标是：（1）在边缘网络通过卸载优化移动资源应用，如图像处理，增强虚拟现实；（2）在发送到中心云平台之前对大数据进行优化；（3）利用无线接入网信息为用户提供内容感知服务信息。

（二）移动边缘计算主要具有以下五个优势：

（1）分布式和低延迟计算。使用可穿戴式摄像头的视觉服务，响应时间需要在 25ms 至 50ms 之间，使用云计算会造成严重的延迟；再比如工业系统检测、控制、执行的实时性高，部分场景实时性要求在 10ms 以内，如果数据分析和控制逻辑全部在云端实现，则难以满足业务要求。移动边缘计算的主要特征之一就是缩短接入核心网的时延。边缘云具备一定的计算能力，部分应用可以直接在边缘云的服务器计算，有效缩短了整个网络时延。

（2）超越用户设备的资源限制。这些用户设备以文本、音频、视频、手势或运动的形式获得数据输入，但由于中间件和硬件的限制，用户设备无法执行复杂的分析，而且执行过程也极为耗电。因此，通常需要将数据发送到云端，进行处理和运算，然后再把有意义的信息通过中继返回终端。

（3）可持续的能源消耗。一些嵌入式小型设备的基础信息采集处理完全可以在用户设备完成，即手机传感器把数据传送到网关后，通过边缘计算进行数据过滤和处理，没必要每条原始数据都传送到云，这省去了大量的能源成本。

（4）应对数据爆炸和网络流量压力。通过在边缘设备上执行数据分析，可有效应对数据爆炸，减轻网络的流量压力。边缘计算能够缩短设备的响应时间，

减少从设备到云数据中心的数据流量，以便在网络中更有效的分配资源。

(5) 智能计算。业务流程优化、运维自动化与业务创新驱动业务走向智能，边缘侧智能能够带来显著的效率提升与成本优势。

(三) 移动边缘计算的部署方案

如图 2-3 所示，根据移动边缘计算服务器的部署位置不同，移动边缘计算大致分为两种部署方案，一种方案如图 2-3 (a) 每个基站单独与边缘服务器直接连接，边缘服务器提供服务支持，并将处理后的数据或无法处理的业务通过汇聚节点发送至核心网。第二种方案如图 2-3 (b) 基站与边缘服务器无法直接连接，所有基站的数据通过汇聚节点发送至边缘服务器，边缘服务器通过处理将结果返回，并与核心网直接连接。

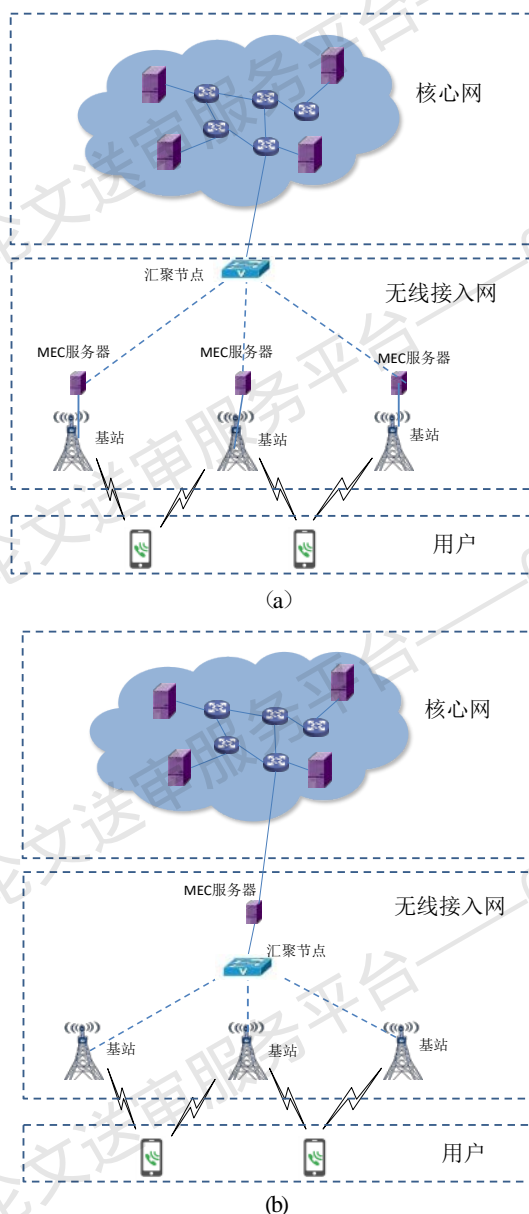


图 2-3 移动边缘计算的部署方案

Fig. 2-3 Deployment scheme of MEC

（四）移动边缘计算的应用

移动边缘计算已经在很多业务场景展现出它的优势，如表 2-4 所示，本章节将介绍四种基于移动边缘计算的典型业务场景，包括车联网、监控视频流分析、增强虚拟现实和工业物联网。

（1）车联网。车联网技术可以帮助增加车辆安全，减少交通拥堵，感知车辆行为，以及一些增值服务，如智能约车、智慧停车等^[86-88]。然而，现有的技术无法满足车联网对时延的需求。移动边缘计算可以通过路边基站连接边缘云，增加车联网的计算和地理信息服务。通过移动边缘计算服务，车辆对应用的时延可以满足终端需求。该技术已经吸引了工业界很多厂商的兴趣，如自动驾驶厂商 (Volvo, Peugeot, BOSCH 等)、通信厂商 (如 Orange, Vodafone, NTT Docomo 等)、通信制造商 (如 Qualcomm, Nokia, Huawei 等)。

目前也有学者关注该领域，文献[89]提出了基于移动边缘计算的车辆网卸载架构，在这个架构中，作者研究了车辆对基础设施 (Vehicle to Infrastructure, V2I) 和车辆对车辆 (Vehicle to Vehicle, V2V) 两种通信模式的传输效率，提出了一种预测混合模式的卸载方式，车辆根据不同的任务类型需求选择不同的传输方式，从而达到整体的纳什均衡。文献[90]提出了将信息中心网络、移动边缘计算与车辆融合，并给出了该方向可能的挑战和研究趋势。文献[91]提出了一种新颖的基于移动边缘计算技术的车辆网架构，该架构通过网络切片技术实现用户不同需求，此外车辆根据路面信息单元的信息的预先切换机制，保证车辆移动过程中信息获取的稳定性。

表 2-5 移动边缘计算的业务场景

Table 2-5 Application scene of mobile edge computing

场景	内容
车联网	可以为车联网应用提供托管服务，针对车辆快速移动做出有效的决策辅助
监控视频流分析	可以将视频分析放在本地，无线从云端或视频端下载数据信息
增强虚拟现实	减少 AR 延迟，提高数据处理精度
工业物联网	将工业内各种传感器数据汇总，分析，处理等

（2）监控视频流分析。视频流分析^[92]有非常广泛的应用，如车辆车牌识别、人脸识别和家庭安全监视等，其基本操作包括物体识别和分类。视频分析算法的计算复杂度较高，所以往往是将分析工作与视频获取设备（如摄像头等）分开，以简化设备的设计成本。如果这些视频处理任务迁移到中心云去处理^[93]，将会占用大量的网络带宽。因此将视频处理任务迁移到靠近边缘的地方，整个系统既能缩短传输时延，也能避免网络拥塞。

(3) 增强虚拟现实。增强现实技术 (Augmented Reality, AR) 是一种实时地计算摄影机影像的位置及角度并加上相应图像、视频、3D 模型的技术, 这种技术的目标是在屏幕上把虚拟世界套在现实世界并进行互动。AR 应用高度本地化, 并且要求低时延和密级的数据处理, 如博物馆虚拟参观、在线游戏等。基于移动边缘技术的 AR 系统能够精准的将请求内容进行处理和分析, 然后把 AR 结果返回终端用户, 该技术吸引了学术界和工业界强烈的兴趣^[94-96]。文献[94]利用连续凸逼近算法, 为 AR 应用提供高效的通信和计算资源分配, 实验证明该方法与分开卸载的方法相比, 可以节省整个系统更多的能耗。

(4) 工业物联网。在工业物联网中, 各式各样工业生产中的用户设备如传感器等将产生大量数据。将这些数据全部传输到远程的中心云将会造成网络拥塞, 同时影响工业物联网中不同任务的时延。为了应对这些挑战, 边缘云可以与中心云协同处理, 对实时数据进行存储、处理和分析。文献[97]提出通过边缘计算和软件定义网络两种技术, 来高效处理工业物联网中产生的数据。文献[98]针对现有物联网中实时采集的传感数据总体质量低下的问题, 提出传感数据的异常检测算法, 该算法通过基于时间序列的传感数据, 建立异常检测模型。然后针对模型实时分析数据是否产生异常, 结果表明该算法检测准确率较高且检测时间短。

2.3.2 计算卸载问题研究现状

从终端用户的角度考虑, 移动边缘计算的关键问题之一就是终端任务的计算卸载。合理的计算卸载不仅可以节省终端的计算能耗, 也能提高任务的处理速度。用户设备的任务主要有以下三种决策选择:

(1) 本地执行。由于边缘云无法为用户设备的任务提供计算资源, 或者任务过于简单, 整个任务的计算在用户的用户设备执行。

(2) 全部卸载。用户设备的任务全部卸载到边缘云的服务器处理, 然后边缘云将计算好的结果返还给用户设备。

(3) 部分卸载。这种方式是计算卸载问题常见的决策方式。用户设备的任务根据其应用类型、数据量、延迟容忍等因素, 将任务分为多个子任务, 决定哪个子任务是否卸载。任务模型可以抽象为有向无环图, 如图 2-4 所示, 主要有三种典型的依赖类型: 顺序依赖、并行依赖和全部依赖。移动应用第一步子任务收集输入/输出数据和最后一步展示计算结果通常需要再本地执行, 因此节点子任务 1 和最终节点必须本地执行。除此之外, 其他子任务直接存在依赖关系, 可以在本地执行, 也可以卸载到边缘云执行。

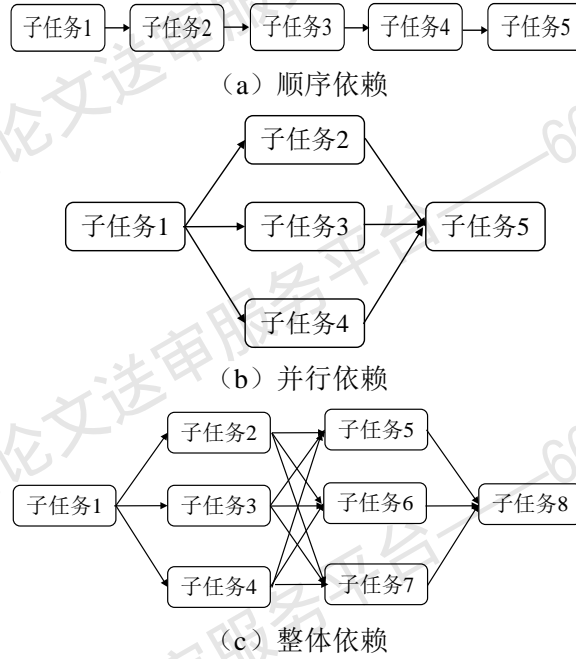


图 2-4 子任务拓扑图

Fig. 2-4 Topologies of subtasks

引入计算卸载的重要优势有两点：（1）有极大的可能性缩短执行时延；（2）节省用户设备能耗 E 。二者通常是计算卸载相关文章的目标函数，下面将详细介绍其内容。

用户设备独立执行任务的执行时延是 D_l 。任务卸载到边缘云的执行时延包括三部分：（1）卸载数据传输到边缘云的传输时延 D_t ；（2）在边缘云计算任务的计算时延 D_r ；（3）从边缘云计算完成后，将结果传输返回给用户设备的传输时延 D_t' 。如果 $D_l < D_t + D_r + D_t'$ ，用户设备在不考虑能耗的条件下，没有必要将任务卸载到边缘云执行，反之，用户设备可以选择将任务卸载到边缘云执行用于缩短时延。

在卸载任务的情况下用户设备一般考虑三种能耗：（1）用户设备将任务数据传输到边缘云的能耗 E_t ；（2）用户设备接收计算结果的能耗 E_t' ；（3）在本地执行任务需要消耗设备的能量 E_l 。如果 $E_l < E_t + E_t'$ ，用户设备将没有必要将任务卸载到边缘云执行，反之，用户设备可以将任务卸载到边缘云执行用于节省用户设备能耗。

（一）单个用户设备的计算卸载

标题文献[99]的目标就是最小化任务的执行时延。作者通过用户设备的应用缓冲排队状态、用户设备和边缘云的处理能力，以及用户设备和边缘云之间的信道特征，利用一维搜索算法选择最优的卸载决策。仿真实验将提出的算法与本地执行策略、边缘云执行策略作比较，实验结果表明提出的最优策略能够比单独在

本地执行的策略缩短 80% 的时间，比单独在边缘云执行策略缩短 44% 的时间。

文献[100]的作者提出了一种低复杂度的李雅普诺夫动态计算卸载算法。该算法在每个时隙做出卸载决策并随后为终端用户分配中央处理器的资源和传输功率。实验结果表明本算法能够比本地执行缩短 64% 的执行时间。与此同时，本算法能够避免终端应用突然中断的情况。文献[99]和[100]的弊端就是卸载决策没有考虑用户设备的能耗，而在现在移动网络中，用户设备电池的快速消耗对任务的有效执行带来巨大的障碍。

文献[101]将计算卸载问题建模为一个有限制条件的马尔科夫决策过程，作者提出了两种资源分配策略来求解该问题。第一种策略基于在线学习，根据任务的动态变化实时做出调整，第二种策略是利用一定数量的知识做线下的预先计算。数值实验表明线下预先计算策略能够优于在线策略 50% 的能耗。

在文献[102]中作者将用户设备的任务分为不能卸载的子任务和 N 个可以卸载的子任务。作者提出了一种连续优化方法，决定哪些子任务应该卸载到边缘云，然而计算复杂度却达到 $O(2^N)$ 。为了减少优化算法的复杂度，作者又提出了一种次优算法将算法复杂度减少到 $O(N)$ 。实验表明，最优算法能节省 48% 的能耗，而次优算法只是稍显不足（节省 47% 的能耗）。

文献[103]假设边缘云的计算资源稀缺的情况下，任务也可以远程卸载到中心云计算。计算卸载决策采用顺序方式，首先，用户设备决定是否将任务卸载到边缘云计算，如果任务被卸载到边缘云，接下来边缘云将根据任务量和用户的需求判断是否将任务传输到中心云。该问题是一个非凸带约束条件的二次平方式，也是 NP 难问题。因此，作者提出了一种半正定松弛和随机方法来降低整个系统的费用（全部能耗、执行时延和卸载费用），该方法比单独在用户设备计算任务要节省 70% 的费用。

文献[104]对计算卸载问题中的能耗和时延权衡做了深入的理论分析，证明信道质量越好，选择计算卸载的概率越高。实验证明天线越多，选择卸载任务的用户设备的能耗越明显。

（二）多个用户设备的计算卸载

前面的文章仅考虑单用户设备的场景，问题相对简单。文献[105]做出更进一步的拓展，将计算卸载问题从单用户设备的场景转换到多用户设备的场景。作者利用优化调度方法和计算卸载策略，保障每个用户设备的用户体验、低能耗和平均最短等待时延等目标。实验证明该策略表现优秀，但是严重依赖于其他用户设备的不同任务需求。

文献[106]也是考虑的多个用户设备的场景。作者将所有的用户设备分为两组，第一组的用户设备允许将任务卸载到边缘云，由于计算资源受限，第二组的

用户设备则不得不在本地计算。算法根据任务的排队长度（即需要处理的数据量），每隔一定的时间周期动态的分组。对于第一组的用户设备，算法为用户设备动态调整传输功率，并为它们分配计算资源。实验表明在满足时延的约束条件下，越多的天线被利用，在用户设备就需要越少的传输功率。

但是，文献[106]并未考虑到多个用户设备与基站通信时的干扰。因此，文献[107]在文献[106]的基础上考虑了多个基站的多蜂窝干扰的场景，由于建模问题是个非凸问题，作者利用连续凸逼近的方法收敛到局部最优解，并通过数值实验验证了该算法的有效性。

文献[108]考虑了多个用户设备场景下的部分卸载问题。该问题被建模为一个高复杂度的非线性约束问题，并通过线性规划的方法求解。与在本地计算相比，该方法可以节省用户设备 40% 的能耗。该方法的不足之处在于问题假设系统中的用户设备都具有相同的信道质量和相同的计算能力，显然，这些假设无法应用于实际网络中。

文献[109]考虑了 5G 微小蜂窝架构下的任务卸载问题，将任务计算和通信建成功耗模型，同时满足计算能力和服务时延需求。为了求解该问题模型，作者提出人工鱼群算法并给出了收敛性证明，实验证明了算法的有效性。

文献[110]利用动态电压调整的方法，用来优化用户设备的计算速度、传输功率，作者针对能耗和时延两种不同目标的算法，通过变量替代的方法将非凸问题转化为凸优化问题，并求解最优值。实验证明该算法可以有效地缩短时延，节省能耗。

文献[111]研究了在多个信道干扰的环境下多用户计算卸载问题，通过提出权重系数来权衡计算卸载问题的能耗和执行时延。该文章的目标主要有两个：（1）根据权重系数选择用户设备是否将任务卸载到边缘云计算；（2）一旦确定计算卸载，选择最合适的无线信道用于传输数据。作者提出了分布式计算卸载算法来达到纳什均衡，实验证明该算法远远优于所有用户设备将任务放在本地计算。

文献[112]对多个用户设备场景下的能耗和时延分析。在这种场景下，计算卸载问题需要考虑优化边缘云和用户设备之间通信和计算资源。实验证明在这个系统下，用户设备越多，所需的时延和能耗越多，但该方法依然能比只在用户设备计算节省大约 90% 的能耗。

文献[113]问题模型考虑计算卸载决策和计算资源分配。边缘云根据所有用户设备的计算需求和服务器的计算资源做出卸载决策，仿真结果说明该机制在不同系统参数下非常有效。

综上所述，如表 2-6 所示，在近几年计算卸载问题相关文章中，问题建模主要考虑时延和能耗两个目标，求解方法主要线性规划、启发式方法、博弈论、李

雅普诺夫优化方法等,场景主要考虑单个用户设备和多个用户设备。该问题的研究趋势主要是针对目标的改进、新场景的转换以及新的求解方法等。

表 2-6 计算卸载问题相关文章比较

Table 2-6 Comparison of computation offloading paper

文献	任务卸载类型	求解方法	目标	用户设备数量
[99]	全部卸载	一维搜索找到最优卸载决策	缩短时延	单个用户设备
[100]	全部卸载	基于李雅普诺夫优化的动态计算卸载	(1) 缩短时延 (2) 卸载成功率	单个用户设备
[101]	全部卸载	实时学习分配策略, 线下预先计算策略	(1) 节省能耗 (2) 满足时延约束条件	单个用户设备
[102]	部分卸载	自适应连续优化算法	(1) 节省能耗 (2) 满足时延约束条件	单个用户设备
[103]	全部卸载	启发式方法	权衡能耗和时延	单个用户设备
[104]	部分卸载	迭代方法	权衡能耗和时延	单个用户设备
[105]	全部卸载	线下随机策略	(1) 节省能耗 (2) 满足时延约束条件	多个用户设备
[106]	全部卸载	通信和计算资源分配的方法	(1) 节省能耗 (2) 满足时延约束条件	多个用户设备
[107]	全部卸载	分布式迭代算法	(1) 节省能耗 (2) 满足时延约束条件	多个用户设备
[108]	部分卸载	资源分配机制	(1) 节省能耗 (2) 满足时延约束条件	多个用户设备
[109]	部分卸载	人工鱼群算法	(1) 节省能耗 (2) 满足计算能力和时延约束条件	多个用户设备
[110]	部分卸载	变量替代的方法	(1) 节省能耗 (2) 满足时延约束条件	多个用户设备
[111]	全部卸载	计算卸载博弈的方法	权衡能耗和时延	多个用户设备
[112]	部分卸载	通信和计算资源分配的方法	权衡能耗和时延	多个用户设备
[113]	部分卸载	图着色方法	权衡能耗和时延	多个用户设备

2.3.3 资源管理问题研究现状

经过上节讨论,用户设备选择将部分任务或者全部任务卸载到边缘云计算,边缘云应该为这些任务分配合适的计算资源。根据子任务之间是否可以并行,将会有不同的资源分配方式。如果子任务不能并行,那么只有一个物理节点为这些

子任务分配资源；反之，子任务可以选择在不同计算节点分布式计算。

本小节主要调研为卸载任务分配合适的计算资源（资源管理问题）的研究现状，主要分成单个计算节点和多个计算节点两部分。

（一）单计算节点的资源管理

文献[114]考虑在任务较多的情况下满足任务的时延需求。每个任务如何放置取决于任务的优先级和边缘云的剩余计算资源。如果边缘云的计算能力无法满足任务需求，调度器将委托远程的中心云处理。为了满足每个任务的时延需求，作者提出了一种优先级合作策略，该策略为每个优先级设定了几个缓冲区阈值，如果缓冲区满，任务将会被送到中心云处理。最优的缓冲区阈值通过一种低复杂度的递归算法确定，该策略能够将任务在容忍时延内完成的概率提高 25%。

文献[115]在求解这个问题时，不仅把缩短任务的执行时延作为目标，也考虑节省边缘云的计算能耗。这篇文章考虑了基站可以接入多个边缘云服务器的场景，并提出了一个离散马尔科夫决策过程的框架来求解最优值的策略。基站根据每个服务器的计算资源状态分配任务，从而减小执行时延和能耗。但是，这种方法的弊端是边缘云服务器的数量越多，计算复杂度就会越高。

文献[116]为了提高设备和基站的能源利用率，设计了一种新颖的移动边缘计算框架 COMED。该框架的目的是最小化用户设备和基站的整体能耗，并利用李雅普诺夫优化理论对在线任务有效调度，仿真证明该算法比随机分配资源节能 30% 以上。

文献[117]在解决资源管理问题时，同时将计算卸载和内容缓存建模成一个优化问题。作者把问题转化成凸优化问题，并用分布式方法求解。仿真模拟 4 至 10 个用户设备连接到基站，证明该算法在不同系统参数下的有效性。

针对边缘云服务器的资源管理问题，文献[118]则是在求解问题方法上的创新。该文章引入博弈论的思想和强化学习方法，让边缘云通过与环境的交互，自适应的动态调整分配策略，实验证明该方法可以使服务器的利用率接近 50%。

文献[119]主要考虑时分多址（Time Division Multiple Access, TDMA）和正交频分多址（Orthogonal Frequency Division Multiple Access, OFDMA）两种多用户移动边缘计算系统的资源分配问题。在 TDMA 移动边缘计算系统中，资源分配问题建模成在时延约束条件下的最小化能耗，该问题是凸优化问题。在 OFDMA 移动边缘系统中，资源分配问题建模为混合整数规划问题。为了降低问题的复杂度，算法效果略差于最优解，但明显优于贪婪算法。

（二）多计算节点的资源管理

文献[120]将多个小区基站集群协作，根据时间周期性生成新的集群。首先小区基站尽最大努力服务与它连接的用户设备，如果无法独立处理，小区基站将

任务转发给其他基站的集群。实验仿真表明这种机制能够缩短 50% 的执行时延。

在文献[120]的基础上, 文献[121]分析了集群规模对卸载任务的执行时延的影响, 并通过设计不同的回程拓扑(星型, 树型, 自组网), 最终证明自组网拓扑结构的连接可以缩短 90% 的执行时延, 并且实验验证了随着基站个数的增加并不一定缩短执行时延, 因此合适的集群规模对系统的整体性能起到关键的作用。

文献[122]考虑时延和能耗为每个基站找到最优的集群规模。该文章提出了三种不同的集群处理方式。第一种集群策略在选择基站目标是时延最小, 该策略能够缩短 22% 的计算时延。第二种集群策略是为了节省整个基站集群的能耗, 该策略能够节省 61% 的能耗。第三种策略目的是为了节省集群和每个基站的能耗, 避免第二种策略出现的能耗不均衡的问题。

表 2-7 资源管理问题相关文章比较

Table 2-7 Comparison of resource management paper

文献	求解方法	目标	计算节点个数
[114]	一维搜索找到最优卸载决策	(1) 最大化服务任务的数量 (2) 满足时延约束	单计算节点
[115]	离散马尔科夫决策框架	(1) 缩短时延 (2) 节省能耗	单计算节点
[116]	李雅普诺夫优化理论	(1) 最小化整体能耗	单计算节点
[117]	分布式方法	利用率函数(计算资源、能耗和缓存)	单计算节点
[118]	博弈论和强化学习	服务器的利用率	单计算节点
[119]	KKT 条件	节省能耗	单计算节点
[120]	集群协作模式	缩短时延	多计算节点
[121]	无	分析不同网络拓扑对时延和能耗的影响	多计算节点
[122]	三种集群处理方式	(1) 缩短时延 (2) 节省能耗	多计算节点
[123]	为请求用户建立集群	(1) 缩短时延 (2) 节省能耗	多计算节点
[124]	对请求用户有效调度	(1) 缩短时延 (2) 节省能耗	多计算节点
[125]	考虑通信计算负载的自适应算法	权衡计算节点的通信和计算负载	多计算节点

文献[122]只是考虑了单个用户设备的场景, 而文献[123]则考虑了多个用户设备场景。该文章的核心思想是同时为每个用户设备的请求有效地分配计算集群的计算和通信资源, 从而提高用户体验。通过优化集群计算资源保证了任务的请求时延, 实验结果表明用户的满意度超过 95%。

与文献[123]类似, 文献[124]通过对用户设备的调度为多个终端用户分配集群的计算资源。算法分为两步, 第一步是本地计算资源分配, 每个小区基站为它的用户设备分配计算资源; 第二步是为每个用户设备建立计算集群。仿真结果表

明在保证所有计算节点适度能耗的情况下，用户的满意度超过 95%。

文献[125]在满足任务卸载的时延条件下，权衡了小区基站的通讯和计算负载。作者提出了一种自适应算法能够针对当前小区基站的计算和通信负载选择合适的基站，实验证明了该方法的可行性。

综上所述，如表 2-7 所示，在近几年资源管理问题相关文章中，问题建模出了考虑时延和能耗两个主要目标外，也有考虑服务器利用率、计算负载等。求解方法主要马尔科夫决策框架、李雅普诺夫优化方法等，场景主要考虑单计算节点和多计算节点。该问题的研究趋势主要是针对目标的改进、有效的求解方法等。

2.4 本章小结

本章首先对无线网络技术做了总结和概述，将无线网络从 1G 网络到 5G 网络演进做了详细叙述，并介绍了超密集蜂窝网络技术。接下来介绍了软件定义无线网络相关研究，包括软件定义无线网络的基本概念及背景和软件定义无线网络的研究进展。最后，对移动边缘计算研究现状进行概述，并详细介绍了移动边缘计算的基本概念及背景、计算卸载问题研究现状和资源管理问题研究现状。

第3章 面向低能耗的多控制器部署策略

3.1 引言

软件定义无线网络^[126]将无线网络中的控制层和基础设施层解耦，其中在控制层中运行网络控制功能的服务器，称为软件定义无线网络控制器。控制器可以看作是依靠软件定义无线网络中的服务器运行的网络操作系统。随着软件定义无线网络中的网络元素（即交换机）不断扩展，单个软件定义无线网络控制器由于其控制能力有限，很难及时有效地控制整个网络的全部交换机。因此，为了提高控制层的性能，多个软件定义无线网络控制器被部署到网络中。

在传统的软件定义网络^[127]中，如何将多个控制器合适的部署在合适的位置是一个关键问题，称为控制器部署问题（Controller Placement Problem, CPP）^[128]。这个问题包括两部分：控制器数目的确定和多个控制器的部署。近年来，考虑到网络延迟、可靠性和节能等因素，控制器部署问题在不同拓扑的有线网络中得到了很好的研究，许多启发式算法也被提出来解决这一问题，如模拟退火算法^[129]、粒子群算法^[130]等，而软件定义无线网络架构中的控制器部署问题才刚刚起步。文献[131]给出了在无线信道不确定的情况下，考虑平均响应时间和最大响应时间的控制器部署问题模型，并利用随机规划方法对其模型进行求解。为了评价其性能，在 Internet topology zoo^[132]中进行了实验验证，结果表明了该方案的有效性。文献[133]提出了一种改进的离散粒子群算法方案用于求解控制器部署问题，实验结果证明了该方案的可行性。控制器通过无线通信技术与基站/接入设备连接，能耗是软件定义无线网络多控制器部署问题应该考虑的一个关键指标。虽然能耗对无线设备的使用寿命影响较大，但这些文献中的控制器部署问题模型并没有考虑这一关键要素。

数据场方法^[135]受到物理场理论的启发，它描述了与整个空间的每个数据点相关联的对象之间的相互作用。其基本思想是用场模型描述空间数据对象之间的虚拟交互。数据对象可以根据势能自适应地分成不同的簇。本章基于数据场方法，提出了一种新的自适应控制器部署算法（Adaptive Controller Placement Algorithm, ACPA）来解决软件定义无线网络中的控制器部署问题。仿真结果表明，该方法的部署方案能有效地降低最小传输功率。本章的主要工作概括如下：

（1）重新构建了软件定义无线网络中多控制器部署问题模型。在无线网络中，传输设备的能耗主要受传输功率的影响，较少的最小传输功率有助于节省控制器传输设备的总能耗。通过建立数学模型，对多控制器的部署策略进行了评估；

(2)提出了一种有效的自适应控制器部署算法来解决多控制器的部署问题。自适应控制器部署算法包括以下两个阶段：第一，根据数据场方法，将网络中的节点划分为不同的子网络，并针对不同的拓扑确定控制器的数量。第二，针对每个子网，采用穷举法选择最优的控制器部署位置。仿真结果表明，该算法能有效地降低控制器的平均最小传输功率，有助于延长设备实际应用的时间，同时还可以有效降低时间复杂度。

3.2 网络场景和问题模型

3.2.1 软件定义无线网络架构

软件定义无线网络架构是一种将软件定义网络技术应用用于无线网络设备管理的新型架构。软件定义无线网络通过在网络中实现集中控制，可以简化无线网络体系架构中差异化任务的实现。如图 3-1 所示，软件定义无线网络架构由三个不同的层组成：应用层、控制层和基础设施层。接下来，将详细描述各层以及接口的具体功能。



图 3-1 软件定义无线网络架构

Fig. 3-1 Architecture of SDWN

1) 应用层。应用层负责管理所有业务应用程序，软件定义无线网络应用层可以将由程序请求的网络行为提交给控制器。同时，它还可以抽象和封装自己的功能，向外界提供北向代理接口。应用程序可以是视频、安全监控等。

2) 控制层。软件定义无线网络控制器主要负责两个功能，一是将应用层请求通过处理转换为基础设施层的流表指令，二是为应用层提供底层网络的抽象模型。软件定义无线网络控制器由两部分组成：软件定义无线网络控制逻辑单元和控制平面接口驱动程序。

3) 基础设施层。该层由车辆网络、蜂窝网络、无线传感器网络、WLAN 等

无线网络基础设施组成。该层中的基础设施设备具备交换机的功能，执行数据包的转发功能。

4) 接口。各层之间根据接口进行通信。软件定义无线网络架构中有两种类型的接口：北向接口和南向接口。北向接口连接应用层和控制层。最广泛使用的北向接口是 FML 和 Frantic。基础设施层和控制层之间的通信通过南向接口进行，其中 Openflow 是一种应用广泛的南向接口。

基于上述网络架构，本章考虑了该架构下的软件定义无线网络控制器的关键问题，即控制器的部署问题，通过优化部署提高控制器的效率。控制器部署问题模型将在下一小节介绍，然后通过提出的自适应控制器部署算法来解决该问题。

3.2.2 控制器部署问题模型

Friis 传输公式^[136]广泛应用于通信工程中，它将传输信号功率、天线增益、距离、波长和接收信号功率连接起来。该公式首先由 Harald T. Friis 提出，具体公式如公式（3-1）所示。

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2}. \quad (3-1)$$

下面，将详细介绍这个公式： d 表示天线之间的距离， $P_r(d)$ 表示距离 d 接收到的信号功率， P_t 表示传输的信号功率。 G_t 和 G_r 分别是传输天线和接收天线的天线增益，而 λ 是表示接收天线的有效孔径面积的波长。

根据 Friis 传输方程，传输信号功率是无线传输的一个重要因素。根据这个传输公式，可以得到最大距离与最小传输功率之间的关系。然而，这种传输方程不能直接用于能量模型的计算。因此，在本节中，假设接收信号功率是相同的，并通过算法试图找到最小传输信号功率，控制器的部署问题模型公式是求最小的平均传输信号功率，如公式（3-2a）所示。

$$\min \frac{1}{n} \sum_{i=1}^n P_t(i) = \frac{1}{n} \sum_{i=1}^n \frac{P_r(d) \times (4\pi d_{\max})^2}{G_t G_r \lambda^2}, \quad (3-2a)$$

$$\text{s.t.} \quad \sum_{s \in S, c \in C} x_{sc} = 1, \quad (3-2b)$$

$$L(c) \geq \sum_{s \in S(c)} l(s), \forall c \in C, \quad (3-2c)$$

$$D_{sc} \leq R_c. \quad (3-2d)$$

其中，公式（3-2a）表示需要满足最小平均发射功率的模型公式。 n 代表控制器的数量， $P_t(i)$ 代表控制器 i 的传输信号功率。公式（3-2b）表示每个网络交换机应该只分配给一个控制器，避免多个控制器控制一个交换机导致信息不一致的情

况。如果交换机 s 可以由控制器 c 控制, 则 x_{sc} 设置为 1。公式 (3-2c) 中, 属于它们的控制器的交换机数量不应超过控制器的处理能力。 $L(c)$ 表示控制器 c 的最大控制能力, $l(s)$ 表示由控制器 c 控制的交换机 s 。如公式 (3-2d) 所示, 每个交换机和它们的控制器之间的距离不应超过控制器的覆盖能力。 D_{sc} 表示控制器 c 与控制的交换机 s 之间的距离, R_c 代表控制器 c 的覆盖半径。在本章中, 将使用这个目标模型来对控制器部署策略进行评估。根据实验测试和参考文献[137], G_i 设为 15dB, G_r 设为 2dB, λ 设为 0.33m, P_t 设为 1W。

3.3 自适应控制器部署算法

为不同的控制器找到合适的位置是一个 NP (Non-deterministic Polynomial) 难问题。传统的启发式算法采用试错法, 计算复杂度较高。因此, 本节将提出一个自适应控制器部署算法来解决这个问题。该算法包括两个阶段: 确定控制器数量和找到最佳位置部署控制器, 这两个阶段在控制器部署中起着重要的作用。适当数量的控制器可以帮助控制器有效地与它们的交换机通信, 这可以节省控制器传输设备的能耗, 并减少控制器和交换机之间的时间延迟。此外, 将控制器部署在最佳位置能够提高该子网络中控制器的处理效率。在第一阶段, 数据场根据数据节点的位置自适应分簇, 采用数据场的方法确定每个网络拓扑的控制器数量, 这有助于减少时间复杂度。在第二阶段, 对于每个子网络, 采用穷举法找到满足最小平均发射功率的最优位置来部署控制器。

3.3.1 控制器数量确定

本小节将介绍数据场方法, 并利用这种方法来确定控制器的数量。数据场是描述整个空间中每个数据点之间相互作用的理论。数据场函数是受物理场的启发, 根据对象的物理性质由数据分布而来。它的场函数对扩散对象之间的数据强度进行数学建模。在数据域的搜索空间中, 所有的采样数据对象不仅辐射它们的数据强度, 而且还接收来自其他数据的数据强度。物体之间的拓扑关系可以用等势线来表示, 直观地反映了物体的相互作用特征。

假设一个数据对象 O 在一个给定的空间 Ω 中, 这个空间中的任意点 x 都会产生一个势值 $\varphi(x)$, 那么势值 $\varphi(x)$ 应该满足以下条件:

- (1) $\varphi(x)$ 是空间 Ω 中定义的连续、光滑、有限函数。
- (2) $\varphi(x)$ 是数据对象 O 与点 x 之间的距离的递减函数。当距离为 0 时, $\varphi(x)$ 达到最大值; 当距离趋于无穷大时, $\varphi(x) \rightarrow 0$ 。

原则上, 符合上述准则的函数可以用来定义数据场的势函数。数据场包括两

种典型的场：远程场和近程场。拟核力场函数和拟重力场函数是数据场的两种典型的势函数。拟核力场函数在远程距离上表现良好，而拟重力场函数在近程距离中具有优势^[134]。每个场函数都可以依赖数据之间的关系对数据进行聚类。通过实验验证，拟核力场函数更适合揭示控制器部署问题中数据的分簇特性，接下来将详细介绍该函数。

在数据空间 Ω 中，数据对象 x_i 产生一个具有质量 m_i 的虚拟场。如果在空间 Ω 存在一个任意点 $x = (x_1, x_2, \dots, x_n)^T$ ，那么数据场的势函数定义为公式 (3-3)，场强定义为公式 (3-4)。

$$\varphi(x) = \sum_{i=1}^n \varphi_i(x) = \sum_{i=1}^n \left(m_i \times e^{-\|x-x_i\|^k / \sigma} \right). \quad (3-3)$$

$$F(x) = \sum_{i=1}^n \left((x_i - x) \times m_i \times e^{-\|x-x_i\|^k / \sigma} \right). \quad (3-4)$$

其中 $\|x - x_i\|$ 表示数据场中数据对象 x_i 与数据点 x 之间的距离。 m_i ($m_i \geq 0, \sum_{i=1}^n m_i = 1$) 被视为数据对象 x_i 的质量，它表示来自数据场 x_i 的强度。 $\sigma \in (0, +\infty)$ 是控制数据对象之间交互距离的影响因子， $k \in N$ 表示数据对象和点之间的距离指标。

算法 3-1 自适应控制器部署算法的第一阶段

输入： 数据点 x_i ，影响因子 σ ，质量 m ；

输出： 分簇后的子网 S_j 。

初始化：

初始化自适应控制器部署算法的参数；

Begin:

While $i < n$ **do**

 根据公式 (3-3) 计算数据场每个数据点的势能 $\phi(x_i)$ ；

 根据公式 (3-4) 计算数据场每个数据点的场强 $F(x_i)$ ；

End while

 在同一个等势线的点形成子网 S_j ；

While $j < S$ **do**

If $number(S_j) = 1$ **then**

 将 S_j 与附近的子网结合，形成新的子网；

End if

End while

End

拟核力场是一种有效的对不同数据点分簇的方法，可以自适应地生成合适的簇。然而，由于该算法的分簇结果可能产生离群点，不能直接应用于控制器部署问题的求解，在这种情况下，一个离群点部署的控制器被用来控制自身，这是毫无意义的。因此，第一阶段对这种离群点的情况进行了处理，具体步骤如下：首

先,根据数据场的势能,采用数据场方法将网络划分为若干个子网络。然后,如果一个子网中的节点数只有一个,则该节点将与最近的子网相结合,形成一个新的子网,具体过程在算法 3-1 中给出。

3.3.2 控制器部署位置确定

本小节将采用穷举法来找到每个子网络中的最佳控制器部署位置。在建立新的子网络之后,用公式(3-2a)计算每个子网络中的节点,然后选择平均传输功率最小值作为控制器部署在该子网络中的位置。同时,根据约束条件判断子网络中的节点数是否满足一个控制器的覆盖容量。如果节点数满足其控制器的覆盖容量,则意味着只有一个控制器足以控制该子网络中的交换机。否则,该子网将通过调整其参数而分裂成多个子网络,所有节点数满足子网络中控制器的覆盖容量。最后,确定控制器的数量及其控制的交换机。该算法的伪码在算法 3-2 中给出。

算法 3-2 自适应控制器部署算法的第二阶段

输入: 子网 S_j ;

输出: 最佳控制器部署位置 x^* ;

Begin:

While $j < S$ **do**

 根据公式(3-2)计算每个位置节点的最小传输功率;

If $L(x_j^*) < \text{number}(S_j)$ **or** $R(x_j^*) < D_{x_j^*S}$ **then**

 重新调整参数 σ , 将子网重新分簇成多个子网直到每个子网都满足控制器的控制能力;

End if

End while

End

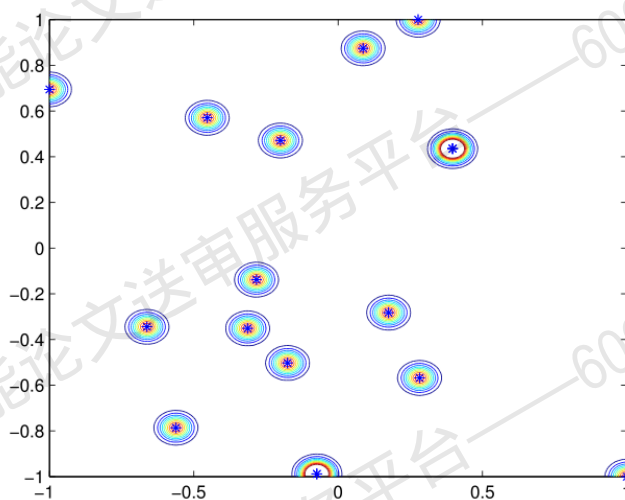
假设在 N 个节点位置中寻找 M ($M \geq 2$) 个控制器的部署位置,所需的计算次数为 C_N^M , 因此算法的时间复杂度为 $O(N^M)$ 。而自适应控制器部署算法中,假设 $N = N_1 + N_2 + \dots + N_M$, 计算次数为 $N_1 * (N_1 - 1) + N_2 * (N_2 - 1) + \dots + N_M * (N_M - 1)$, 所以该算法的时间复杂度为 $O(N^2)$ 。通过以上分析可以得出,当控制器部署个数超过 2 时,该算法可以有效减少计算复杂度。

3.4 仿真结果

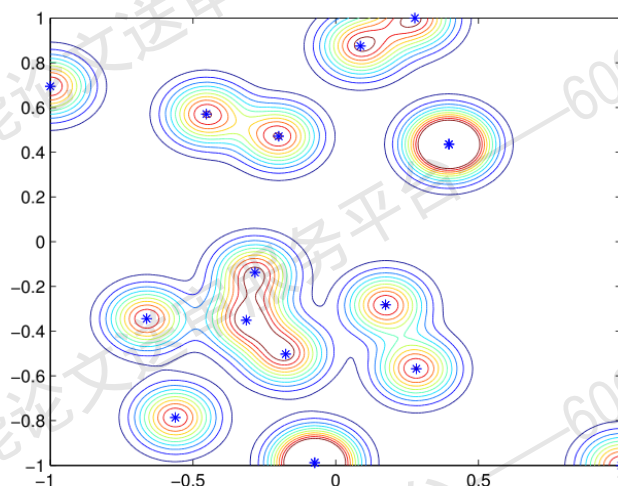
本节将通过在 Matlab 2013a 上的仿真实验来验证自适应控制器部署算法的性能。第一部分给出了实验中影响因子的讨论,第二部分给出了用自适应控制器部署算法搜索到最优位置的详细步骤,最后将自适应控制器部署算法与三种典型算法进行了比较,验证其有效性。

3.4.1 影响因子的讨论

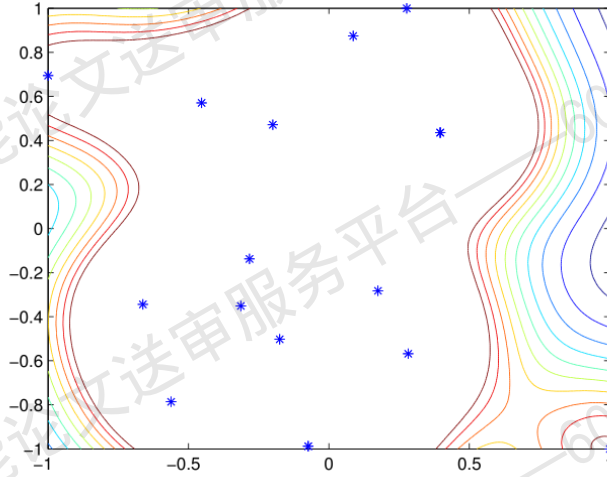
从图 3-2 可以看出, 空间分布主要取决于数据对象之间的交互作用。不同的影响因子对数据场的空间分布有很大的影响。当 σ 值很小时, 数据对象之间的交互范围很短, 在极端情况下, 每个数据对象之间没有交互。当 σ 值很大时, 数据对象之间的相互作用非常强, 每个对象在极端条件下的势值大约等于 1。考虑到势函数可以反映数据的强度分布, 势场分布的极端情况不能明显地产生显著的整体估计。因此, 有必要通过寻找适当的 σ 值来获得最佳的势场分布。



(a) $\sigma = 0.03$



(b) $\sigma = 0.13$

(c) $\sigma = 0.8$ 图 3-2 不同 σ 值下的势场分布Fig. 3-2 Potential distribution of different σ

根据信息论相关理论^[137]，熵是系统的不确定性度量。熵越大，系统的不确定性越大。对于数据空间中的潜在分布，如果每个数据对象近似等于势值的位置，则数据不确定性的分布具有最大熵。如果目标位置势值是离散的或不对称的，则不确定性是最低的，并且它具有最小熵。因此，最小势值分布熵的对应 σ 值是最优的。

假设在空间 Ω 中有 n 个数据对象 $x = \{x_1, x_2, x_3, \dots, x_n\}$ ，每个数据对象的势值分别为 $\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n$ ，那么势熵的构建公式为式 (3-5)

$$H = -\sum_{i=1}^n \frac{\varphi_i}{Z} \log\left(\frac{\varphi_i}{Z}\right). \quad (3-5)$$

其中 $Z = \sum_{i=1}^n \varphi_i$ 是标准化因子， $0 \leq H \leq \log(n)$ ， $H = \log(n) \Leftrightarrow \varphi_1 = \varphi_2 = \varphi_3 = \dots = \varphi_n$ 。

为了寻找影响因子 σ 的最优值，引入了黄金分割法，这是一种理想的优化方法。该方法的思想是把试验点放在黄金分割点时作出每一个选择。假设初始区间为 $[a, b]$ ，目标函数为 $\min f(x)$ ，黄金分割法的步骤如下：

(1) 设置初始空间的最小值和最大值为 $a_1 = a$ ， $b_1 = b$ 。精确度为 $\varepsilon > 0$ ，初始值 $k = 1$ ，那么接下来计算 λ_1 ， μ_1 ， $f(\lambda_1)$ 和 $f(\mu_1)$ ，

$$\lambda_1 = a_1 + 0.382(b_1 - a_1). \quad (3-6)$$

$$\mu_1 = a_1 + 0.618(b_1 - a_1). \quad (3-7)$$

(2) 如果 $(b_k - a_k) < \varepsilon$ ，停止迭代。否则，当 $f(\lambda_k) \geq f(\mu_k)$ ，转到步骤 (3)。如果 $f(\lambda_k) < f(\mu_k)$ ，转到步骤 (4)。

(3) 设置 $a_{k+1} = \lambda_k$ ， $b_{k+1} = b_k$ ， $\lambda_{k+1} = \mu_k$ ， $\mu_{k+1} = a_{k+1} + 0.618(b_{k+1} - a_{k+1})$ ，计

算 $f(\mu_{k+1})$ ，转到步骤 (5)。

(4) 设置 $a_{k+1} = a_k$ ， $b_{k+1} = \mu_k$ ， $\mu_{k+1} = \lambda_k$ ， $\lambda_{k+1} = a_{k+1} + 0.382(b_{k+1} - a_{k+1})$ ，计算 $f(\lambda_{k+1})$ ，转到步骤 (5)。

(5) 设置 $k = k + 1$ ，返回步骤 (2)。

图 3-3 中示出了黄金分割法在上述网络拓扑中搜索的具体过程。整个搜索过程分为 8 轮，影响因子 σ 的初始空间为 $[0,1]$ 。精度要求是 $\varepsilon = 0.1$ 。在每一轮中，将势熵计算四次，并将最佳区间选择为接近最优值。通过黄金分割搜索，最佳影响因子 σ 为 0.146。

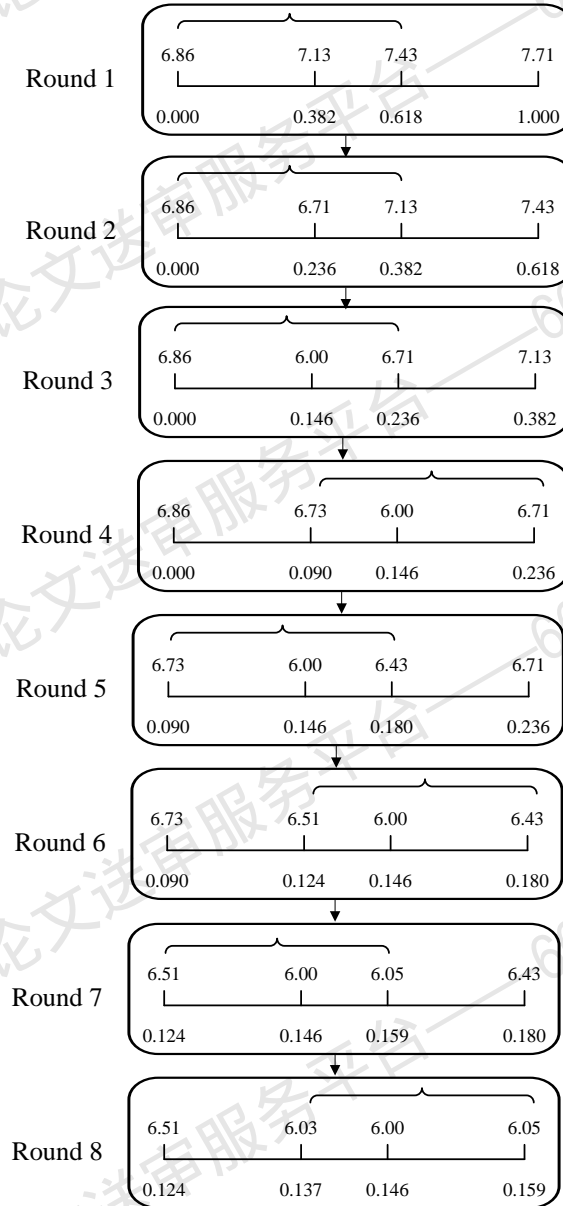


图 3-3 黄金分割法的具体过程

Fig. 3-3 Concrete procedure of the golden section method

3.4.2 控制器部署结果

在本小节中, 对自适应控制器部署算法进行了两种典型网络拓扑的测试: China 网络拓扑和 USA 网络拓扑。两个网络拓扑的原始位置在图 3-4 (a) 和图 3-5 (a) 中示出。从图 3-4 (b) 可以看出, 根据自适应控制器部署算法, 等势线是由节点之间的相互作用产生的, 原始网络被分成七个子网络。然后, 将离群点与其最近的子网相结合, 构成一个新的子网。最后的子网数目是 4, 控制器的部署位置在图 3-4 (c) 中被标记。类似地, 图 3-5 (b) 展示了该算法在另一个拓扑上寻找控制器的部署位置的细节, 图 3-5 (c) 展示出了最终子网络及其控制器的位置。

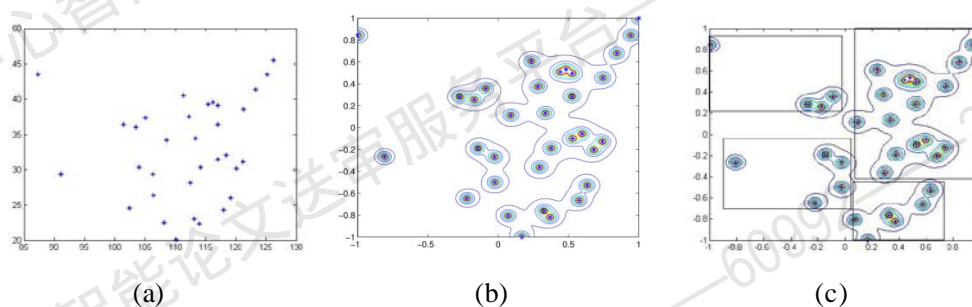


图 3-4 China 网络拓扑的控制器部署结果

Fig. 3-4 Controller placement result of China topology

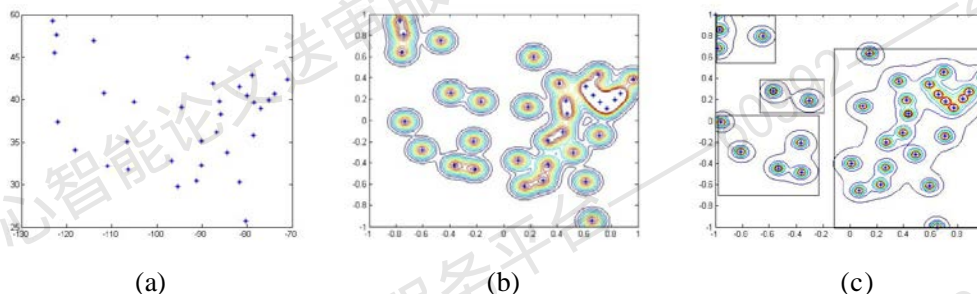


图 3-5 USA 网络拓扑的控制器部署结果

Fig. 3-5 Controller placement result of USA topology

3.4.3 仿真结果

本小节评估了自适应控制器部署算法的实验性能。测试网络拓扑是从 Internet topology zoo^[132]中随机挑选了七种不同数量节点的网络拓扑。每个网络拓扑的控制器数目由自适应控制器部署算法生成。表 3-1 表示每个网络拓扑的具体节点数量 and 控制器数目。每个网络拓扑节点的位置不同, 实验仿真效果与节点的位置有着显著的关系。如果节点分布均匀, 很难适当地分簇, 这就增加了时间复杂度。为了验证提出算法的性能, 将该算法与粒子群算法 (Particle Swarm Algorithm,

PSO)^[138]、遗传算法 (Genetic Algorithm, GA)^[139]和贪婪算法 (Greedy Algorithm, GL)^[140]进行了比较。这三种算法提前确定控制器的个数,并通过实验确定了自适应控制器部署算法的影响因子 σ 。三种算法的粒子数为 20,每个算法运行 50 代。对于粒子群算法的参数, c_1 和 c_2 分别代表了社会影响和个体影响, w 是先前速度的权重系数。对于遗传算法的参数, P_c 代表了模拟染色体遗传模式的交叉概率,而 P_m 是模拟基因突变的突变概率。每种算法的具体参数参见表 3-2 中。

表 3-1 不同网络拓扑的控制器数量

Table 3-1 Controller numbers of different topologies

网络拓扑	节点数量	控制器数量
China	33	4
USA	34	4
Aarnet	18	2
Arn	29	3
Bandcon	21	3
Bellcanada	47	3
Esnet	17	2

表 3-2 不同算法的参数设置

Table 3-2 Parameters of different algorithms

算法	参数
粒子群算法	$c_1 = 0.1, c_2 = 2.0, w(0.9 \rightarrow 0.4)$
遗传算法	$P_c = 0.8, P_m = 0.1$
自适应控制器部署算法	$m_i = 1, k = 2$

图 3-6 描绘了关于七种不同网络拓扑下最小传输功率的仿真结果。如图所示,在这些网络拓扑中,自适应控制器部署算法具有相同的控制器数目条件下的平均最小传输功率,这意味着自适应控制器部署算法可以为控制器提供更好的位置部署方案来节省它们的传输功率。相对而言,粒子群算法的性能优于其他两个,但略低于自适应控制器部署算法。遗传算法能够达到不错的搜索精度,其性能优于贪婪算法。特别是在 China 和 Bellcanada,遗传算法能够达到接近粒子群算法的结果。贪婪算法在这些算法中表现最差,因为它容易陷入局部最优。此外,不同算法所获得的实验结果也受节点数目和控制器数目的影响,较少的节点数目和更多的控制器数目需要更少的最小传输功率。通过实验证明,自适应控制器部署算法在软件定义无线网络中具有求解控制器部署问题的能力,且性能良好。

节省平均传输功率可能会影响其他性能指标。为了验证这一点,本节添加了另一个实验来证明自适应控制器部署算法对平均计算时延的影响。从图 3-7,与七种网络拓扑中的其他算法相比,自适应控制器部署算法能够达到较低的平均计算时延的效果。特别是在网络拓扑 **Am** 中,自适应控制器部署算法效果比贪婪算法略好,明显优于粒子群算法和遗传算法。此外,遗传算法比自适应控制器部署算法更差,但优于其他两种算法。此外,贪婪算法的性能受不同网络拓扑结构的影响比较大。最后,可以得出结论,自适应控制器部署算法可以很好地平衡最小传输功率和计算时延。

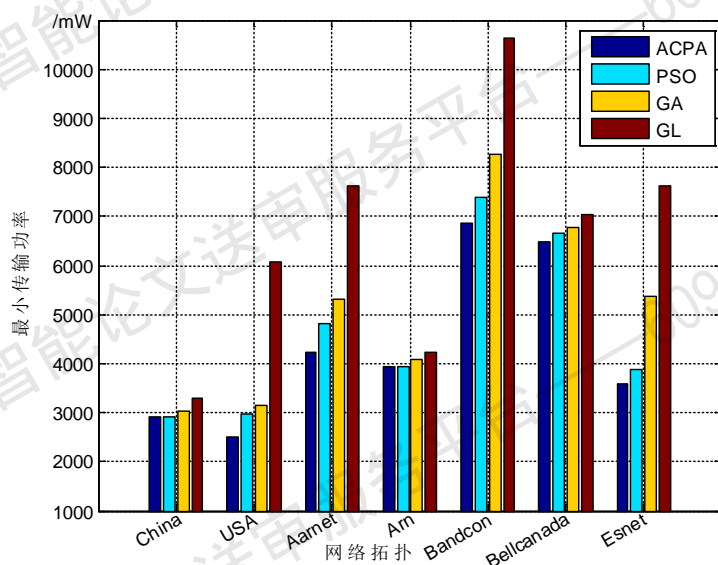


图 3-6 不同网络拓扑下的平均最小传输功率

Fig. 3-6 Minimal average transmitted power results for different topologies

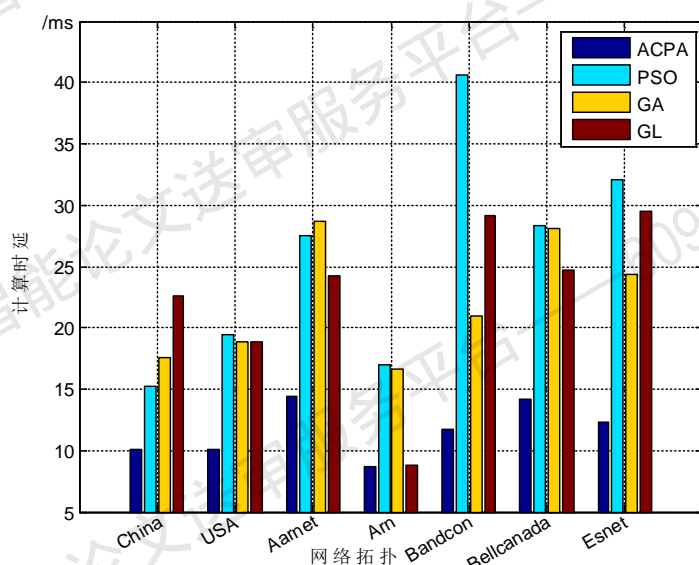


图 3-7 不同网络拓扑下的计算时间

Fig. 3-7 Average latency results for different topologies

3.5 本章小结

本章研究了软件定义无线网络中的控制器部署问题。为了延长控制器在实际软件定义无线网络应用的寿命,综合考虑最小传输功率的问题重新建立模型。为了解决控制器部署问题,有效解决目标问题,本章提出了一种自适应的控制器部署算法,该方法分为两个阶段:控制器数量确定和控制器部署位置确定。仿真实验表明该算法能够自适应地生成不同网络拓扑结构的控制器数目,有效地降低了算法的复杂度。此外,与其他对比算法相比,该算法能更有效地降低能耗。

第4章 面向低时延的多用户计算卸载服务策略

4.1 引言

近年来,随着物联网(Internet of Things, IoT)的快速发展^[141],用户设备产生的海量数据给5G网络带来了挑战。此外,新增加的5G网络业务场景如工业物联网(Industrial Internet of Things, IIoT)^[142],对延迟、能源效率和其他指标提出了更高的要求。为了应对以上挑战,超密集蜂窝网络通过将密集部署小蜂窝基站,使得5G网络能够为IIoT用户设备提供强大的访问能力。

软件定义网络是一种新型的网络技术,它将控制平面和基础设施平面分离开来。因此,软件定义网络被认为是实现快速配置和有效管理的一种高效的网络体系架构,为实现灵活的网络控制和管理提供了潜力,这种新的网络架构被称为基于软件定义网络的工业物联网体系架构。为了满足基于软件定义网络的工业物联网体系架构对5G业务^[143-144]的要求,即低延迟、低能耗、高可靠性,移动边缘计算技术应运而生。该技术将计算能力从网络中心下沉到边缘,为5G网络中的用户设备的任务提供高效的计算卸载服务。

本章研究的计算卸载问题基于软件定义网络的工业物联网体系架构中。在此架构中,控制平面和基础设施平面进行分离,即SDWN控制器与分布式小蜂窝基站和宏蜂窝基站分离。SDWN控制器可以从用户设备和蜂窝基站收集信息,并从全局的角度感知网络状态,为计算卸载任务作出合理决策。根据SDWN控制器的决策指令,用户设备选择将任务卸载到小蜂窝基站边缘云或宏蜂窝基站边缘云中。本章将计算卸载问题建模为一个混合整数非线性问题,该问题的主要目标是在满足能耗约束条件下最大限度地减少全部用户设备的延迟。作为一种启发式优化算法,布谷鸟搜索算法可以在降低计算复杂度的前提下获得NP难问题的近似最优解。因此,基于布谷鸟搜索算法^[145-147]良好的收敛能力,本章提出了一种基于布谷鸟搜索算法的计算卸载策略。本章主要做了以下工作:

(1) 构建了基于软件定义网络的工业物联网体系架构。在SDWN控制器的帮助下,用户设备、小蜂窝基站和宏蜂窝基站的全局信息可以被获取并分析。因此,该框架有助于为计算卸载策略提供最佳决策。

(2) 设计了一个该网络体系架构下的计算卸载问题模型,考虑到在能耗约束条件下最小化全部用户设备的延迟。此外,本章提出一种离散布谷鸟算法用于有效解决计算卸载问题模型。

(3) 通过大量的实验来评估面向低时延的多用户计算卸载服务策略的性能。

实验结果验证了所提策略能够在能量消耗约束下能有效地降低用户设备的时延，此外，还分析了算法参数和任务参数对实验结果的影响。

4.2 系统模型

4.2.1 通信模型

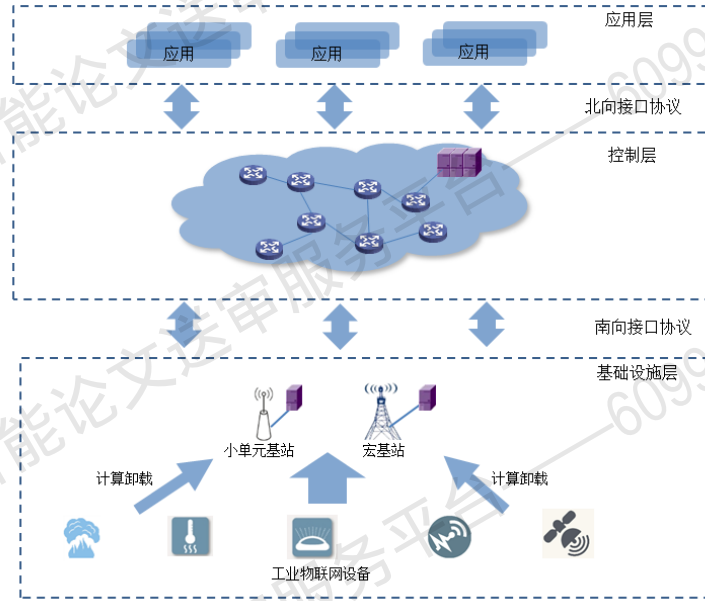


图 4-1 基于软件定义网络的工业物联网体系架构

Fig. 4-1 Architecture of SDN-based IIoT

在本节中，图 4-1 描述了基于软件定义网络的工业物联网体系架构，该体系架构考虑一个宏蜂窝基站和多个小蜂窝基站随机分布在网络中的情况。在此情况下， $N = \{1, 2, \dots, N\}$ 表示 N 个小蜂窝基站的集合， $M = \{1, 2, \dots, M\}$ 表示 M 个工业物联网设备的集合。此外，用户设备的计算卸载模式如图 4-2 所示，每个小蜂窝基站通过无线传输方式与宏蜂窝基站。在每个工业物联网设备中的任务可以被卸载到宏蜂窝基站或者小蜂窝基站，同时，每个设备最多连接一种基站卸载任务。假设在设备 m 中的任务 J_m 表示为 $J_m = \{A_m, B_m, T_m^{\max}\}$ ，其中 A_m 表示任务 J_m 传输到宏蜂窝基站或者小蜂窝基站的传输量， B_m 表示任务 J_m 所需要的中央处理器（Central Processing Unit, CPU）全部数量， T_m^{\max} 表示任务 J_m 传输到宏蜂窝基站或者小蜂窝基站的时延容忍。根据不同任务时延容忍的不同需求，任务可以选择不同的模式卸载。接下来，本小节用 $\mathbb{X} = \{x_m\}, m = 1, 2, \dots, M$ 表示计算卸载模式的参数，其中 $x_m = 0$ 代表设备 m 选择将任务卸载到宏蜂窝基站处理，否则， $x_m = n, (n \in N)$ 表示设备 m 选择将任务卸载到小蜂窝基站 n 处理。基于此，本文定

义了一个二进制函数 $F(x_m, i)$, ($i=0,1,2,\dots,N$), 该函数的公式如式 (4-1) 所示。

$$F(x_m, i) = \begin{cases} 1, & \text{if } x_m = i, \\ 0, & \text{otherwise.} \end{cases} \quad (4-1)$$

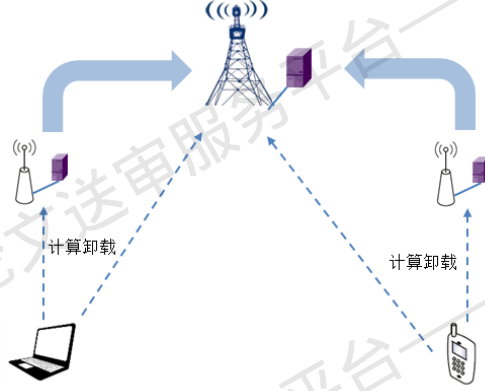


图 4-2 计算卸载模式

Fig. 4-2 Computation offloading pattern

(1) 任务卸载到宏蜂窝基站边缘云的情况。在这个情况下, 本文用 w_{m0} 表示通信模型中设备 m , ($m \in \mathbb{M}$) 和宏蜂窝基站边缘云之间的信道带宽。除此之外, p_{m0} 用来表示设备 m , ($m \in \mathbb{M}$) 传输到宏蜂窝基站边缘云的传输功率, σ^2 表示高斯白噪声常量, h_{m0} 表示设备 m , ($m \in \mathbb{M}$) 和宏蜂窝基站边缘云之间的信道增益, $k_{m0} \in \{0,1\}$ 表示设备 m , ($m \in \mathbb{M}$) 和宏蜂窝基站边缘云的干扰参数。因此, 根据香农公式可知此情况下的数据传输率的公式如式 (4-2) 所示。

$$r_{m0} = w_{m0} \log_2 \left(1 + \frac{p_{m0} |h_{m0}|^2}{\sigma^2 + \sum_{m=1}^M k_{m0} p_0 |h_{m0}|^2} \right). \quad (4-2)$$

(2) 任务卸载到小蜂窝基站边缘云的情况。在这个情况下, 本文用 w_{mn} 表示通信模型中设备 m , ($m \in \mathbb{M}$) 和小蜂窝基站边缘云 n , ($n \in \mathbb{N}$) 之间的信道带宽。此外, p_{mn} 用来表示设备 m , ($m \in \mathbb{M}$) 传输到小蜂窝基站边缘云 n , ($n \in \mathbb{N}$) 的传输功率, σ^2 表示高斯白噪声常量, h_{mn} 表示设备 m , ($m \in \mathbb{M}$) 和小蜂窝基站边缘云 n , ($n \in \mathbb{N}$) 之间的信道增益, $k_{mn} \in \{0,1\}$ 表示设备 m , ($m \in \mathbb{M}$) 和小蜂窝基站边缘云 n , ($n \in \mathbb{N}$) 的干扰参数。因此, 根据香农公式可知在此情况下的数据传输率的公式如式 (4-3) 所示。

$$r_{mn} = w_{mn} \log_2 \left(1 + \frac{p_{mn} |h_{mn}|^2}{\sigma^2 + \sum_{m=1}^M k_{mn} p_0 |h_{mn}|^2} \right). \quad (4-3)$$

4.2.2 计算模型

(1) 宏蜂窝基站边缘云的计算模型。任务从设备 $m, (m \in \mathbb{M})$ 卸载到宏蜂窝基站边缘云的时延公式为 $T_{m0} = T_{m0}^f + T_{m0}^c + T_{m0}^q$ 。其中, 传输时延 T_{m0}^f 由公式 $T_{m0}^f = \frac{A_m}{r_{m0}}$ 计算所得。宏蜂窝基站边缘云中计算时延公式为 $T_{m0}^c = \frac{B_m}{C_{m0}}$, C_{m0} 代表宏蜂窝基站边缘云的计算能力。宏蜂窝基站边缘云中任务 J_m 的最大排队时延为 $T_{m0}^q = \sum_{m=1}^M T_{m0}^f F\{x_m = 0\}$, 其中 $\{x_m = 0\}$ 代表设备 m 将任务 J_m 卸载到宏蜂窝基站边缘云。

此外, 宏蜂窝基站边缘云中所产生的计算能耗公式为 $E_{m0} = \eta_{m0} C_{m0}^3 T_{m0}^c$, 其中 η_{m0} 代表宏蜂窝基站边缘云的每个 CPU 周期能耗系数。

(2) 小蜂窝基站边缘云的计算模型。与宏蜂窝基站边缘云的计算模型类似, 某个设备 m 将任务 J_m 卸载到小蜂窝边缘云 n 的传输时延为 $T_{mn}^f = \frac{A_m}{r_{mn}}$, 小蜂窝基站边缘云的计算时延为 $T_{mn}^c = \frac{B_m}{C_{mn}}$, 其中 C_{mn} 是小蜂窝基站边缘云 n 的计算能力, 最大排队时延为 $T_{mn}^q = \sum_{m=1}^M T_{mn}^f F\{x_m = n\}$, 其中 $\{x_m = n\}$ 代表设备 m 将任务 J_m 卸载到小蜂窝基站边缘云 n 。基于以上模型, 某个设备 m 将任务 J_m 卸载到小蜂窝边缘云 n 的总时延为 $T_{mn} = T_{mn}^f + T_{mn}^c + T_{mn}^q$ 。

某个用户设备 m 将任务 J_m 卸载到小蜂窝基站边缘云 n 的总能耗为 $E_{mn} = \eta_{mn} C_{mn}^3 T_{mn}^c$, 其中 η_{mn} 是小蜂窝基站边缘云 n 每个 CPU 周期能耗系数。

4.2.3 计算卸载问题模型

(1) 通信资源约束。对某一可卸载的任务 J_m , 可以有两种可选择边缘计算的方式, 即小蜂窝基站边缘云计算和宏蜂窝基站边缘云计算。那么通信约束条件应该是, 为每个信道分配的带宽不超过无线链路的总带宽。在本节内容中, 假设每个小蜂窝基站的带宽相同。那么计算卸载问题模型将存在以下约束条件:

$$\sum_{m=1}^M w_{m0} F\{x_m = 0\} \leq w_{\max}^{MBS}, \quad \sum_{m=1}^M w_{mn} F\{x_m = n\} \leq w_{\max}^{SBS}。$$

(2) 计算资源约束。所卸载的计算任务不能超过每个计算节点的最大计算能力: $\sum_{m=1}^M A_m F\{x_m = 0\} \leq C_{\max}^{MBS}, \quad \sum_{m=1}^M A_m F\{x_m = n\} \leq C_{\max}^{SBS}。$

(3) 能耗约束。计算卸载的任务产生的能耗不能超过边缘云能耗的最大值:

$$\sum_{m=1}^M E_{m0} F\{x_m = 0\} \leq E_{\max}^{MBS}, \quad \sum_{m=1}^M E_{mn} F\{x_m = n\} \leq E_{\max}^{SBS}。$$

(4) 问题模型。根据以上模型和约束条件, 计算卸载问题的公式构建为:

$$\min \sum_{m=1}^M \sum_{n=1}^N (T_{m0} F\{x_m = 0\} + T_{mn} F\{x_m = n\}), \quad (4-4a)$$

$$\text{s. t.} \quad \sum_{m=1}^M w_{m0} F\{x_m = 0\} \leq w_{\max}^{MBS}, \quad (4-4b)$$

$$\sum_{m=1}^M w_{mn} F\{x_m = n\} \leq w_{\max}^{SBS}, \quad (4-4c)$$

$$\sum_{m=1}^M A_m F\{x_m = 0\} \leq C_{\max}^{MBS}, \quad (4-4d)$$

$$\sum_{m=1}^M A_m F\{x_m = n\} \leq C_{\max}^{SBS}, \quad (4-4e)$$

$$\sum_{m=1}^M E_{m0} F\{x_m = 0\} \leq E_{\max}^{MBS}, \quad (4-4f)$$

$$\sum_{m=1}^M E_{mn} F\{x_m = n\} \leq E_{\max}^{SBS}. \quad (4-4g)$$

该问题是二进制优化问题, 因此是个非凸的优化问题, 本章将采用启发式算法寻找满意解。

4.3 基于布谷鸟算法的计算卸载策略

4.3.1 布谷鸟搜索算法

布谷鸟搜索算法 (Cuckoo Search Algorithm, CSA) 是布谷鸟育雏行为和莱维飞行结合的一种启发式算法^[145], 该算法目前在很多工程领域的优化问题上取得了成功应用。本小节将对布谷鸟搜索算法进行了全面详尽的描述。

(一) 算法背景

自然界中有 1000 多种不同的鸟类, 在行为和特征上都很常见。例如, 所有的雌鸟都会产蛋, 但蛋的形状各不相同。许多鸟在安全的地方筑巢, 来增加巢穴的安全性。而有一类狡猾的鸟类, 被称为“孵卵寄生虫”。这些鸟自己不筑巢, 而是在另一种鸟的巢中产卵, 让寄主鸟照顾它的幼崽。布谷鸟就是这类鸟中的其中的一种, 它的策略是通过移除宿主鸟产下的一个鸟蛋并把自己产下的鸟蛋放入宿主巢中。然后, 布谷鸟通过模仿寄主鸟蛋的图案和颜色来改变它的蛋, 这项技能

需要高精度来确保布谷鸟的成功。布谷鸟产卵的时机也是选择寄主鸟刚刚产卵时,这一过程将在一段时间后获得利益;布谷鸟将在寄主蛋从巢中取出,利用寄主鸟的本能行为喂养布谷鸟蛋,从而增加了布谷鸟幼鸟的照料和食物供应机会。布谷鸟的狡猾具有遗传性,小布谷鸟会模仿寄主鸟类的叫声以获得更多的喂养机会。另一方面,如果宿主鸟认出了布谷鸟的蛋在自己的巢中,要么把布谷鸟蛋扔出去,要么干脆离开自己的巢建一个新的巢。因此,布谷鸟必须更准确地模仿寄主鸟的蛋,而寄主鸟必须提高其确定寄生蛋的技能,这就是所谓的生存斗争。

莱维飞行实质上是一种随机行走过程,它通过高频率的短距离飞行和低频率的长距离飞行来实现。低频率的长距离飞行,可以扩大搜索范围,增加种群的多样性,找到全局最优解。高频率的短距离飞行,可以实现小范围的搜索精度,缩小搜索范围,在小范围内实现全局最优。它的灵感来自于自然界中一些鸟类和果蝇的行为,是许多动物和昆虫觅食采用的随机行走过程。Levy 飞行由两个部分组成:一是方向,二是步长,方向是按照均匀分布进行选择,而步长选择是服从 Levy 分布的游走步长。莱维飞行在处理优化问题方面具有较好的效果。

(二) 算法介绍

2009 年,英国剑桥大学学者 Xin-She Yang 提出了一种新的启发式优化算法——布谷鸟搜索算法。Yang 和 Deb 提出了在优化环境中使用布谷鸟搜索算法的建议。到目前为止,对该算法的研究显著增加,并且布谷鸟搜索算法已经成功地其他优化方法中占据了适当的位置。这种算法是基于布谷鸟的寄生行为和莱维飞行。在布谷鸟搜索算法中,步长的选择是利用具有 Levy 分布特征的法则来完成的。Levy 飞行可以在不确定的区域中最大限度的进行有效的搜索。该算法比较简单,参数少,易于实现,具有很好的全局搜索能力,因此受到国内外许多学者的关注。布谷鸟搜索算法是一种有效的基于启发式的群智能优化算法,它能有效地在问题的搜索空间中平衡局部搜索和全局搜索。

1、布谷鸟搜索算法的规则

布谷鸟搜索算法模拟了布谷鸟寻找鸟窝的行为,并引入了鸟类的 Levy 飞行行为模式。布谷鸟搜索算法使用鸟窝位置代表解。假设每个鸟窝只有一个蛋,每一个蛋代表一个解。这样可以用更好的解来代替不好的解。为了区别于其他鸟类,布谷鸟有一种特殊的产卵方式。以下三个理想化规则对标准布谷鸟搜索算法进行了分类和描述:

- (1) 每只布谷鸟一次只产一个蛋,并会随机选择一个鸟窝来放置它产下的蛋;
- (2) 通过随机选择的方式寻找到的一组鸟窝中,将最好的一个鸟窝(解)保留到下一代;

(3) 鸟窝数量 n 固定不变, 另外鸟窝主人能够发现鸟窝中鸟蛋是外来的概率是 P_a , 其中 $P_a \in [0,1]$ 。如果鸟蛋被鸟窝主人发现, 鸟窝主人可以把该鸟蛋丢弃, 也可以直接抛弃这个鸟窝, 在其它地方建一个新的鸟窝。

2、布谷鸟搜索算法的数学模型

上述三个理想化条件表明每一个鸟巢只包含一个鸟蛋, 并不会发生一个鸟巢有很多鸟蛋的情况。每一代只会保留最好的鸟巢位置, 然后通过随机更新的方式更新其它的鸟巢位置, 如果更新后的新位置优于旧位置, 就保留新位置, 舍弃旧位置。此外, 还设定了一个宿主鸟发现鸟蛋的概率 P_a , 如果宿主鸟发现某个鸟巢中的鸟蛋的概率大于 P_a , 就说明宿主鸟发现这个鸟巢中的鸟蛋不是自己所产鸟蛋的概率很大, 然后就会扔掉自己鸟巢中的鸟蛋或者放弃鸟巢并建立一个新的鸟巢。

布谷鸟搜索算法主要核心是两个位置更新公式, 其中一个进行的是局部搜索, 另一个进行的是全局搜索。全局搜索位置更新公式为式 (4-5)。

$$x'_{ik} = x'_{ik} + 0.01 \times \alpha \times R \times Levy \times (x'_{jk} - p'_{gk}). \quad (4-5)$$

其中, 鸟巢 i 在第 t 代中的位置是 $x'_i = \{x'_{i1}, x'_{i2}, \dots, x'_{iM}\}$, M 表示维数。 x'_i 是鸟巢 i 在第 t 代中位置 x_i 经过全局更新后产生的新位置, $\alpha > 0$ 是 $[0,1]$ 之间的随机数, 表示步长, 应与利益问题的规模相关, 一般情况下设 $\alpha=1$ 。 x'_{ik} 表示布谷鸟 i 在 k 维上的数值, 类似地, x'_{jk} 表示布谷鸟 j 在 k 维上的数值。 $p'_g = \{p'_{g1}, p'_{g2}, \dots, p'_{gM}\}$ 是整个搜索空间中算法所能找到的当前最优解。 R 是随机数, 服从 $(0,1)$ 之间的均匀分布。通过莱维飞行的随机行走能更有效地在搜索空间找到解, 因为它的步长在长期内要长得多。 $Levy(\beta)$ 服从 Levy 分布: $Levy(\beta) \sim u = t^{-1-\beta}, (0 < \beta \leq 2)$, 其中 β 是一个参数, 它是在一个单位间隔内事件发生的平均值或期望值, Levy 分布通过公式 (4-6) 产生。

$$Levy(\beta) \sim \frac{\phi \times \mu}{|v|^{1/\beta}}. \quad (4-6)$$

其中 μ 和 v 服从标准正态分布, $\beta=1.5$ 。

$$\phi = \left(\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta / 2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{(\beta-1)/2}\right)} \right). \quad (4-7)$$

通过 Levy 飞行进行完全局搜索之后, 还会有一部分解再次进行一次局部搜索更新位置。最后比较保留好的一组解。 x'_i 随机搜索的位置更新公式如 (4-8) 所示。

$$x_{ik}^{t+1} = x_{ik}^t + R \times (x_{sk}^t - x_{uk}^t). \quad (4-8)$$

其中 R 是随机数, 服从 $(0,1)$ 之间的均匀分布, x_{sk}^t 和 x_{uk}^t 是第 t 代时随机生成的两个位置 s 和 u 。

基于以上假设, 布谷鸟算法主要分为四步:

第 1 步: 初始化。根据搜索空间的维度随机产生鸟巢位置;

第 2 步: 值的更新。鸟巢基于莱维飞行模式产生新的位置, 从第 t 代到第 $t+1$ 代的位置;

第 3 步: 局部搜索。随机选取一部分位置进行更新, 通过适应值函数比较最后保留最优解;

第 4 步: 重复第 2、3 步直到满足最大迭代次数。

3、布谷鸟搜索算法的参数

与许多其他搜索技术相比, 布谷鸟搜索算法的最大优点之一是, 它使用的控制参数更少。布谷鸟搜索算法的主要参数设置如表 4-1 所示。除了以下控制参数以外, 还有鸟蛋的数量、种群规模、最大迭代次数、适应度函数和问题维度等构成要素。

表 4-1 布谷鸟搜索算法参数设置

Tab.4-1 Parameters of Cuckoo Search Algorithm

参数	符号	范围	常用
鸟巢数量	N	$[15, 50]$	15
被发现的概率	P_a	$[0, 1]$	0.25
步长	α	$[0, +\infty)$	1

4.3.2 基于布谷鸟搜索算法的计算卸载策略

由于基于软件定义网络的工业物联网场景下的边缘计算卸载问题被表述为非凸优化问题。最近, 许多启发式算法为求解该类问题的近似最优解展现了足够的可行性。因此, 本节将采用布谷鸟算法来解决边缘计算卸载问题。首先介绍算法的搜索模式。在二维搜索空间中, 布谷鸟算法中粒子符合图 4-3 所示的搜索模式。具体而言, 图 4-3 (a) 示出了全局搜索模式, 图 4-3 (b) 示出了局部搜索模式。通过布谷鸟算法的搜索模式发现, 标准的布谷鸟算法是为了处理连续性问题而设计的, 不能直接应用于卸载问题。因此, 本节将提出改进的离散布谷鸟算法, 以便有效地解决计算卸载问题, 具体改进方法如下所述。

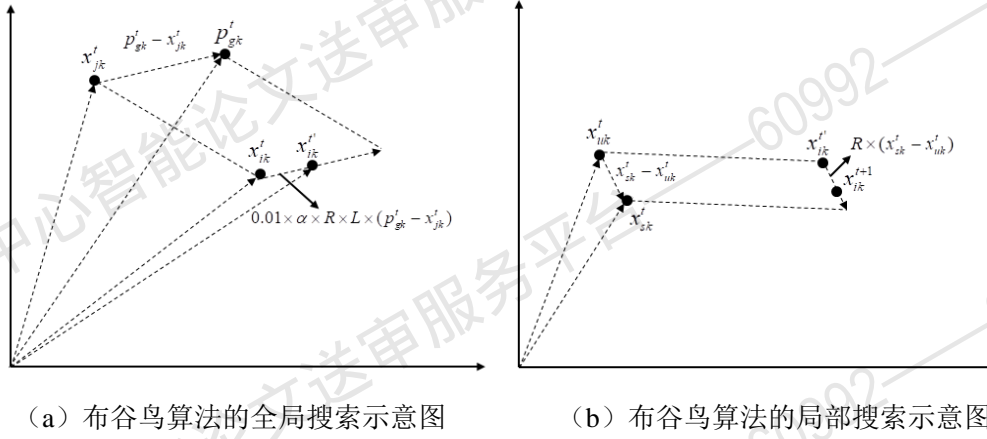


图 4-3 布谷鸟算法搜索示意图

Fig. 4-3 The search graph of Cuckoo Search Algorithm

算法 4-1 基于布谷鸟算法的边缘计算卸载策略

输入：工业物联网参数：宏蜂窝基站边缘云，小蜂窝基站边缘云集合 $\mathbb{N} = \{1, 2, \dots, N\}$ ，工业物联网设备 $\mathbb{M} = \{1, 2, \dots, M\}$ ，任务集合 $J = \{A, B, T^{\max}\}$ ，数据传输速率 r_{m0} ，传输功率 p_{m0} ，信道增益 h_{m0} ，数据传输功率 r_{mn} ，传输功率 p_{mn} ，信道增益 h_{mn} ，高斯噪声 σ^2 。

布谷鸟算法参数：布谷鸟个数 N ，鸟窝位置集合 $\{x_1, x_2, \dots, x_N\}$ ，最大迭代次数 i_{\max} ，被发现概率 P_a ，移动权重因子 α 。

输出：最优计算卸载策略 x^* 。

初始化：

针对每个布谷鸟，初始化算法参数；

Begin:

For $i < i_{\max}$ **do**

根据公式 (4-5) 更新布谷鸟算法的位置；

根据公式 (4-9) 和公式 (4-10) 将位置离散化；

根据公式 (4-4) 计算并获取较优解；

If $r > p_a$ **then**

更新位置，并根据公式 (4-9) 和公式 (4-10) 将位置离散化；

计算并获取最优解；

End if

输出最优计算卸载策略 x^* 。

End for

End

标准布谷鸟算法的位置更新是为解决连续问题而设计的，而在计算卸载问题的解空间应该是 $\{0, 1\}$ 。因此，本节设计的改进方案是，在布谷鸟算法的搜索过程之后重新添加一个新的算子，从而帮助解决方案满足要求。首先，将布谷鸟算法在每次迭代次数之后对位置重新排序，并计算出在 k 维上的最小值 x_k^{\min} 和最大值 x_k^{\max} ，此操作算子 ϕ 的公式如式 (4-9) 所示。

$$\phi(x_{ik}(t+1)) = \frac{x_{ik}(t+1) - x_k^{\min}}{x_k^{\max} - x_k^{\min}}. \quad (4-9)$$

根据公式 (4-9) 的操作, 布谷鸟算法的位置被映射到 $[0,1]$ 之间, 接下来的操作算子被用来修改位置从而满足解空间的要求。通过以下公式 (4-10), 位置根据判断条件将位置离散为 0 或者 1, 此外, 在本节中 $rand(a)$ 设置为 0.5。

$$x_{ik}(t+1) = \begin{cases} 0, & \text{if } rand(\bullet) < rand(a), \\ 1, & \text{otherwise.} \end{cases} \quad (4-10)$$

根据标准布谷鸟算法和以上的离散操作算子, 基于布谷鸟算法的计算卸载策略的伪代码如算法 4-1 所示。

4.3.3 布谷鸟搜索算法的收敛性分析

本小节采用确定的马尔可夫链分析基于布谷鸟算法的边缘计算卸载策略的全局收敛性。首先, 将先介绍确定的马尔可夫链相关的概念和定理。然后, 利用确定的马尔可夫链定理证明了全局收敛性。更多理论推导细节可以参考文献 [148]。

定义 4-1 假设存在随机变量序列 $x_t: t=1,2,\dots$, 且 x_t 的确定状态空间是 $S=\{a_1, a_2, \dots, a_n\}$ 。如果存在一个随机值 $k>0$ 和 $a_i, a_j, a_{i,0}, a_{i,1}, \dots, a_{i,k-1} \in S$, 那么公式 (4-11) 是可行的。

$$\begin{aligned} P(x_{k+1} = a_j | x_0 = a_{i,0}, x_1 = a_{i,1}, \dots, x_{k-1} = a_{i,k-1}, x_k = a_i) \\ = P(x_{k+1} = a_j | x_k = a_i). \end{aligned} \quad (4-11)$$

其中 x_t 的确定状态空间被认为确定的马尔科夫链。条件概率 $P(x_{k+1} = a_j | x_k = a_i)$ 定义为在 t 时刻下的状态 a_i 转移到 $t+1$ 时刻下的状态 a_j 的转移概率。此外, 条件概率用 $p_{ij}(t)$ 表示。

定义 4-2 如果转移概率与时刻 t 无关, 即 $p_{ij}(t_1) = p_{ij}(t_2) = p_{ij}, a_i, a_j \in S$, 那么这个确定的马尔科夫链是同质的。假设 $P = (p_{ij})_{n \times n}$ 是确定同质的马尔科夫链的转移概率矩阵, 且 $0 \leq p_{ij} \leq 1$ 和 $\sum_{j=1}^n p_{ij} = 1, i=1,2,\dots,n$ 。

$$\begin{aligned} p_j(t) &= p\{x_t = a_j\} \\ &= \sum_{i=1}^n P(x_t = a_j | x_{t-1} = a_i) P(x_{t-1} = a_i) \\ &= \sum_{i=1}^n p_{ij} p_i(t-1), j=1,2,\dots,n. \end{aligned} \quad (4-12)$$

$$\begin{aligned}
& (p_1(t), p_2(t), \dots, p_n(t)) \\
&= (p_1(t-1), p_2(t-1), \dots, p_n(t-1)) \times P \\
&= (p_1(t-2), p_2(t-2), \dots, p_n(t-2)) \times P^2 \\
&= \dots = (p_1(0), p_2(0), \dots, p_n(0)) \times P^t.
\end{aligned} \tag{4-13}$$

从公式 (4-12) 和公式 (4-13) 分析过程可知, 确定同质的马尔科夫链依赖于初始分布和它的转移概率。

定义 4-3 假设 $A = (a_{ij})$ 是 n 阶非负实矩阵, 那么该矩阵具有以下属性。

- (1) 矩阵 A 是本原的。若存在正整数 k , 使得 A^k 是正的, 则称 A 是本原的;
- (2) 矩阵 A 是可约的。若存在方阵 C, T , 使得通过相同的行和列的置换可将矩阵 A 变换成形式 $\begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$, 则称矩阵 A 是可约的, 否则称矩阵 A 是不可约;

- (3) 矩阵 A 是随机的。若 $\sum_{j=1}^n a_{ij} = 1, i=1, 2, \dots, n$, 则称矩阵 A 是随机的;

- (4) 矩阵 A 是稳定的。若矩阵 A 有相同的行, 则称矩阵 A 是稳定的。

定理 4-1 假设 P 是一个 n 阶本原的随机矩阵, 那么当 $k \rightarrow \infty$, P^k 能够收敛到一个正定稳定随机的矩阵。

$$\begin{aligned}
P^\infty &= \lim_{k \rightarrow +\infty} P^k = \lim_{k \rightarrow +\infty} \begin{pmatrix} C^k & \dots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \dots & T \end{pmatrix} \\
&= \begin{pmatrix} C^\infty & \dots & 0 \\ R^\infty & \dots & T \end{pmatrix}
\end{aligned} \tag{4-14}$$

其中 P^∞ 是稳定随机矩阵, 满足以下条件:

$$P^\infty = (p_{ij})_{n \times n} = \begin{cases} p_{ij} > 0, 1 \leq i \leq n, 1 \leq j \leq m, \\ p_{ij} = 0, 1 \leq i \leq n, m \leq j \leq n. \end{cases} \tag{4-15}$$

接下来, 根据以上定义和定理, 基于布谷鸟算法的计算卸载策略将被证明能够收敛到稳定状态。由于布谷鸟个数是 L , 每个布谷鸟的维数是 M , 布谷鸟算法的状态空间是 $\Omega = \{0, 1\}^{L \times M}$ 。

首先, 布谷鸟算法的转移概率是 $p_{ij} \in [0, 1]$, 那么基于布谷鸟算法的计算卸载策略的转移概率矩阵 P 是正定的。因此, 很明显转移概率是本原的。

其次, 基于引理 4-1, 布谷鸟算法的转移概率 P 可以被分解为矩阵 $C, R, T, 0$ 。

引理 4-1 在布谷鸟算法中, $\forall X^i, X^k \in X_s, i=1, 2, \dots, |F|, k=1, 2, \dots, |F|$, 满足以下条件:

$$\forall k > i, p_{k,i} = 0. \tag{4-16}$$

$$\forall k < i, p_{k,i} > 0. \tag{4-17}$$

证明过程如下：

(1) 公式 (4-16) 的证明。假设算法在迭代次数 t 的位置 X^t 状态记为 x^t 。类似地，算法在迭代次数 $t+1$ 的位置状态记为 x^{t+1} 。根据布谷鸟算法的更新模式，可以得到

$$Fitness(x^{t+1}) \geq Fitness(x^t) \Rightarrow \forall k > i, p_{k,i} = 0. \quad (4-18)$$

(2) 公式 (4-17) 的证明。假设布谷鸟在迭代次数 t 时的最优位置是 x_*^t ， x_*^{t+1} 代表布谷鸟在迭代次数 $t+1$ 时的最优位置。在布谷鸟算法中，如果 $p_a > 0$ ，每个布谷鸟位置将会选择搜索公式 (4-5)，该模式有利于算法跳出局部最优。基于此，布谷鸟向更优位置的方向移动， $Fitness(x_*^{t+1}) \geq Fitness(x_*^t)$ ，同时 $\exists k < i, p_{k,i} > 0$ 证明完毕。

根据引理 4-1，可以得到转移概率矩阵 P

$$P = \begin{pmatrix} p_{1,1} & 0 & \cdots & 0 \\ p_{2,1} & p_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ p_{|S|,1} & p_{|S|,2} & \cdots & p_{|S|,|S|} \end{pmatrix} = \begin{pmatrix} C & \cdots & 0 \\ R & \cdots & T \end{pmatrix}. \quad (4-19)$$

特别地， $C = (p_{1,1}) = 1 \neq 0$ ， $R = (p_{2,1}, p_{3,1}, \cdots, p_{|S|,1})^T$ ，和 $T = \begin{pmatrix} p_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ p_{|S|,2} & \cdots & p_{|S|,|S|} \end{pmatrix}$ 。

最后，基于以上公式， C^∞ 等于 1， P^∞ 是一个稳定随机的矩阵，且有非零项，所以 $R^\infty = (1, 1, \cdots, 1)^T$ 。因此

$$P = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (4-20)$$

可以得出 $\lim_{k \rightarrow +\infty} P\{Fitness(x^t) = Fitness(x_*)\} = 1$ ，其中 $Fitness(x_*)$ 代表解空间里的最优适应值，同时也是计算卸载优化问题的最优策略。总而言之，根据以上分析，基于布谷鸟算法的计算卸载策略的收敛性证明完毕。

4.4 仿真结果

在本节中，边缘计算卸载策略将在 Matlab 2013 上进行了仿真和评估。首先，第 4.4.1 小节介绍了仿真实验场景和参数。然后，为了证明该方案的有效性，在第 4.4.2 小节将实验结果与其他现有算法进行了比较，并讨论了算法和任务的参数影响。

4.4.1 仿真实验参数设置

图 4-4 描述了工业物联网的仿真实验场景，工业物联网场景包括三个部分：宏蜂窝基站边缘云，小蜂窝基站边缘云和用户设备。宏蜂窝基站边缘云的个数设置为 1，且它的传输半径最多为 500m。小蜂窝基站边缘云的个数设置为 10，能够为 50 个物联网设备提供给计算服务，物联网设备随机的部署在边缘云网络中。

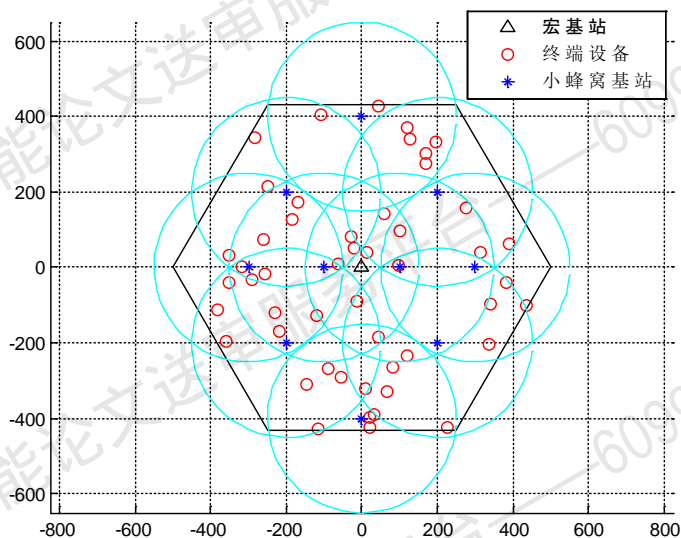


图 4-4 工业物联网场景

Fig. 4-4 Scenario of IIoT

从物联网设备到宏蜂窝基站边缘云的传输信道带宽设置为 5MHz，此外，从物联网设备到小蜂窝基站边缘云的传输信道带宽设置为 180KHz。物联网设备的传输功率从[50mW, 100mW]随机产生，高斯噪声干扰变量设置为 $\sigma^2 = -100$ dBm，干扰因子为 -100 dBm。信道增益表示 $h = d^{-\lambda}$ ，其中 λ 代表路径损失因子且设置为 4。每个物联网设备产生的任务量范围是[10M, 20M]。CPU 的个数范围是[3, 10]，每个任务的时延容忍范围是[0.05s, 3s]。边缘云服务器的计算能力是 5GHz，单元能耗是 4J/GHz。为了验证本章提出的计算卸载策略的有效性，遗传算法（Genetic algorithm, GA）^[139]和粒子群算法（Particle swarm algorithm, PSO）^[138]用于做测试比较。粒子群算法是一种经典的仿生优化算法，具有快速收敛能力。每个算法的粒子个数设置为 20，迭代次数设置为 50。表 4-2 给出了每个算法的用于求解计算卸载问题的具体参数，参数设置参考不同启发式算法的常用设置。具体来说，对于布谷鸟算法，鸟蛋被发现的概率设为 $P_a = 0.25$ ，移动的权重因子是 $\alpha = 1$ 。对于粒子群算法， c_1 代表个体影响因子， c_2 代表社会影响因子， w 代表上一代速度的权重因子，从 0.9 线性递减到 0.4。对于遗传算法， P_c 和 P_m 是影响算法性能的两个重要参数， P_c 代表基因染色体的交叉概率， P_m 代表基因染色

体的变异概率。实验结果选取不同算法运行 10 次中的最优结果。

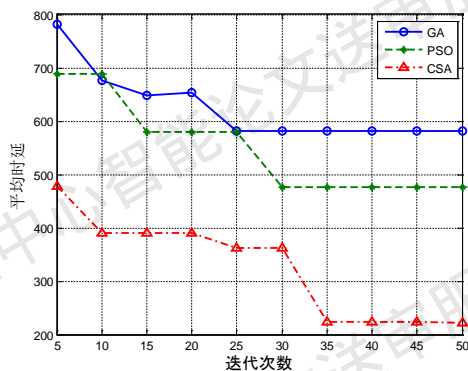
表 4-2 算法实验参数设置

Tab.4-2 Parameters of Algorithms

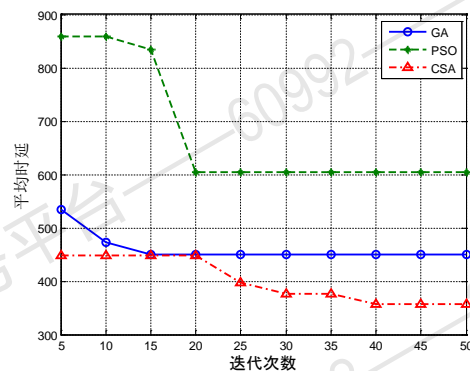
算法	参数
布谷鸟算法	$P_{\alpha} = 0.25, \alpha = 1$
粒子群算法	$c_1 = 2.0, c_2 = 2.0, w = [0.4, 0.9]$
遗传算法	$P_c = 0.8, P_m = 0.1$

4.4.2 数值仿真结果

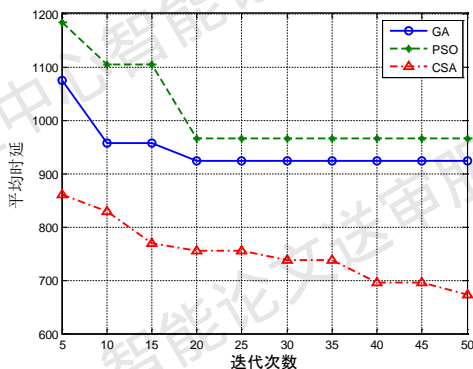
图 4-5 示出了模拟在工业物联网区域中随机分布的不同数目的物联网设备之间的平均延迟时间变化。具体而言,物联网设备的数目被设置为[50, 60, 70, 80, 90, 100],以测试设备个数对实验结果的影响。与其他两种算法相比,布谷鸟算法在不同的物联网设备数目下的平均延迟时间最低。相比之下,粒子群算法的性能稍逊于布谷鸟算法。特别是在图 4-5 (d) 和图 4-5 (e) 中,粒子群算法能够实现接近布谷鸟算法的实验结果。此外,遗传算法在所有子图的性能最差。从图 4-5 中的不同子图,可以得出物联网设备的数目在算法上具有显著的影响。相对来说,更多的物联网设备需要更大的计算复杂度。



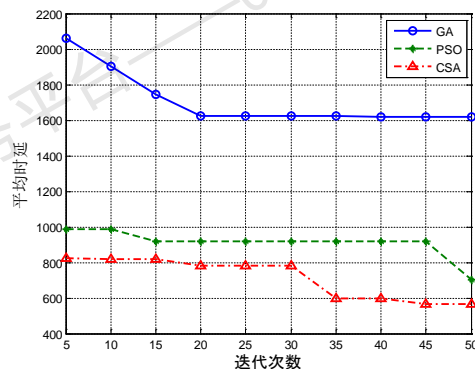
(a) 物联网设备个数为 50



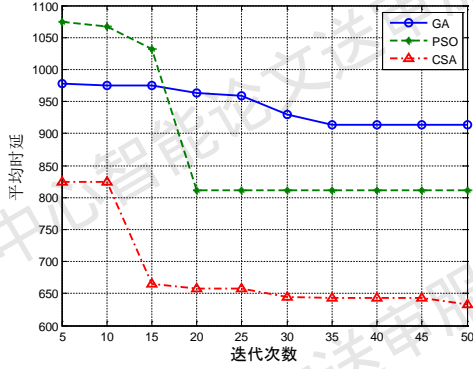
(b) 物联网设备个数为 60



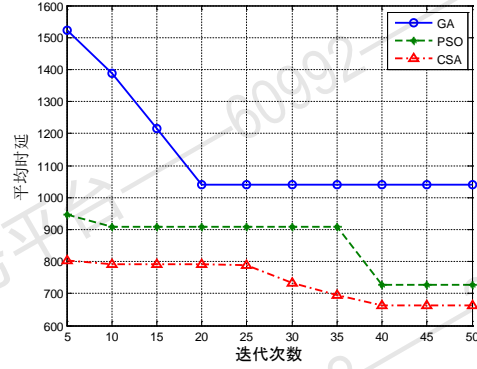
(c) 物联网设备个数为 70



(d) 物联网设备个数为 80



(e) 物联网设备个数为 90

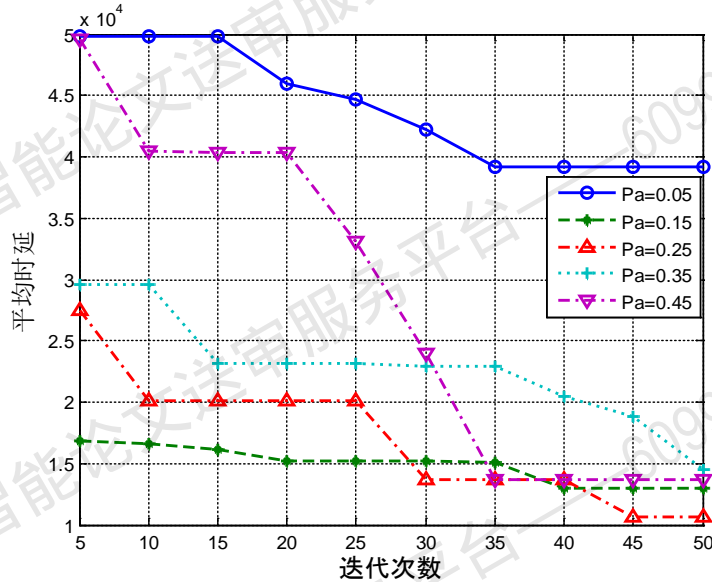


(f) 物联网设备个数为 100

图 4-5 不同物联网设备个数下的平均时延变化

Fig. 4-5 Average delay time among different number of UEs

随后, 本小节对布谷鸟算法的两个关键参数 (P_a 和 α) 进行了实验仿真。如图 4-6 (a) 所示, 随着迭代次数的增加, 收敛速度随 P_a 的不同而变化。由此可见, $P_a=0.05$ 表现最差, $P_a=0.25$ 表现最好。 P_a 表示布谷鸟蛋被发现的概率, 实验结果说明当 P_a 过低时, 布谷鸟蛋不易被发现, 无需重新寻找新的鸟巢位置, 因此算法就容易陷入局部最优。此外, 当迭代次数等于 50 时, $P_a=0.15$ 、 $P_a=0.35$ 和 $P_a=0.45$ 能够求出近似的平均延迟时间。从图 4-6 (b) 来看, $\alpha=0.01$ 具有最高的平均延迟时间, 而 $\alpha=10$ 具有最佳的收敛能力。

(a) 参数 P_a 对算法的影响

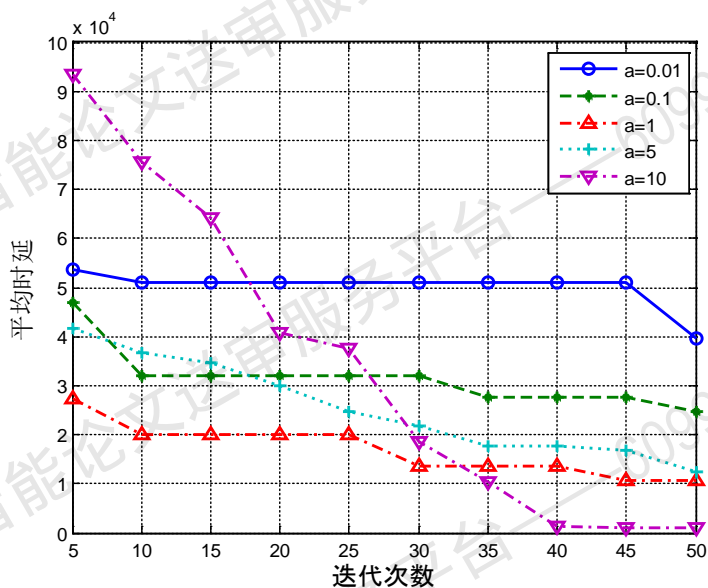
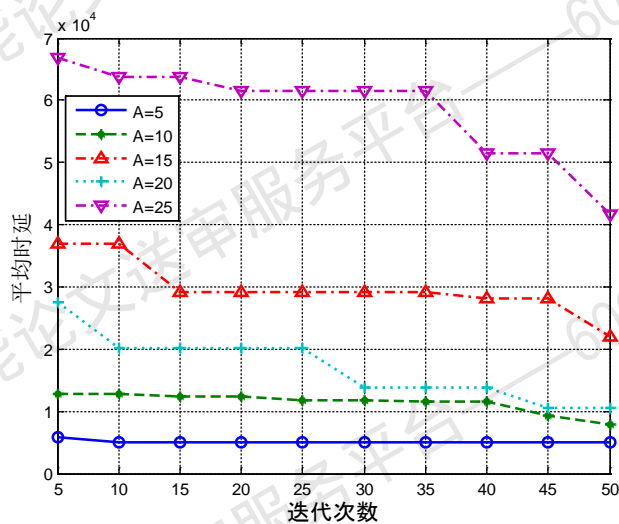
(b) 参数 α 对算法的影响

图 4-6 不同布谷鸟算法参数下的平均时延变化

Fig. 4-6 Average delay time among different factors of CSA

最后, 根据所提出的方案, 分析了不同任务参数对平均时延结果的影响。如图 4-7 所示, 本节讨论了任务的三个属性值 (A, B, T), 其中 $A=5$, $B=7$, $T=0.01$ 是相对最优的结果。特别是在图 4-7 (a) 中, $A=5$ 在迭代过程中表现明显优于其他取值下的结果, 说明任务量越小, 算法性能表现越好。从图 4-7 (b) 可以看出, 根据本章提出的方案, B 值越小, 结果越好。此外, 在图 4-7 (c) 中, 任务所能容忍的 T 值越大, 平均延迟时间就会显著增加。

(a) 任务参数 A 的影响

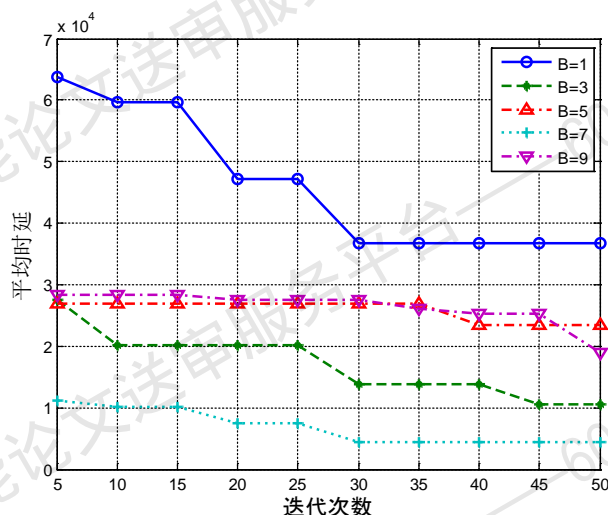
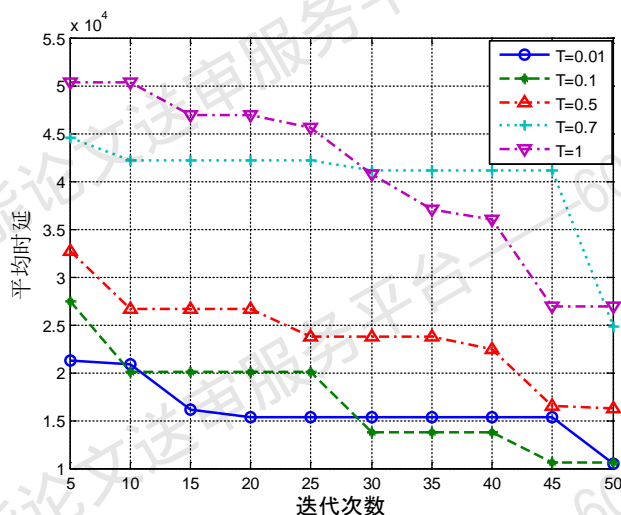
(b) 任务参数 B 的影响(c) 任务参数 T 的影响

图 4-7 不同任务参数下的平均时延变化

Fig. 4-7 Average delay time among different factors of task

4.5 本章小结

本章提出了一种在基于软件定义网络的工业物联网架构下的计算卸载策略。在这种新的网络架构中,所提出的布谷鸟算法计算卸载策略能够在考虑整个网络系统平均延迟时间的情况下,为工业物联网设备提供合理的计算卸载决策。此外,通过马尔科夫收敛过程分析全局收敛性,证明了该方法的有效性。仿真结果表明,与其他对比算法相比,该方案具有更好的性能。最后,讨论了算法参数和任务参数的影响,验证了该算法的不同参数对实验结果的影响。

第5章 面向经济效益的多用户计算卸载服务策略

5.1 引言

随着工业信息化的迅猛发展,无线网络已被视为未来工业体系转型中极为重要的一部分。依靠广泛的无线和智能设备的应用,工业物联网^[149]可以通过高质量的监测和控制来保护工业能力。然而,随着网络规模的不断扩大,工业物联网越来越复杂,面临着传输机制不合理、带宽有限、传输不可靠、能耗增加等前所未有的挑战^[150]。

为了解决这些问题,传统的云计算技术^[151]能够为工业物联网的用户设备提供计算卸载服务,但是长延迟无法满足设备的服务质量。因此,为了克服云计算的瓶颈,满足低延时、低能耗、高可靠性的计算服务需求,移动边缘计算^[152]技术将云计算的功能从网络中心转移到边缘。与云计算相比,边缘计算具有以下优势:一方面,它可以有效地缩短用户设备与云端之间的传输距离。另一方面,它可以避免大量数据交换造成的网络拥塞,提高网络回传能力。然而,边缘计算不能像云计算那样拥有足够的计算能力和存储能力,因此,设计一个有效的计算卸载策略是十分必要的,能够为所有工业物联网设备提供有效可靠的计算服务。

在工业物联网中,不同类型的边缘云相互竞争,提供灵活的接入,满足工业物联网设备的计算卸载服务。同时,工业物联网设备可以任意选择不同的边缘云来获得满意的计算服务。在这种情况下,工业物联网设备之间存在竞争关系,它们倾向于以更好的服务质量和更低的支付费用访问边缘云。同时,为了吸引更多的客户,每个边缘云需要设置适当的价格最大化他们的利润。边缘云与工业物联网设备之间的交互作用可以被认为是一个典型的斯坦科尔伯格博弈问题^[153-154]。

本章提出了斯坦科尔伯格博弈模型来分析边缘云和工业物联网设备之间的交互作用。其中,边缘云代表领导者,工业物联网设备代表追随者。对于工业物联网设备,来自其他设备的社会交互信息能够为每个可能的边缘云提供建设性评价。此外,潜在的社会交互信息可以帮助工业物联网设备做出适当的决定。根据现实场景中是否可以直接获取社会交互信息,本章考虑完全信息和不完全信息两种情况,建立了斯坦科尔伯格博弈模型。通过博弈数学理论验证了效用函数满足价格和需求策略下模型的凹性,保证了纳什均衡的存在。最后,采用动态迭代算法获得子博弈的纳什均衡解,从而保证了边缘云和工业物联网设备获得最价格和需求值的满意解。本章的主要工作可归纳如下:

- (1) 建立了斯坦科尔伯格博弈模型来解决面向经济效益的多用户计算卸载

服务问题。边缘云表示领导者，工业互联网设备表示追随者。为了考虑对边缘云的合理评价，在构建工业互联网设备的效用函数模型中引入了社会交互信息。

SDWN 控制器负责获取网络状态，为计算卸载服务策略做出合理决策；

(2) 在其他工业互联网设备的完全社会交互信息的情况下，设计了一种基于斯坦科尔伯格博弈的计算卸载策略，并通过博弈理论证明了斯坦科尔伯格博弈均衡点的存在性和唯一性；

(3) 在不完全的社会交互信息的情况下，设计了一种基于贝叶斯斯坦科尔伯格博弈的计算卸载策略。社会交互信息的期望值通过工业互联网设备的输入度和输出度来估计。此外，还通过博弈理论证明了贝叶斯斯坦科尔伯格博弈均衡点的存在性和唯一性；

(4) 仿真结果验证了所提出的博弈方法在完全信息和不完全信息情形下的有效性。实验表明，该方法在完全信息和不完全信息下，可使总需求值和总效用值达到近似最优。

如图 5-1 所示，面向经济效益的多用户计算卸载服务问题将在工业互联网架构中讨论。该架构由各种工业互联网设备和边缘云组成。工业互联网设备包括不同应用场景中（例如机器人、车辆和工厂设备等）连接到工业互联网中的用户设备。边缘云由具有边缘服务器的基站（如宏蜂窝基站和小蜂窝基站）组成，为工业互联网设备提供计算卸载服务。在介绍具体内容之前，本章出现主要的符号总结在表 5-1 中。

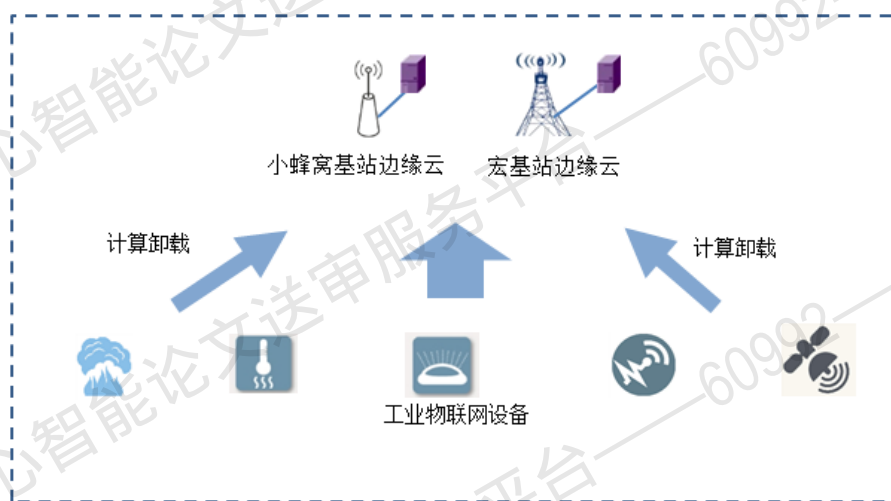


图 5-1 工业互联网架构

Fig. 5-1 IIoT Architecture

表 5-1 本章的符号定义
Table 5-1 Description of Notations

符号	定义
$\mathcal{N} = \{1, 2, \dots, N\}$	边缘云集合
N	边缘云的个数
$\mathcal{M} = \{1, 2, \dots, M\}$	工业物联网设备集合
M	工业物联网设备的个数
J_m	工业物联网设备的任务
Q_m	任务量大小
B_m	任务所需的 CPU 个数
T_m^{\max}	每个任务的最大时延容忍
w_{mn}	工业物联网设备和边缘云之间的信道带宽
p_{mn}	工业物联网的传输功率
σ^2	信噪功率系数
h_{mn}	工业物联网设备和边缘云之间的信道增益
T_{mn}	工业物联网设备和边缘云之间的全部时延
E_{mn}	工业物联网设备和边缘云之间的能耗
Q	工业物联网设备的计算需求集合
\mathcal{P}	边缘云的价格集合

5.2 系统模型

5.2.1 通信模型

本章考虑了随机分布在无线网络中的各种边缘云和工业物联网设备的工业物联网体系结构。 $\mathcal{N} = \{1, 2, \dots, N\}$ 代表了边缘云集合， $\mathcal{M} = \{1, 2, \dots, M\}$ 代表了工业物联网设备的集合。此外，假设边缘云能够通过无线链路彼此通信。每个工业物联网设备中的任务可以卸载到边缘云，而且每个工业物联网设备只能同时与一个边缘云通信。假设在工业物联网设备 m 中有需要计算服务的任务 $J_m = \{Q_m, B_m, T_m^{\max}\}$ ，其中 Q_m 代表传输到边缘云的任务量， B_m 代表任务 J_m 计算所需的 CPU 个数， T_m^{\max} 代表任务 J_m 计算最大的时延容忍。根据不同的

$\{Q_m, B_m, T_m^{\max}\}$, 任务可以分为多种类型的任务, 这些任务可以通过边缘云来完成。不同类型的任务可以是时延敏感或能耗敏感的计算任务^[155]。然而, 本章更关注的是社会交互信息在计算卸载模型中的影响。

在本章的通信模型中, w_{mn} 代表边缘云 $n(n \in \mathcal{N})$ 和工业物联网设备 $m(m \in \mathcal{M})$ 的信道带宽。 ρ_{mn} 表示工业物联网设备 m 的传输功率。 σ^2 表示高斯白噪声常量。 h_{mn} 表示工业物联网设备 m 和边缘云 n 之间的信道增益。基于以上符合, 根据香农公式可知此情况下的数据传输率 r_{mn} 的公式如公式 (5-1) 所示。

$$r_{mn} = w_{mn} \log_2 \left(1 + \frac{\rho_{mn} |h_{mn}|^2}{\sigma^2 + \sum_{j=1, j \neq m}^M k_{jn} \rho_n |h_{jn}|^2} \right). \quad (5-1)$$

$k_{jn} \in \{0,1\}$ 表示工业物联网设备 m 和边缘云 n 的影响系数。工业物联网设备 j 是除了工业物联网设备 m 之外选择边缘云的设备。

5.2.2 计算模型

在本节的问题模型中, 计算模型考虑两种影响因素: 时延和能耗。下面将具体介绍两种因素的模型公式。

(1) 全部时延。基于以上的任务模型, 任务 J_m 从工业物联网设备 m 卸载到边缘云 n 传输时延 T_{mn}^f 表示为公式 $T_{mn}^f = \frac{Q_m}{r_{mn}}$ 。边缘云 n 中计算任务 J_m 的时延公式

为 $T_{mn}^c = \frac{B_m}{C_{mn}}$, C_{mn} 代表边缘云的计算能力。不同的边缘云类型具有不同的计算能力, 本章定义宏蜂窝基站边缘云的计算能力大于小蜂窝基站边缘云的计算能力。

边缘云中任务 J_m 的最大排队时延为 $T_{mn}^q = \sum_{m=1}^M T_{mn}^f F\{x_m = 0\}$, 其中 $\{x_m = 0\}$ 代表设备 m 将任务 J_m 卸载到边缘云。从工业物联网设备 m 卸载到边缘云 n 的全部时延公式为 $T_{mn} = T_{mn}^f + T_{mn}^c + T_{mn}^q$ 。

(2) 能耗。任务 J_m 在边缘云 n 的计算能耗公式为 $E_n^c = \eta_n T_{mn}^c$, 其中 η_n 代表边缘云 n 的每个 CPU 周期能耗系数。

从工业物联网设备 m 卸载到边缘云 n 的传输能耗公式为 $E_{mn}^c = \rho_{mn} T_{mn}^f$ 。

5.2.3 斯坦科尔伯格模型

如图 5-2 所示, 边缘云和工业物联网设备之间的交互可以被建模为两阶段的

斯坦克伯格博弈框架。在第一阶段，每个服务提供商（即边缘云）设置计算资源的最佳价格。在第二阶段，用户（即工业物联网设备）根据边缘云的价格确定所需的最佳计算资源量。

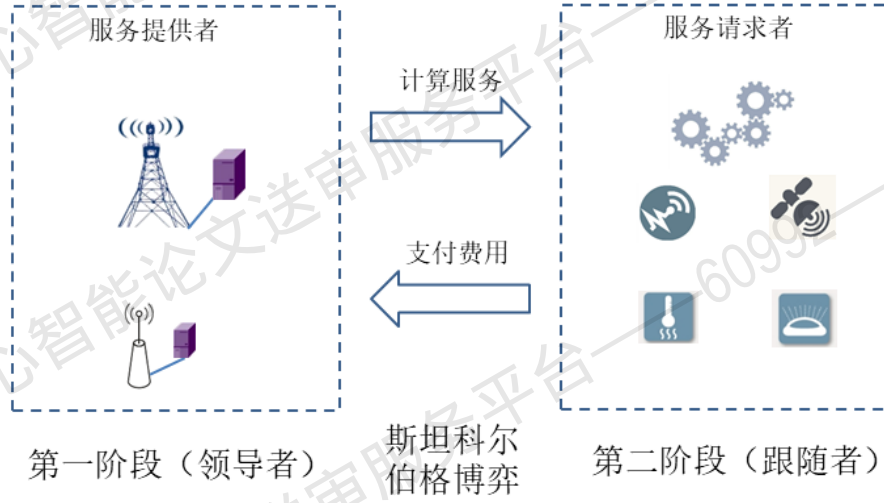


图 5-2 工业物联网中基于斯坦科尔伯格的计算卸载策略

Fig. 5-2 Stackelberg Model based Computation Offloading in the IIoT

斯坦科尔伯格博弈模型由服务提供者、消费者、效用函数和行动策略四部分组成。边缘云表示为消费者的服务提供者，它宣布将计算资源出售给工业物联网设备的价格，工业物联网设备称作消费者，它们为边缘云付费并从边缘云接收计算服务。行动策略包括计算需求集和价格集。此外，假设工业物联网设备的计算需求集合为 $Q = \{Q_1, Q_2, \dots, Q_M\}$ ，边缘云的计算服务定价集合为 $P = \{P_1, P_2, \dots, P_N\}$ 。本章模型中的服务提供者和消费者的效用函数如下：

(1) 工业物联网设备的效用函数。本章考虑了社会交互信息对工业物联网设备的影响。对于工业物联网设备，每个边缘云的评价都是通过对数函数的满意度来建模的。除了考虑工业物联网设备对每个边缘云的满意度外，工业物联网设备还参考其他工业物联网设备对每个边缘云的满意度。在这种情况下，可以由不同的工业物联网设备对每个边缘云进行适当的评估。问题模型包括自身满意度、其他设备评价和在边缘云计算的成成本。对工业物联网设备 $m \in \mathcal{M}$ ，那么它的效用函数 $F_m(P_n, Q_m, Q_{-m})$ 公式建模为公式 (5-2)。

$$F_m(P_n, Q_m, Q_{-m}) = U_m + \sum_{j=1}^M \phi_{mj} U_j - D_m - \zeta E_{mn}^T. \quad (5-2)$$

其中 U_m 采用对数函数，能够恰当地反映工业物联网设备与边缘云之间的满意程度的关系。

$$U_m = a \ln(1 + Q_m). \quad (5-3)$$

在公式 (5-3) 中， a 表示工业物联网设备的满意度系数， Q_m 表示工业物联网设

备 m 向边缘云申请的计算需求量。

在公式 (5-2) 中, $\sum_{j=1}^M \varphi_{mj} U_j$ 表示从其他工业物联网设备的社会交互信息, 由于工业物联网设备的网络特性, 工业物联网设备更倾向于考虑社会交互信息。特定地, 矩阵 φ_{mj} 表示工业物联网设备 m 从工业物联网设备 j 获得的影响系数, 取值范围在 $[0,1]$ 之间。 φ_{mj} 的值越高表示两个工业物联网设备的社交关系越近。此外, 本章考虑工业物联网设备 m 和工业物联网设备 j 彼此之间的影响是一样的, 即 $\varphi_{mj} = \varphi_{jm}$ 。

D_m 表示工业物联网设备 m 的支付开销, 公式表示为公式 (5-4)。

$$D_m = P_n \times Q_m. \quad (5-4)$$

E_{mn}^T 是工业物联网设备 m 的传输能耗, ς 是能耗的系数因子, 用于确定传输能耗的适当权重。

(2) 边缘云的效用函数。对于边缘云 n , 其效用函数等于利润减去自身的开销, 公式表示为式 (5-5)。

$$G_n(P_n, Q_m) = \sum_{m=1}^M (P_n \times Q_m - \xi E_n^C). \quad (5-5)$$

其中 E_n^C 是边缘云 n 的计算能耗, ξ 是能耗的系数因子, 用于确定计算能耗的适当权重。

5.3 基于完全信息的斯坦科尔伯格分析

在这种情况下, 本节假设所有工业物联网设备都愿意向其它设备报告自己的私有信息, 并从社交网络中获取正确的信息。例如, 在软件定义的网络场景中, 信息可以由控制器发送, 工业物联网设备可以通过控制器从附近的工业物联网设备获取信息。因此, 在完全信息的情况下, 本章将此问题表示为斯坦科尔伯格博弈对策。

5.3.1 完全社会交互信息的问题模型

在这种情况下, 每一个边缘云独立地宣布其价格, 而不需要与其他边缘云合作, 它可以通过分布式的方式做出决策。边缘云 n 的定价不仅要依赖工业物联网设备的反馈, 还要依赖其他边缘云的动作。因此, 问题模型公式可以建模为公式 (5-6)。

$$\begin{aligned}
& \max G_n(P_n^* | P_{-n}^*, Q^*), \\
& s.t. \quad Q^* \in [\underline{Q}, \bar{Q}], \\
& \quad P_n^* \in [\underline{P}, \bar{P}].
\end{aligned} \tag{5-6}$$

其中 $P_{-n}^* = \{P_1^*, P_2^*, \dots, P_{n-1}^*, P_{n+1}^*, \dots, P_N^*\}$ 表示除边缘云 n 以外的整个边缘云的最优价格。 $[\underline{Q}, \bar{Q}]$ 是计算需求集合 $Q^* = \{Q_1^*, Q_2^*, \dots, Q_M^*\}$ 的上限值和下限值, $[\underline{P}, \bar{P}]$ 是价格集合 P_n^* 的上限值和下限值。

基于以上边缘云的最优价格 P_n^* , 针对工业物联网设备 m , 基于完全社会信息的模型建立为公式 (5-7)。

$$\begin{aligned}
& \max F_m(Q_m | P_n^*, Q_{-m}), \\
& s.t. \quad T_{mm} \leq T_m^{\max}, \\
& \quad Q_m \in [\underline{Q}, \bar{Q}], \\
& \quad P_n^* \in [\underline{P}, \bar{P}],
\end{aligned} \tag{5-7}$$

其中, Q_m 定义为工业物联网设备 m 的计算资源需求策略, $[\underline{Q}, \bar{Q}]$ 是其上限值和下限值, $[\underline{P}, \bar{P}]$ 是价格集合 P_n^* 的上限值和下限值。 Q_{-m} 表示除工业物联网设备 m 之外的其他工业物联网设备的计算资源需求策略集。当边缘云设定的价格超出工业物联网设备的预期时, 没有设备愿意为这个边缘云的计算服务付费。相反, 当价格很低时, 没有边缘云希望提供计算服务。因此, 本章给出了价格约束 $[\underline{P}, \bar{P}]$, 以保证交易。

5.3.2 斯坦科尔伯格博弈纳什均衡分析

基于上述模型公式, 所有的边缘云和工业物联网设备都希望能够最大限度地利用它们的效用函数值。接下来, 本节将分析边缘云和工业物联网设备之间具有完整信息的斯坦科尔伯格博弈。

定义 5-1 假设 P_n^* 是边缘云的最优单价, Q_m^* 是工业物联网设备的最优计算需求量, 那么 (P_n^*, Q_m^*) 是最优的纳什均衡点满足以下条件:

$$\begin{aligned}
& F_m(P_n^*, Q_m^*, Q_{-m}^*) \geq F_m(P_n^*, Q_m, Q_{-m}^*) \&\& \\
& G_n(P_n^*, Q_m^*) \geq G_n(P_n, Q_m^*).
\end{aligned} \tag{5-8}$$

定理 5-1 考虑具有固定数量工业物联网设备的动态计算需求。对于某个工业物联网设备, 其效用函数满足公式 (5-2), 那么该工业物联网设备存在一个且唯一的纳什均衡点。

证明: 对任一工业物联网设备 m , 其效用函数是 $F_m(P_n, Q_m, Q_{-m})$, 具体函数

公式如公式 (5-9) 所示。

$$\begin{aligned}
 F_m(P_n, Q_m, Q_{-m}) &= U_m + \sum_{j=1}^M \varphi_{mj} U_j U_m - D_m - \zeta E_{mn}^T \\
 &= a \ln(1 + Q_m) \left(1 + \sum_{j=1}^M \varphi_{mj} U_j \right) - P_n \times Q_m - \zeta \rho_{mn} \frac{Q_m}{r_{mn}}.
 \end{aligned} \tag{5-9}$$

公式 (5-9) 的一阶导数为：

$$\begin{aligned}
 \frac{\partial(F_m(P_n, Q_m, Q_{-m}))}{\partial Q_m} &= \frac{a}{(1 + Q_m)} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j \right) - P_n - \frac{\zeta \rho_{mn}}{r_{mn}}.
 \end{aligned} \tag{5-10}$$

接下来，公式 (5-9) 的二阶导数为：

$$\begin{aligned}
 \frac{\partial(F_m^2(P_n, Q_m, Q_{-m}))}{\partial Q_m^2} &= -\frac{a}{(1 + Q_m)^2} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j \right).
 \end{aligned} \tag{5-11}$$

由于 $a > 0$ 和 $(1 + Q_m)^2 > 0$ ，效用函数的二阶导数为负。可以观察到工业物联网设备的效用函数是严格的凹函数，证明了纳什均衡的存在性。定理 5-1 证明完毕。

定理 5-2 考虑固定数量的工业物联网设备具有动态价格。对于每个边缘云，其效用函数满足公式 (5-5)，那么该边缘云存在唯一的纳什均衡点。

证明： 根据以上公式，边缘云 n 的效用函数如公式 (5-12) 所示。

$$\begin{aligned}
 G_n(P_n, Q_m^*) &= \sum_{m=1}^M P_n \times Q_{mn}^* - \xi E_n^C \\
 &= \sum_{m=1}^M P_n \times \left(\frac{ar_{mn}(1 + \sum_{j=1}^M \varphi_{mj} U_j)}{r_{mn} P_n + \zeta \rho_{mn}} - 1 \right) - \xi \sum_{m=1}^M \eta_n \frac{B_m}{C_{mn}} \\
 &= \sum_{m=1}^M \left(\frac{ar_{mn} P_n \left(1 + \sum_{j=1}^M \varphi_{mj} U_j \right)}{r_{mn} P_n + \zeta \rho_{mn}} - P_n \right) - \xi \sum_{m=1}^M \eta_n \frac{B_m}{C_{mn}}.
 \end{aligned} \tag{5-12}$$

那么，公式（5-12）的一阶导数为：

$$\begin{aligned} & \frac{\partial(G_n(P_n, Q_m^*))}{\partial P_n} \\ &= \sum_{m=1}^M \left[\frac{ar_{mn} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right) (r_{mn} P_n + \zeta \rho_{mn})}{(r_{mn} P_n + \zeta \rho_{mn})^2} - \frac{ar_{mn}^2 P_n \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right)}{(r_{mn} P_n + \zeta \rho_{mn})^2} - 1 \right] \\ &= \sum_{m=1}^M \left[\frac{ar_{mn} \zeta \rho_{mn} (1 + \sum_{j=1}^M \varphi_{mj} U_j)}{(r_{mn} P_n + \zeta \rho_{mn})^2} - 1 \right]. \end{aligned} \quad (5-13)$$

可以得到公式（5-12）的二阶导数为：

$$\begin{aligned} & \frac{\partial(G_n^2(P_n, Q_m^*))}{\partial P_n^2} \\ &= \sum_{m=1}^M \frac{-2(P_n r_{mn} + \zeta \rho_{mn}) r_{mn}^2 a \rho_{mn}}{(P_n r_{mn} + \zeta \rho_{mn})^4} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right) \\ &= \sum_{m=1}^M \frac{-2ar_{mn}^2 \zeta \rho_{mn}}{(P_n r_{mn} + \zeta \rho_{mn})^3} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right). \end{aligned} \quad (5-14)$$

由于 $a > 0, P_n > 0, \zeta \rho_{mn} > 0, r_{mn} > 0$ ，那么效用函数的二阶导数是

$$\frac{\partial(G_n^2(P_n, Q_m^*))}{\partial P_n^2} = \sum_{m=1}^M \frac{-2ar_{mn}^2 \zeta \rho_{mn}}{(P_n r_{mn} + \zeta \rho_{mn})^3} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right) < 0. \text{ 可以看出，边缘云的效用函}$$

数是严格的凹函数，证明了纳什均衡点存在。定理 5-2 证明完毕。

基于定理 5-1 和定理 5-2，纳什均衡点分别存在于第 I 阶段和第 II 阶段。最后，可以得出斯坦科尔伯格纳什均衡点存在且唯一的结论。

在这种情况下，假设博弈模型中的每个参与者都将自己的信息传递给一个中央控制器，以确保其他参与者能够相互获取策略。然而，这种情况会造成大量的传输成本。本节采用反向归纳法来解决这个问题。首先，通过推导得到计算资源的最优需求。然后，根据计算资源的最优需求，确定最优定价方案。具体流程如下。

根据公式（5-10），让一阶导数值等于零，

$$\begin{aligned} & \frac{\partial(F_m(P_n, Q_m, Q_m))}{\partial Q_m} \\ &= \frac{a}{1 + Q_m} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right) - P_n - \frac{\zeta \rho_{mn}}{r_{mn}} \\ &= 0. \end{aligned} \quad (5-15)$$

那么可以得到

$$Q_m^* = \begin{cases} \underline{Q}, & \Psi < \underline{Q}, \\ \Psi, & \underline{Q} < \Psi < \bar{Q}, \\ \bar{Q}, & \Psi > \bar{Q}. \end{cases} \quad (5-16)$$

其中 $\psi = \frac{ar_{mn} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right)}{r_{mn} P_n + \zeta \rho_{mn}} - 1$ 。那么令公式 (5-13) 的一阶导数等于 0,

$$\begin{aligned} & \frac{\partial G_n(P_n, Q_m)}{\partial P_n} \\ &= \sum_{m=1}^M \left(\frac{ar_{mn} \zeta \rho_{mn} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right)}{(r_{mn} P_n + \zeta \rho_{mn})} - 1 \right) \\ &= 0. \end{aligned} \quad (5-17)$$

那么可以得到

$$P_n^* = \begin{cases} \underline{P}, & \Phi < \underline{P}, \\ \Phi, & \underline{P} < \Phi < \bar{P}, \\ \bar{P}, & \Phi > \bar{P}. \end{cases} \quad (5-18)$$

其中 $\Phi = \sum_{m=1}^M \frac{1}{r_{mn}} \left(\sqrt{ar \zeta \rho_{mn} \left(1 + \sum_{j=1}^M \varphi_{mj} U_j\right)} - \zeta \rho_{mn} \right)$ 。

为了得到公式 (5-6) 和 (5-7) 博弈模型的纳什均衡, 提出了一种动态迭代算法。该动态迭代算法易于实现, 具有良好的收敛性能。算法 5-1 给出了纳什均衡动态迭代算法的伪代码。在本算法中, 根据导数方向计算下一代价格 $P_n(t+1)$ 和计算需求 $Q_m(\tau+1)$ 。随着迭代次数的增加, 数值逐渐逼近最优值。具体而言, 这个算法的计算复杂度为 $O(N^{t_{\max}} \times M^{\tau_{\max}})$, 其中 t_{\max} 和 τ_{\max} 分别是边缘云和工业物联网设备的最大迭代次数。此外, 系数 α 和 β 分别是边缘云和工业物联网设备的搜索步数大小。

5.4 基于不完全信息的斯坦科尔伯格分析

在没有全局管理的情况下, 工业物联网设备由于自身的自私行为而无法从对方获取个人信息。在这种情况下, 本节利用工业物联网设备的部分信息, 这些信息可以通过历史信息或机器学习方法获得。因此, 在信息不完全的情况下, 此问题被表示为贝叶斯斯坦科尔伯格博弈。

算法 5-1 基于动态迭代算法求解纳什均衡

输入：边缘云集合 $\mathcal{N} = \{1, 2, \dots, N\}$ ，工业物联网设备集合 $\mathcal{M} = \{1, 2, \dots, M\}$ ；

输出：最优价格 $P_n^*, (n \in \mathcal{N})$ ，最优计算需求 $Q_{mn}^*, (m \in \mathcal{M}, n \in \mathcal{N})$ ；

初始化：

边缘云价格 $P_n, (n \in \mathcal{N})$ ；

工业物联网设备需求 $Q_{mn}, (m \in \mathcal{M}, n \in \mathcal{N})$ ；

While $t < t_{\max}$ **do**

在每个时间槽 t ：

根据公式 $P_n(t+1) = P_n(t) + \alpha \frac{\partial G_n}{\partial P_n}$ ，更新每个边缘云 n ；

While $\tau < \tau_{\max}$ **do**

在每个时间槽 τ ：

根据公式 $Q_{mn}(t+1) = Q_{mn}(t) + \beta \frac{\partial F_m}{\partial Q_{mn}}$ ，更新每个设备计算需求 m ；

$\tau = \tau + 1$ ；

End while

$t = t + 1$ ；

End while

5.4.1 不完全社会交互信息的问题模型

在现实世界中，社交信息往往是不完整的，或者是被工业物联网设备所隐藏的。因此，此博弈被建模为贝叶斯斯坦科尔伯格博弈框架，并利用贝叶斯分析来获得预期的社会互动信息。具体地说，假设工业物联网设备的分布信息可以通过历史信息或机器学习方法来计算。为了简化社会交互信息，社会系数被认为 $\varepsilon = \varphi_{mj} > 0$ ，这部分表示为社会交互信息的期望。那么，公式（5-9）重新修改为

$$\begin{aligned}
 F'_m(P_n, Q_m, Q_{-m}) &= E(F_m(P_n, Q_m, Q_{-m})) \\
 &= U_m + \varepsilon E\left(\sum_{j=1}^M U_j\right) U_m - D_m - \zeta E_{mn}^T.
 \end{aligned} \tag{5-19}$$

不完全的社会交互信息包括产生不同输入度和输出度的工业物联网设备的社会信息结构。输入度表示工业物联网设备影响的其他工业物联网设备的数量，输出度表示影响该工业物联网设备的其他工业物联网设备的数量。每个工业物联网设备的社会交互信息的输出度定义为 $k \in \mathcal{K} = \{1, 2, \dots, K\}$ ，输入度定义为 $l \in \mathcal{K}$ 。一个输出度为 k 的工业物联网设备能够影响 k 个其他工业物联网设备，一个输入度为 l 的工业物联网设备代表被 l 个其他工业物联网设备影响。特定地，

$H: \mathcal{K} \rightarrow [0,1]$ 和 $I: \mathcal{K} \rightarrow [0,1]$ 是输出度和输入度的边际概率分布。一致性理论表明一个工业物联网设备的平均输出度等于平均输入度, 即 $\sum_{k=1}^K H(k)k = \sum_{l=1}^K I(l)l = k$, 其中 k 代表社会交互信息的平均水平。对一个工业物联网设备来说, 它选择邻居设备的概率分布为 $\bar{I}(l) = \frac{1}{k} I(l)l$ 。那么一个输出度为 k 的工业物联网设备 m 的期望效用函数公式为

$$F'_m(P_n, Q_m, Q_{-m}) = U_m + \varepsilon k \text{Ave}(U_{-m}) U_m - D_m - \zeta E_{mn}^T. \quad (5-20)$$

其中 $\text{Ave}(U_{-m})$ 表示工业物联网设备的邻居设备的平均影响力, 其公式建模为

$$\text{Ave}(U_{-m}) = \sum_l \bar{I}(l) \left(\sum_k H(k) a \ln(1 + Q(k, l)) \right). \quad (5-21)$$

由于输入度和输出度信息的分布是已知的, 所以 (5-5) 中的优化问题可以修改为

$$G'_n(P_n, Q_m) = \sum_l \left(\sum_k I(l) H(k) (P(k, l) \times Q(k, l) - \xi E_n^C) \right). \quad (5-22)$$

注解: 社会因子 ε 表示给定的社交交互信息, 该信息描述社交联系的效果, 它可以看作是一个估计值而不是实际值。

5.4.2 贝叶斯斯坦科尔伯格博弈纳什均衡分析

在这一小节中, 将利用反向归纳法对具有不完全社会交互信息的贝叶斯斯坦科尔伯格博弈进行分析。

1) 跟随者博弈。对于由工业物联网设备得到的不完全社会交互信息, 本节分析了跟随者博弈中的贝叶斯纳什均衡, 该均衡由以下定理描述。

定理 5-3 考虑固定数量的工业物联网设备具有动态价格。对于工业物联网设备, 其效用函数满足 (5-19)。在工业物联网设备中存在贝叶斯纳什均衡的唯一性。工业物联网设备的最优期望需求公式如下

$$Q_m^* = \frac{a(2-A)}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1. \quad (5-23)$$

其中 A 的表达式为

$$A = a \varepsilon k \omega \left(-\ln(a \varepsilon k) - \left(a \varepsilon k \left(\ln \left(\frac{\rho_{mn} + P_n r_{mn}}{r_{mn}} + \ln(a) \right) - 1 \right) / (a \varepsilon k) \right) \right). \quad (5-24)$$

在公式 (5-24) 中, 函数 $\omega(\cdot)$ 是一个特殊函数, 表示方程 $Y + \log(Y) = X$ 的解, 该函数的细节可以参考文献[156]。

证明：根据公式（5-20），工业物联网设备 m 的期望效用函数为

$$\begin{aligned} F'_m(P_n, Q_m, Q_{-m}) &= U_m + \varepsilon k \text{Ave}(U_{-m}) U_m - D_m - \zeta E_{mn}^T \\ &= a \ln(1 + Q_m) + \varepsilon k \text{Ave}(U_{-m}) a \ln(1 + Q_m) - P_n \times Q_m - \zeta \rho_{mn} \frac{Q_m}{r_{mn}}. \end{aligned} \quad (5-25)$$

那么公式（5-25）的一阶导数为

$$\begin{aligned} \frac{\partial(F'_m(P_n, Q_m, Q_{-m}))}{\partial Q_m} &= (1 + \varepsilon k \text{Ave}(U_{-m})) \frac{a}{1 + Q_m} - P_n - \frac{\zeta \rho_{mn}}{r_{mn}}. \end{aligned} \quad (5-26)$$

接下来，公式（5-25）的二阶导数为

$$\begin{aligned} \frac{\partial(F'_m(P_n, Q_m, Q_{-m}))}{\partial Q_m^2} &= -\frac{a}{(1 + Q_m)^2} (1 + \varepsilon k \text{Ave}(U_{-m})). \end{aligned} \quad (5-27)$$

显然，效用函数的二阶导数是负的。可以看出，工业物联网设备的效用函数是严格凹的，证明了贝叶斯纳什均衡的存在唯一性。定理 5-3 证明完毕。

为了得到跟随者博弈中贝叶斯纳什均衡的唯一一点，令公式（5-26）等于零，具体公式如下所示

$$(1 + \varepsilon k \text{Ave}(U_{-m})) \frac{a}{1 + Q_m} - P_n - \frac{\zeta \rho_{mn}}{r_{mn}} = 0. \quad (5-28)$$

那么可以得到最优解

$$\begin{aligned} Q_m^* &= \frac{a(1 + \varepsilon k \text{Ave}(U_{-m}))}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1 \\ &= \frac{a(1 + \varepsilon k E(U(k, l) | (k, l) \in \mathcal{K}))}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1. \end{aligned} \quad (5-29)$$

根据公式（5-21）可以得到

$$\begin{aligned}
& E(U(k', l') | (k', l') \in \mathcal{K}) \\
&= \sum_{l'}^K \bar{I}(l') \left(\sum_{k'}^K H(k') a \ln(1 + Q(k', l')) \right) \\
&= \sum_{l'}^K \bar{I}(l') \left(\sum_{k'}^K H(k') a \ln \left(\frac{a(1 + \varepsilon k \Gamma)}{P_n + \frac{\zeta \rho_{mn}}{r_n}} \right) \right) \quad (5-30) \\
&= \bar{a} \ln \left(1 + \frac{a(1 + \varepsilon k \Gamma)}{P_n + \frac{\zeta \rho_{mn}}{r_n}} - 1 \right).
\end{aligned}$$

其中 $\Gamma = E(U(k'', l'') | (k'', l'') \in \mathcal{K})$ 和 $\bar{a} = \sum_{l'}^K \bar{I}(l') \left(\sum_{k'}^K H(k') a \right)$, 由于

$$E(U(k', l') | (k', l') \in \mathcal{K}) = E(U(k'', l'') | (k'', l'') \in \mathcal{K}). \quad (5-31)$$

可以得到

$$\begin{aligned}
& E(U(k', l') | (k', l') \in \mathcal{K}) \\
&= \bar{a} \ln \left(\frac{a(1 + \varepsilon k E(U(k', l') | (k', l') \in \mathcal{K}))}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} \right). \quad (5-32)
\end{aligned}$$

接下来, 可以得到

$$\begin{aligned}
& \frac{1}{a} E(U(k', l') | (k', l') \in \mathcal{K}) \\
&= \ln a + \ln(1 + \varepsilon k E(U(k', l') | (k', l') \in \mathcal{K})) - \ln \left(P_n + \frac{\zeta \rho_{mn}}{r_{mn}} \right). \quad (5-33)
\end{aligned}$$

让 A 表示 $1 - \varepsilon k E(U(k', l') | (k', l') \in \mathcal{K})$, 那么公式 (5-33) 可以修改为

$$\frac{1}{a} \frac{1-A}{\varepsilon k} = \ln a + \ln A - \ln \left(P_n + \frac{\zeta \rho_{mn}}{r_{mn}} \right). \quad (5-34)$$

此外, 可以推导出

$$E(U(k', l') | (k', l') \in \mathcal{K}) + \frac{1-A}{\varepsilon k}. \quad (5-35)$$

综上所述, 跟随者的贝叶斯博弈的最优解如下

$$Q_m^* = \frac{a(1 + \varepsilon k \text{Ave}(U_{-m}))}{P_n + \frac{\rho_{mn}}{r_{mn}}} - 1$$

$$= \frac{a(2-A)}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1. \quad (5-36)$$

2) 领导者博弈。对于由工业物联网设备得到的不完全社会交互信息，接下来的定理分析了领导者博弈中的贝叶斯纳什均衡。

定理 5-4 类似于上面的分析，考虑固定数量工业物联网设备具有动态价格。对于边缘云，其效用函数满足公式 (5-22)，贝叶斯斯坦科尔伯格博弈中的最优价格如公式 (5-37) 所示。

$$P_n^* = a(2-A) - \frac{\rho_{mn}}{r_{mn}}. \quad (5-37)$$

证明：边缘云的期望效用函数表示为

$$G_n'(P_n, Q_m) = P(k, l) \times \left(\frac{a(2-A)}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1 \right) - \xi \eta_n \frac{B_m}{C_m}. \quad (5-38)$$

然后，边缘云的最优价格应满足其一阶导数为零的条件，并将其表示为

$$\frac{a(2-A)}{P_n + \frac{\zeta \rho_{mn}}{r_{mn}}} - 1 = 0. \quad (5-39)$$

接下来，可以得到最优价格的基于贝叶斯公式的统一形式

$$P_n^* = a(2-A) - \frac{\zeta \rho_{mn}}{r_{mn}}. \quad (5-40)$$

定理 5-4 证明完毕。

与算法 5-1 类似，本小节提出了一种动态迭代算法来实现 (5-20) 和 (5-22) 的近似贝叶斯纳什均衡，并在算法 5-2 中给出伪码。在本算法中，根据导数方向计算下一代价格 $P_n(t+1)$ 和计算需求 $Q_m(\tau+1)$ 。随着迭代次数的增加，数值逐渐逼近最优值。特定地，这个算法的计算复杂度为 $O(N^{t_{\max}} \times M^{\tau_{\max}})$ ，其中 t_{\max} 和 τ_{\max} 分别是边缘云和工业物联网设备的最大迭代次数。此外，系数 α 和 β 分别是边缘云和工业物联网设备的搜索步数大小。

综上所述，本章设计了一种基于两阶段斯坦科尔伯格博弈的计算卸载策略，该策略考虑了工业物联网设备中完全和不完全的社会交互信息。在完整的社会交互信息的情况下，工业物联网设备能够获得完整的信息。具体而言，通过后归纳法分析了斯坦科尔伯格博弈，证明了斯坦科尔伯格平衡的存在性和唯一性。在

另一种情况下，工业物联网设备的社会交互信息是不完全的，本章采用贝叶斯斯坦科尔伯格博弈分析，通过满意度因子的期望值来评价博弈均衡。

算法 5-2 基于动态迭代算法求解贝叶斯纳什均衡

输入： 边缘云集合 $\mathcal{N} = \{1, 2, \dots, N\}$ ，工业物联网设备集合 $\mathcal{M} = \{1, 2, \dots, M\}$ ；

输出： 最优价格 $P_n^*, (n \in \mathcal{N})$ ，最优计算需求 $Q_{mn}^*, (m \in \mathcal{M}, n \in \mathcal{N})$ ；

初始化：

边缘云价格 $P_n, (n \in \mathcal{N})$ ；

工业物联网设备需求 $Q_{mn}, (m \in \mathcal{M}, n \in \mathcal{N})$ ；

Begin:

While $t < t_{\max}$ **do**

在每个时间槽 t ：

根据公式 $P_n(t+1) = P_n(t) + \alpha \frac{\partial G'_n}{\partial P_n}$ ，更新每个边缘云 n ；

While $\tau < \tau_{\max}$ **do**

在每个时间槽 τ ：

根据公式 $Q_{mn}(t+1) = Q_{mn}(t) + \beta \frac{\partial F'_m}{\partial Q_{mn}}$ ，更新每个设备计算需求 m ；

$\tau = \tau + 1$ ；

End while

$t = t + 1$ ；

End while

End

5.5 仿真结果

在这一节中，提出的基于斯坦科尔伯格博弈的计算卸载策略的仿真实验在 MATLAB 2013 上进行了验证。在社会交互信息是否完整的两种情况下，对工业物联网设备中基于斯坦科尔伯格博弈的总需求和总效用与迭代次数的关系进行了仿真。除此之外，仿真实验还验证了边缘云的价格和效用值与迭代次数的关系。

5.5.1 仿真实验参数设置

在本节的仿真实验中，超密集网络场景由宏蜂窝基站边缘云、小蜂窝基站边缘云和工业物联网设备组成。此外，宏蜂窝基站边缘云和小蜂窝基站边缘云表示为服务提供者，并且它们的数量分别被设置为一。此外，两个边缘云都能够为不同数量的工业物联网设备提供计算服务，工业物联网设备随机分布在网络场景中。主要的工业物联网场景中的参数参见表 5-2。

表 5-2 参数设置

Table 5-2 Parameter settings

参数	设置值
工业物联网设备和宏蜂窝基站边缘云之间的信道带宽	10MHz
工业物联网设备和小蜂窝基站边缘云之间的信道带宽	1MHz
传输功率	[50mW,100mW]
高斯噪声变量	-100dBm
干扰因子	-100dBm
每个任务量大小	[10M,20M]
任务所需的 CPU 个数	[3,10]
每个任务的最大时延容忍	[0.05s,3s]
宏蜂窝基站边缘云的计算能力	5GHz
宏蜂窝基站边缘云的单元计算能耗	4J/Hz
小蜂窝基站边缘云的计算能力	4GHz
小蜂窝基站边缘云的单元计算能耗	3J/Hz
迭代次数 t	200
迭代次数 τ	200

5.5.2 具有完全社会交互信息的斯坦科尔伯格博弈仿真结果

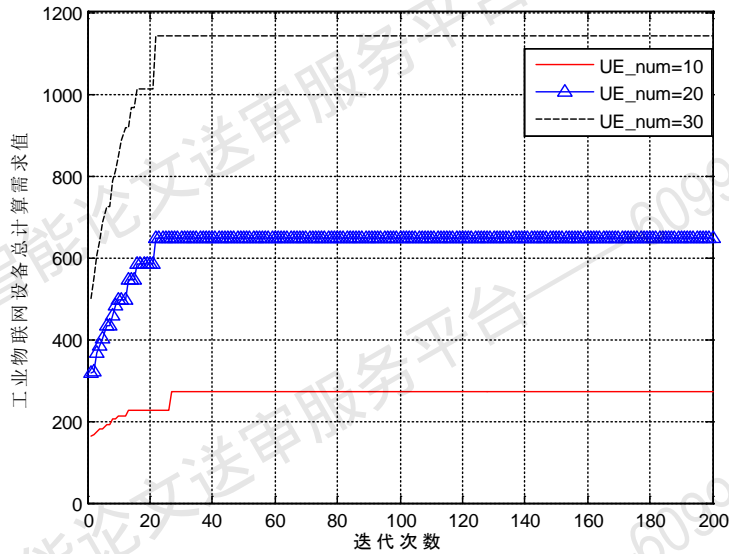
本节针对在工业物联网中提出的具有完全信息的斯坦科尔伯格博弈计算卸载策略进行了评估。此外，工业物联网设备的数量对效用函数的影响将在下面的实验中进行研究。

如图 5-3 所示，工业物联网设备的总需求值和总效用值在迭代次数增加中不断变化。具体而言，工业物联网装置的总需求值变化如图 5-3 (a) 所示，而工业物联网设备的总效用值变化如图 5-3 (b) 所示。随着迭代次数的增加，总需求值和总效用值迅速增加，收敛后保持稳定。图例 UE_num 表示工业物联网设备的数量，设置为{10、20、30}。通过仿真结果可以得出，工业物联网设备的数量越多，工业物联网设备的总需求值和总效用值就越高。因为更多的工业物联网设备能够产生更多的计算需求值，并试图在斯坦科尔伯格博弈中获得更多的效用值。最后证明，本章所提出的算法具有最终达到工业物联网设备的近似最优总需求值和总效用值的能力。

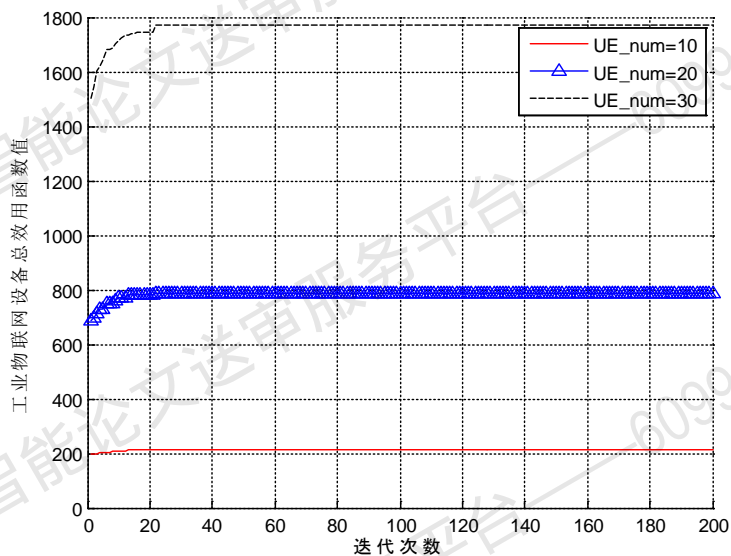
如图 5-4 所示，在工业物联网设备数量不同的情况下，给出了宏蜂窝基站边

缘云和小蜂窝基站边缘云相对于迭代次数的价格变化。具体来说, P_mbs 表示宏蜂窝基站边缘云的价格, 而 P_sbs 表示小蜂窝基站边缘云的价格。每个边缘云的原始价格分别设置为 0.1。首先, 可以很明显地看出, 在第一阶段, 价格随着迭代的变化而快速增长。然后, 在后一阶段, $UE_num = \{10, 20\}$ 情况下的曲线保持不变, $UE_num = \{30\}$ 情况下的曲线仍在增加。随着工业物联网设备数量的增加, 由于工业物联网设备对计算量的要求越来越高, 宏蜂窝基站边缘云和小蜂窝基站边缘云的价格也越来越高。

图 5-5 绘制了在工业物联网设备数量不同的情况下, 宏蜂窝基站边缘云和小蜂窝基站边缘云相对于迭代次数的效用值变化。与图 5-4 类似, G_mbs 表示宏蜂窝基站边缘云的效用值, G_sbs 表示小蜂窝基站边缘云的效用值。显然, 在第一阶段, 效用值随着迭代次数的变化而迅速增加。在后一阶段, $UE_num = \{10, 20\}$ 情况下的曲线保持不变, $UE_num = \{30\}$ 情况下的曲线在暂态稳定后仍在增加。根据图 5-4 和图 5-5, 随着 UE_num 的增加, 价格和效用值都迅速增加。此外, 由于价格上涨, 更多的工业物联网设备使得宏蜂窝基站边缘云和小蜂窝基站边缘云具有更高的效用值。



(a) 工业物联网设备中基于斯坦科尔伯格博弈的总需求值变化



(b) 工业物联网设备中基于斯坦科尔伯格博弈的总效用值变化

图 5-3 工业物联网设备中基于斯坦科尔伯格博弈的总需求值和总效用值变化

Fig. 5-3 Total requirement and total utility versus iterations based on Stackelberg game in IIoT devices

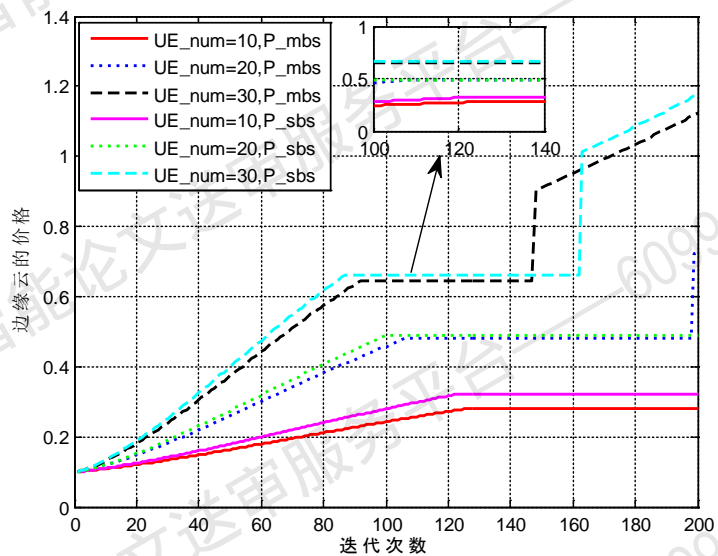


图 5-4 不同边缘云中基于斯坦科尔伯格博弈的价格变化

Fig. 5-4. Price versus iterations based on Stackelberg game in different edge clouds

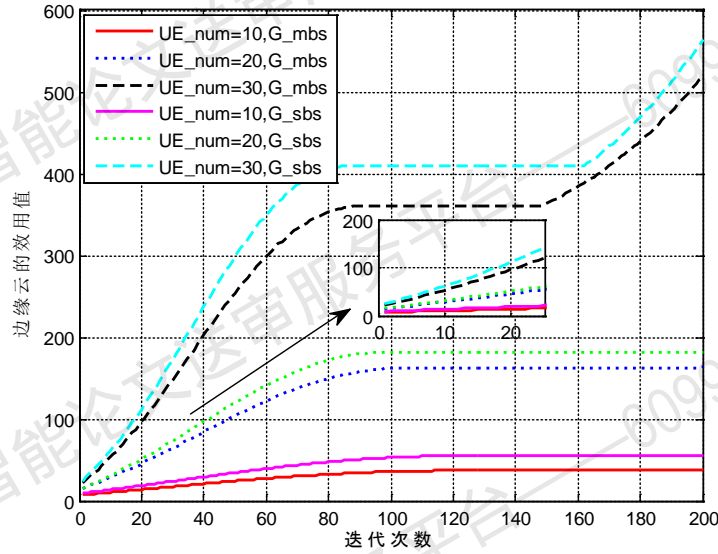


图 5-5 不同边缘云中基于斯坦科尔伯格博弈的效用值变化

Fig. 5-5. Utility versus iterations based on Stackelberg game in different edge clouds

5.5.3 具有不完全社会交互信息的贝叶斯斯坦科尔伯格博弈仿真结果

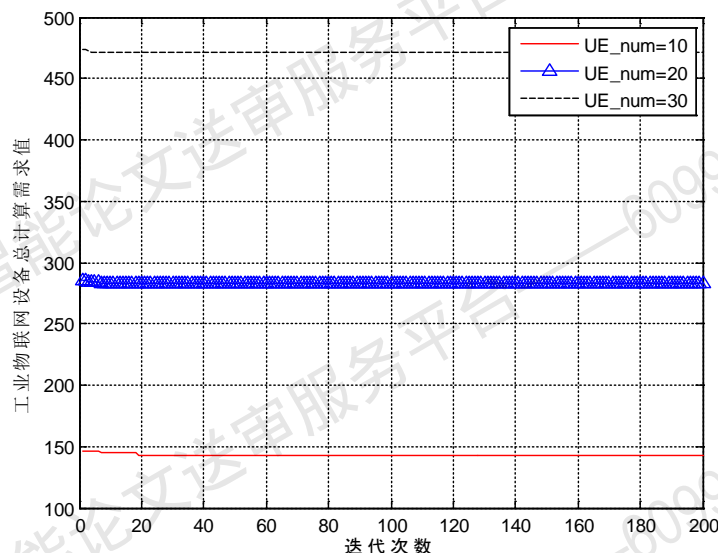
在这一小节中,对本章提出的不完全信息的贝叶斯斯坦科尔伯格博弈计算卸载策略进行了仿真实验验证,并在接下来的实验中分析工业物联网设备数量对性能的影响。

如图 5-6 所示,给出了基于贝叶斯斯坦科尔伯格博弈的工业物联网设备的总需求值和总效用值变化。具体而言,总需求值的变化如图 5-6 (a) 所示,总效用值的变化如图 5-6 (b) 所示。与上面的实验类似, UE_num 被设置为{10, 20, 30}。更多的工业物联网设备能够产生更高的总需求值,而当 $UE_num = \{20\}$ 时,工业物联网设备的总效用值最高。最后,仿真实验证明该算法最终可以收敛到工业物联网设备的近似最优总需求值和总效用值。

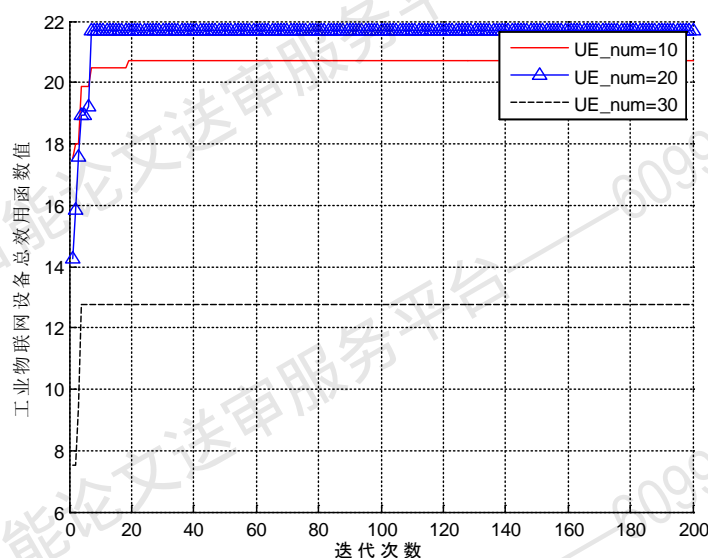
在不完全社会交互信息的情况下,图 5-7 展示出了宏蜂窝基站边缘云和小蜂窝基站边缘云相对于迭代次数的价格变化。具体来说, P_mbs 表示宏蜂窝基站边缘云的价格, P_sbs 表示小蜂窝基站边缘云的价格。与图 5-4 类似,两个边缘云的初始价格都设置为 0.1。可以明显地发现,在整个过程中,价格随着迭代次数的增加而快速增长。此外,由于工业物联网设备的计算需求量增加,工业物联网设备的数量增加也促进了两种边缘云最终价格的提高。

图 5-8 描绘了在工业物联网设备的不同数量的情况下,宏蜂窝基站边缘云和小蜂窝基站边缘云相对于迭代次数的效用值变化。与图 5-5 类似, G_mbs 和 P_sbs 分别表示宏蜂窝基站边缘云和小蜂窝基站边缘云的效用值。可以得到宏

蜂窝基站边缘云和小蜂窝基站边缘云的效用值随着迭代次数的变化而增加。此外,随着设备个数的增加,宏蜂窝基站边缘云和小蜂窝基站边缘云在相同迭代次数下的效用值也随之增加。小蜂窝基站边缘云的效用值略高于宏蜂窝基站边缘云的效用值。



(a) 工业物联网设备中基于贝叶斯坦科尔伯格博弈的总需求值变化



(b) 工业物联网设备中基于贝叶斯坦科尔伯格博弈的总效用值变化

图 5-6 工业物联网设备中基于贝叶斯坦科尔伯格博弈的总需求值和总效用值变化

Fig. 5-6. Total requirement and total utility versus iterations based on Bayesian Stackelberg game in IIoT devices

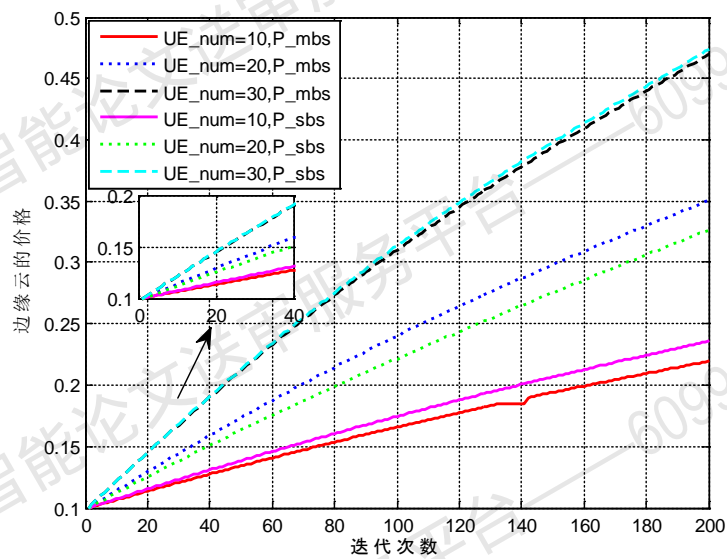


图 5-7 不同边缘云中基于贝叶斯斯坦科尔伯格博弈的价格变化

Fig. 5-7. Price versus iterations based on Bayesian Stackelberg game in different edge clouds

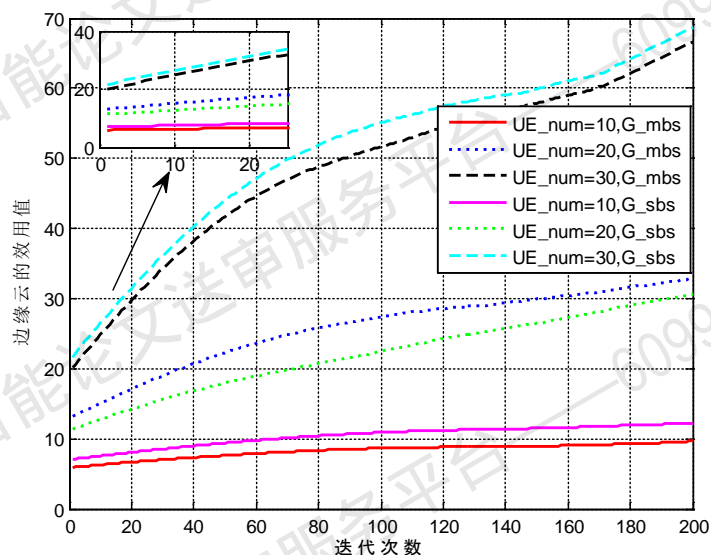
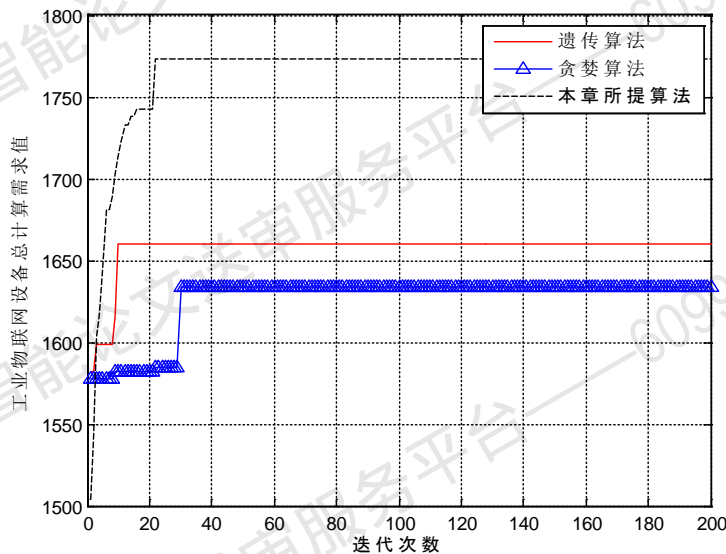


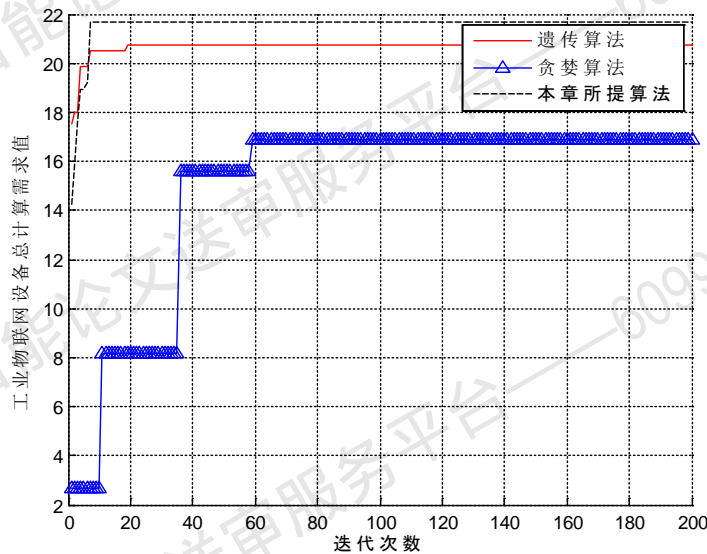
图 5-8 不同边缘云中基于贝叶斯斯坦科尔伯格博弈的效用值变化

Fig. 5-8. Utility versus iterations based on Bayesian Stackelberg game in different edge clouds

5.5.4 与其他算法仿真结果比较



(a) 工业物联网设备中基于斯坦科尔伯格博弈的总需求值比较



(b) 工业物联网设备中基于贝叶斯斯坦科尔伯格博弈的总需求值比较

图 5-9 工业物联网设备在两种情况下的总需求值比较

Fig. 5-9 Comparison of computation requirement for IIoT devices in two cases

为了验证所提方案的性能，本小节使用遗传算法和贪婪算法进行了比较。算法的粒子数为 10，最大迭代次数为 200。对于遗传算法的参数两个重要的参数， P_c 代表模拟染色体遗传模式的交叉概率，取 0.8。 P_m 是模拟基因突变的突变概率，设为 0.1。如图 5-9(a)所示，在具有完全的社会交互信息的情况下，本章所提算法增长迅速，在整个迭代过程中表现良好。最后，所提出的方案为工业物联网设备提供了最高的效用值。从图 5-9(b)可以看出，在社会交互信息不确定的情况下，

与其它两种对比算法相比，本章提出的方案也表现出不错的性能。

5.6 本章小结

在本章工作中，提出了一个基于斯坦科尔伯格博弈的工业物联网计算卸载策略。在第一阶段，边缘云确定计算服务的适当价格。然后，工业物联网设备在第二阶段根据边缘云的价格来决定它们的计算需求。考虑到其他工业物联网设备的社会交互信息，提出了工业物联网设备和边缘云的效用函数模型。此外，本章还分析了信息完整与否的主从对策，证明了两种情形下斯坦科尔伯格均衡的存在性和唯一性。仿真结果验证了本章提出的方案能够在工业物联网中提供合适的解决方案。

第6章 面向智能化的多用户计算卸载服务策略

6.1 引言

近年来,随着传统蜂窝网络中移动通信量的爆炸式增长,无线网络的传输容量面临着前所未有的挑战。思科预测^[157],从2017年到2022年移动数据流量将以46%的增长率增长,到2022年将达到每月77 EB。与此同时,新一代5G网络^[158]中的新业务场景不断增加,对延迟、能效和其他性能提出了更高的要求。为了应对日益严峻的挑战,超密集网络^[159]赋予5G网络巨大的接入能力,其基本思想是尽可能的部署更多的接入节点靠近终端用户。该网络架构由多种宏蜂窝基站和小蜂窝基站组成。依赖于超密集蜂窝网络中小蜂窝的密集部署,用户设备可以获得提供巨大的访问能力。

为了满足超密集网络对计算服务的需求,移动边缘计算^[160-161]被认为是将计算能力从网络中心转移到边缘的有效技术。与传统云计算相比,边缘计算既能够有效缩短用户设备与数据中心之间的数据传输距离,又能避免大量数据交换造成的网络拥塞,提高网络回程能力。然而,边缘计算并不拥有足够的计算能力和存储能力。同时,多个用户设备彼此之间通信和计算资源存在竞争,导致服务质量降低,传输时延延长等问题。因此,在多用户场景中,如何在通信和计算资源的约束下智能有效地卸载多个任务,满足用户设备的服务质量是一个重大的挑战。

移动边缘计算中的计算卸载问题通常被建模为混合整数规划问题,但是在边缘云系统中建模往往比较复杂,其原因是:一方面,卸载到边缘计算系统的任务具有不同的特性、延迟要求、计算要求和能耗等,要准确地模拟这个问题是不可能的。另一方面,在线任务是动态变化的,这就要求边缘云做出适当的在线决策。为了解决这一问题,相关学者采用了许多方法,如凸优化方法^[162]、博弈论^[163]和启发式算法^[164]。然而,这些问题很大程度上依赖于问题模型,并且不适合在高度变化的场景中做出实时卸载决策。

本章将讨论在一个新的场景,即超密集网络中,考虑面向智能化的多用户计算卸载服务问题。具体而言,考虑计算卸载中的通信资源、计算资源和缓存资源的框架,采用深度强化学习方法^[165-166]来帮助管理任务卸载。实验结果表明,该框架能有效地提高边缘计算的回报值,快速收敛到满意解。本章的主要工作概括如下:

(1) 研究了超密集网络中面向智能化的多用户计算卸载服务问题。其中,边缘云表示为服务提供者,用户设备表示为请求者。同时借助SDWN控制器全

局管控网络，为用户设备计算卸载服务的做出合理决策；

(2) 基于计算卸载中的通信资源、计算资源和缓存资源的框架，提出了基于深度 Q 学习算法的计算卸载策略。该策略将卸载决策和缓存决策建模为动作行为，在具有高维行为空间的大型网络中具有计算可行性和高效性；

(3) 仿真实验表明，该算法可以快速收敛到满意解。此外，还分析了不同参数对该策略的影响。

6.2 系统模型

6.2.1 超密集蜂窝网络架构

在大规模 MIMO 天线和毫米波通信技术的推动下，小蜂窝基站的部署在 5G 无线网络中应运而生。然而，考虑到城市环境中的成本和地理部署挑战，很难通过宽带互联网或光链路转发每个小蜂窝基站的回程流量。此外，由于无线传输距离的限制，小蜂窝基站通常不能为整个远程用户提供无线回程服务。因此，在 5G 超密集网络场景下，为了解决社区移动用户频繁切换的问题，宏蜂窝基站和小蜂窝基站的相互协调能够保证移动用户任务的传输。

如图 6-1 所示，超密集网络由三部分组成：宏蜂窝基站、小蜂窝基站和用户设备。具体来说，宏蜂窝基站和小蜂窝基站分别与边缘服务器连接，称为宏蜂窝基站边缘云和小蜂窝基站边缘云。虽然宏蜂窝基站边缘云与用户设备的距离更远，但它可以提供比小蜂窝基站边缘云更强大的计算能力和更大的接入带宽。相比之下，小蜂窝基站边缘云更接近用户设备，可以提供更快的计算服务。用户设备能够选择不同的卸载计算方案，即选择宏蜂窝基站边缘云或小蜂窝基站边缘云卸载任务。

从图 6-2 中，用户设备与边缘云之间的计算卸载服务包括三个步骤。首先，选择用户设备中的任务卸载到小蜂窝基站边缘云或基站边缘云。其次，边缘云选择卸载任务的执行决策。如果任务尚未缓存在边缘云中，则将计算该任务。否则，将直接返回缓存的结果。最后，边缘云将执行的结果返回给用户设备。在介绍具体系统模型之前，先将主要的符号总结在表 6-1 中。

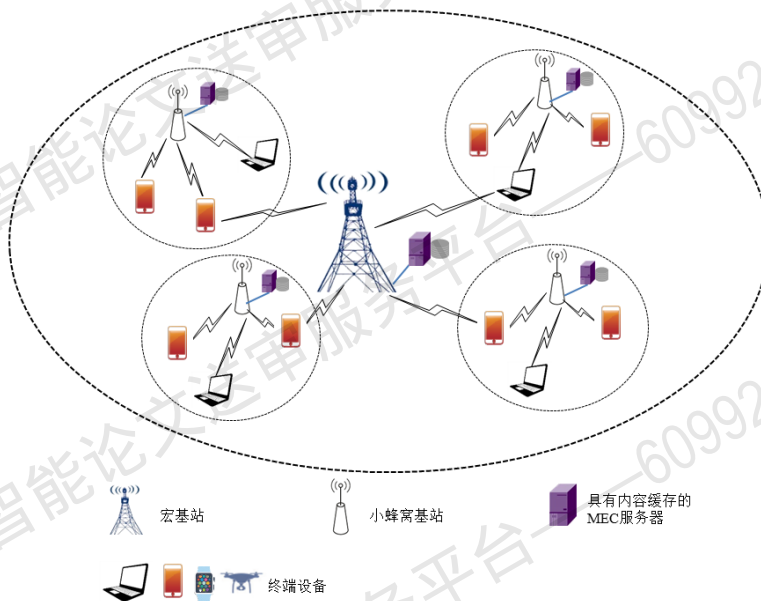


图 6-1 超密集蜂窝网络

Fig. 6-1 Architecture of Ultra-dense networks

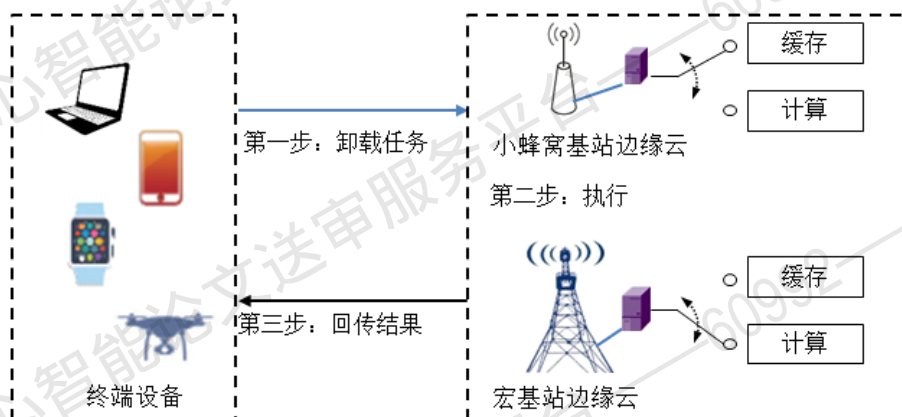


图 6-2 超密集蜂窝网络中的计算卸载和缓存策略

Fig. 6-2 Computation offloading strategy with content cache in Ultra-dense networks

6.2.2 通信模型

本章研究的计算卸载问题基于一种具有一个宏蜂窝基站和多个随机分布的小蜂窝基站的超密集网络架构。在此架构中， $\mathcal{N} = \{1, 2, \dots, N\}$ 记作小蜂窝基站集合， $\mathcal{M} = \{1, 2, \dots, M\}$ 表示用户设备集合。此外，每个小蜂窝基站边缘云都能够通过无线链路与宏蜂窝基站边缘云通信。每个用户设备中的任务可以卸载到宏蜂窝基站边缘云或小蜂窝基站边缘云，并且每个用户设备最多同时可以与一种边缘云通信。假设在每个用户设备 m 中有计算任务 $J_m = \{A_m, A'_m, B_m, T_m^{\max}\}$ 。其中， A_m 和 A'_m 分别表示任务 J_m 计算前后的数据量大小。 B_m 表示任务计算的 CPU 核数， T_m^{\max}

表示每个任务 J_m 的最大时延容忍。

表 6-1 本章的符号定义

Table 6-1 Description of Notations

符号	定义
$\mathcal{N} = \{1, 2, \dots, N\}$	边缘云集合
$\mathcal{M} = \{1, 2, \dots, M\}$	用户设备集合
J_m	用户设备的任务
w_{m0}	用户设备 m 和宏蜂窝基站边缘云之间的信道带宽
w_{mn}	用户设备 m 和小蜂窝基站边缘云 n 之间的信道带宽
p_{m0}	用户设备 m 到宏蜂窝基站边缘云的传输功率
p_{mn}	用户设备 m 到小蜂窝基站边缘云 n 的传输功率
h_{m0}	用户设备 m 和宏蜂窝基站边缘云之间之间的信道增益
h_{mn}	用户设备 m 和小蜂窝基站边缘云 n 的信道增益
α_{m0}	终端用户 m 的接入费用，由宏蜂窝基站边缘云收取
α_{mn}	终端用户 m 的接入费用，由小蜂窝基站边缘云 n 收取
β_{m0}	频谱的使用费用，由宏蜂窝基站边缘云支付
β_{mn}	频谱的使用费用，由小蜂窝基站边缘云 n 支付
ϕ_{m0}	终端用户 m 的计算费用，由宏蜂窝基站边缘云收取
ϕ_{mn}	终端用户 m 的计算费用，由小蜂窝基站边缘云 n 收取
φ_{m0}	计算成本，由宏蜂窝基站边缘云支付
φ_{mn}	计算成本，由小蜂窝基站边缘云 n 支付
γ_{m0}	宏蜂窝基站边缘云支付的回传成本
γ_{mn}	小蜂窝基站边缘云 n 支付的回传成本
ψ_{m0}	存储费用，由宏蜂窝基站边缘云收取
ψ_{mn}	存储费用，由小蜂窝基站边缘云 n 收取

(1) 任务卸载到宏蜂窝基站边缘云。假设 w_{m0} 代表用户设备 m 和宏蜂窝基站边缘云之间的信道带宽， p_{m0} 表示用户设备 m 到宏蜂窝基站边缘云的传输功率。 σ^2 表示高斯白噪声常量， h_{m0} 表示用户设备 m 和宏蜂窝基站边缘云之间的信

道增益。 $k_{m0} \in \{0,1\}$ 表示用户设备 m 和宏蜂窝基站边缘云的影响系数。那么用户设备 m 到宏蜂窝基站边缘云这条链路的传输速率如公式 (6-1) 所示。

$$r_{m0} = w_{m0} \log_2 \left(1 + \frac{p_{m0} |h_{m0}|^2}{\sigma^2 + \sum_{m=1}^M k_{m0} p_0 |h_{m0}|^2} \right). \quad (6-1)$$

在此模型中, 假设 α_{m0} 表示从用户设备 m 收到的接入费用, 由宏蜂窝基站边缘云收费。 β_{m0} 表示频谱的使用花费, 由宏蜂窝基站边缘云支付。那么通信收益可以建模为 $R_{m0}^{comm} = \alpha_{m0} r_{m0} - \beta_{m0} w_{m0}$ 。具体而言, $\alpha_{m0} r_{m0}$ 表示宏蜂窝基站边缘云从用户设备 m 的收入, $\beta_{m0} w_{m0}$ 表示宏蜂窝基站边缘云为带宽使用的花费。

(2) 任务卸载到小蜂窝基站边缘云。假设 w_{mn} 代表用户设备 m 和小蜂窝基站边缘云 n 之间的信道带宽, p_{mn} 表示用户设备 m 到小蜂窝基站边缘云 n 的传输功率。 σ^2 表示高斯白噪声常量, h_{mn} 表示用户设备 m 和边缘云 n 之间的信道增益。 $k_{mn} \in \{0,1\}$ 表示用户设备 m 和小蜂窝基站边缘云 n 的影响系数。那么用户设备 m 到小蜂窝基站边缘云 n 这条链路的传输速率为

$$r_{mn} = w_{mn} \log_2 \left(1 + \frac{p_{mn} |h_{mn}|^2}{\sigma^2 + \sum_{m=1}^M k_{mn} p_n |h_{mn}|^2} \right). \quad (6-2)$$

此外, 与上一模型类似, 假设 α_{mn} 表示从用户设备 m 收到的接入费用, 由小蜂窝基站边缘云 n 收费。 β_{mn} 表示频谱的使用花费, 由小蜂窝基站边缘云 n 支付。那么该通信收益可以建模为 $R_{mn}^{comm} = \alpha_{mn} r_{mn} - \beta_{mn} w_{mn}$ 。具体而言, $\alpha_{mn} r_{mn}$ 表示小蜂窝基站边缘云 n 从用户设备 m 的收入, $\beta_{mn} w_{mn}$ 表示小蜂窝基站边缘云 n 为带宽使用的花费。

6.2.3 计算模型

(1) 宏蜂窝基站边缘云中的计算模型。宏蜂窝基站边缘云的计算时延建模为 $T_{m0}^c = \frac{B_m}{C_{m0}}$, 其中 C_{m0} 表示宏蜂窝基站边缘云的计算能力。那么任务 m 在宏蜂窝基站边缘云中的计算速率建模为

$$q_{m0} = \frac{A_m}{B_m / C_{m0}} = \frac{A_m C_{m0}}{B_m}. \quad (6-3)$$

宏蜂窝基站边缘云中的计算能耗表示为 $e_{m0} = \varpi_{m0} T_{m0}^c$, 其中 ϖ_{m0} 表示宏蜂窝基站边

缘云中单位时间内消耗的能耗速率。

假设 ϕ_{m0} 表示用户设备 m 支付的计算费用，由宏蜂窝基站边缘云收费。同时，宏蜂窝基站边缘云所需支付的计算费用定义为 φ_{m0} 。那么计算收益可以建模为 $R_{m0}^{comp} = \phi_{m0}q_{m0} - \varphi_{m0}e_{m0}$ 。具体来说， $\phi_{m0}q_{m0}$ 表示宏蜂窝基站边缘云从用户设备 m 中的收入， $\varphi_{m0}e_{m0}$ 表示宏蜂窝基站边缘云中服务器使用的花费。

(2) 小蜂窝基站边缘云 n 中的计算模型。小蜂窝基站边缘云 n 的计算时延建模为 $T_{mn}^c = \frac{B_m}{C_{mn}}$ ，其中 C_{mn} 表示小蜂窝基站边缘云 n 的计算能力。那么任务 m 在小蜂窝基站边缘云 n 中的计算速率建模为

$$q_{mn} = \frac{A_m}{B_m / C_{mn}} = \frac{A_m C_{mn}}{B_m}. \quad (6-4)$$

小蜂窝基站边缘云 n 中的计算能耗表示为 $e_{mn} = \varpi_{mn} T_{mn}^c$ ，其中 ϖ_{mn} 表示小蜂窝基站边缘云 n 中单位时间内消耗的能耗速率。

假设 ϕ_{mn} 表示用户设备 m 支付的计算费用，由小蜂窝基站边缘云 n 收费。同时，小蜂窝基站边缘云 n 所需支付的计算费用定义为 φ_{mn} 。那么计算收益可以建模为 $R_{mn}^{comp} = \phi_{mn}q_{mn} - \varphi_{mn}e_{mn}$ 。具体来说， $\phi_{mn}q_{mn}$ 表示小蜂窝基站边缘云 n 从用户设备 m 中的收入， $\varphi_{mn}e_{mn}$ 表示小蜂窝基站边缘云 n 中服务器使用的花费。

6.2.4 缓存模型

在此无线网络场景中，本章考虑在边缘云中缓存 D 个内容文件。缓存策略由二进制数值确定，1 表示该内容缓存在边缘云中，0 表示该内容没有缓存在边缘云中。

在此模型中，内容流行度定义为 $\mathcal{G} = \{g_1, g_2, \dots, g_D\}$ ，其中 D 表示内容文件的最大类型个数。每个用户设备以概率 g_d 请求内容文件 d 。通常来说， \mathcal{G} 服从 Zipf 分布，公式建模如下：

$$g_d = \frac{1/d^\varepsilon}{\sum_{d=1}^D 1/d^\varepsilon}. \quad (6-5)$$

其中内容流行度由参数 ε 控制，范围在 $[0.5, 1.5]$ 之间。那么通过缓存内容 A'_m 节省回传带宽的效益可以建模为

$$l_{A'_m} = \frac{g_{A'_m} A'_m}{T_{A'_m}}. \quad (6-6)$$

其中 A'_m 是下载请求缓存内容文件的持续时间。在本章中，假设内容缓存的费用

已经提前设定。回传的成本费用记作 γ_{m0} ，由宏蜂窝基站边缘云支付。此外， ψ_{m0} 表示内容文件 A'_m 缓存在宏蜂窝基站边缘云的存储费用，由宏蜂窝基站边缘云定价并收取费用。那么，宏蜂窝基站的缓存回报值可以建模为 $R_{m0}^{cache} = \psi_{m0} l_{A'_m} - \gamma_{m0} A'_m$ 。具体而言， $\psi_{m0} l_{A'_m}$ 表示宏蜂窝基站边缘云从用户设备 m 收取到的收入， $\gamma_{m0} A'_m$ 表示宏蜂窝基站边缘云为回程带宽支付的使用费用。

类似地，将回传的成本费用记作 γ_{mn} ，由宏蜂窝基站边缘云支付， ψ_{mn} 表示内容文件 A'_m 缓存在小蜂窝基站边缘云 n 的存储费用，由小蜂窝基站边缘云 n 定价并收取费用。那么，小蜂窝基站边缘云 n 的缓存回报值可以建模为 $R_{mn}^{cache} = \psi_{mn} l_{A'_m} - \gamma_{mn} A'_m$ 。具体而言， $\psi_{mn} l_{A'_m}$ 表示小蜂窝基站边缘云 n 从用户设备 m 收取到的收入， $\gamma_{mn} A'_m$ 表示小蜂窝基站边缘云 n 为回程带宽支付的使用费用。

6.3 深度强化学习算法

6.3.1 强化学习

强化学习算法是机器学习的一个重要分支，它不同于一般的有监督学习和无监督学习。该算法模拟了智能体与环境的交互作用，智能体可以从环境中获取观察（状态）并采取行动。然后，环境将回报给智能体。具体而言，强化学习主要用于解决一个多步骤的决策问题，它试图通过在不断变化的情况下的多步骤适当的决策来达到一个目标。与其他机器学习算法不同，强化学习不需要从经验样本中学习。相反，它能够从尝试的行为中获得反馈。强化学习包括实验状态、行为、回报函数和状态转换概率四个部分。综上所述，强化学习算法是解决现实问题的一种无先验知识的有效方法。

Q 学习算法是强化学习中一种应用广泛的算法。在与环境交互的过程中，Q 学习算法试图通过不断的尝试和误差来寻找最优行为。最优行为不仅关系到即时回报，而且关系到后续步骤的回报。Q 学习算法的决策过程是基于马尔科夫决策过程的，它可以用五元组来表示： $\{s_i, a_i, P(s_i, a_i, s_{i+1}), R(s_i, a_i), Q(s_i, a_i)\}$ 。其中， s_i 表示状态空间， a_i 称为操作空间。此外，概率 $P(s_i, a_i, s_{i+1})$ 有助于选择操作 a_i 将状态 s_i 传输到下一个状态 s_{i+1} 。 $R(s_i, a_i)$ 表示处于 s_i 状态的系统选择操作 a_i 时的立即回报值。 $Q(s_i, a_i)$ 是为状态 s_i 选择操作 a_i 时的累积回报值。

假设状态在第 i 步表示为 s_i ，状态的累积回报函数记作 $V^\pi(s_i)$ ，那么它的函数公式为

$$V^{\pi}(s_i) = R_i + \lambda R_{i+1} + \lambda^2 R_{i+2} + \dots \quad (6-7)$$

其中 R_i 是第 i 步的回报值, $\lambda (0 < \lambda < 1)$ 表示折扣因子, 用于评估历史状态回报值对未来回报的影响。 λ 取值越大, 表示训练的智能体与历史经验越相关, 否则, 与立即回报越相关。在 Q 学习算法中, Q 值是状态和动作的评估, 表示为

$$Q(s_i, a_i) = R_i + \lambda V^{\pi}(s_{i+1}). \quad (6-8)$$

根据以上公式, $Q_{i+1}(s_i, a_i)$ 的更新公式见式 (6-9), 其中, 学习率 $\eta (\eta \in (0, 1))$ 控制学习速率。

$$Q_{i+1}(s_i, a_i) = (1 - \eta)Q_i(s_i, a_i) + \eta[R_i + \lambda \max Q_i(s_{i+1}, a_{i+1})]. \quad (6-9)$$

6.3.2 深度 Q 学习算法

算法 6-1 深度 Q 学习算法

输入: 状态 s , 动作 a ;

输出: $Q(s, a)$;

初始化:

根据参数 θ 初始化深度 Q 网络;

Begin:

For $i < T$ do

If 概率 $P < \delta$ then

选择动作 a_i ;

Else

$a_i = \arg \max Q(s, a; \theta)$;

Endif

执行动作 a_i , 获得回报函数值 R_i , 下一代状态 s_{i+1} ;

计算目标 Q 值

$y(s, a, s'; \hat{\theta}) = R + \lambda \max Q(s', a'; \hat{\theta})$;

根据式 (6-10) 最小化损失值 $L(\theta)$ 更新神经网络;

End for

End

深度 Q 网络的思想是利用前馈人工神经网络来逼近 Q 值函数 $Q(s, a; \theta)$ 。与一次计算一个状态动作对 (s, a) 的 Q 值不同, 深度 Q 网络中的 Q 值可以与紧密的网络结构同时计算。这个 Q 网络的输入层是状态 s , 输出层是 Q 值, 对应于在状态 s 执行的操作 a 。参数 θ 代表神经网络的权值, Q 网络通过最小化损失函数更新 θ 来训练:

$$L(\theta) = E[(y(s, a, s'; \hat{\theta}) - Q(s, a; \theta))^2]. \quad (6-10)$$

其中，目标函数 $y(s, a, s'; \hat{\theta}) = R + \lambda \max Q(s', a'; \hat{\theta})$ 根据参数 $\hat{\theta}$ 的变化而更新。

6.4 基于深度强化学习算法的计算卸载和缓存策略

在超密集网络场景下，上行信道条件和计算能力都在动态变化，导致系统状态和行为空间很大，传统的方法很难找到最优解。相比之下，深度强化学习算法不需要捕获良好的模型或先前的信息，它能够从环境中自适应地改进策略。因此，本节引入深度 Q 学习算法来有效地找到整个动作空间中的最佳动作，如图 6-3 所示。具体来说，根据卸载的任务是否缓存在边缘云中，提出了不同的策略。一方面，如果卸载的任务缓存在边缘云上，边缘云应该考虑缓存模型以节省计算延迟。另一方面，如果卸载任务未缓存在边缘云上，则边缘云将直接计算此任务。接下来，本节将详细介绍深度 Q 学习算法模型，包括深度 Q 学习算法模型中的状态、行为和回报函数的设计。

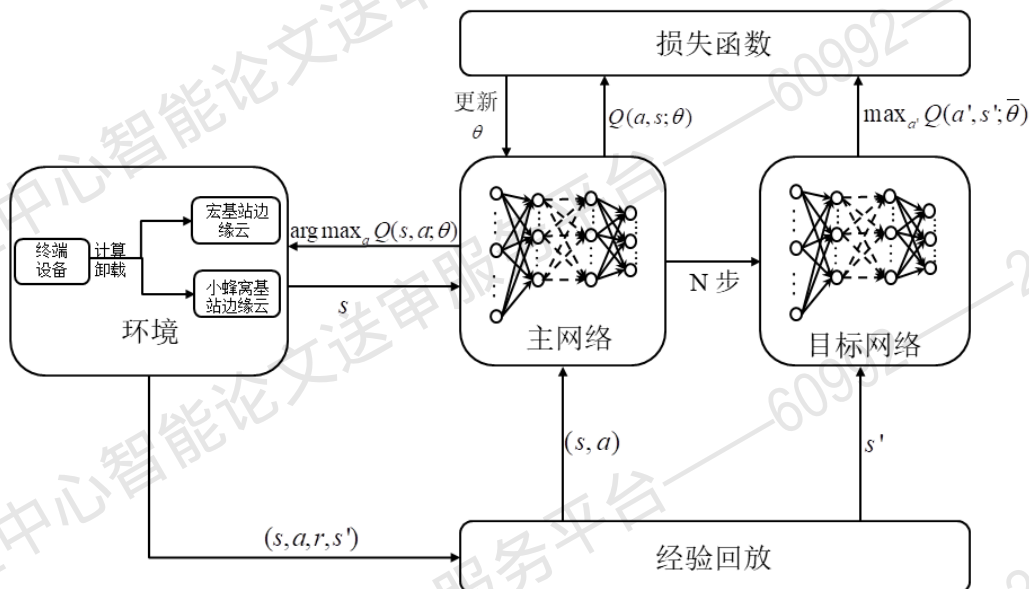


图 6-3 基于深度强化学习的计算卸载和缓存策略

Fig. 6-3 Deep reinforcement learning aided computation offloading mechanism with content cache

(1) 状态空间。在时间槽 t 边缘云 $n \in \mathcal{N}$ 和缓存内容文件 $d \in \mathcal{D}$ 由动作空间的参数实现的。不同的动作 a_m^{comm} 和 a_m^{cache} 决定不同的系统状态。

(2) 动作空间。在这个边缘云系统中，智能体应该根据有限的通信资源来确定将这个任务卸载到边缘云的位置，不管卸载的任务是否已经缓存在服务器中。那么，在 t 时刻的动作 $a_m(t)$ 定义为

$$a_m(t) = \{a_m^{comm}(t), a_m^{cache}(t)\}. \quad (6-11)$$

其中 $a_m^{comm}(t)$ 和 $a_m^{cache}(t)$ 的具体描述如下。

首先，定义行向量 $a_m^{comm}(t) = [a_{m,1}^{comm}(t), a_{m,2}^{comm}(t), \dots, a_{m,N}^{comm}(t)]$ ，其中 $a_{m,i}^{comm}(t)$ ， $i \in \{1, 2, \dots, N\}$ 表示用户设备 m 是否连接到边缘云。具体而言， $a_{m,i}^{comm}(t)$ 的取值是 $\{0, 1\}$ 。 $a_{m,i}^{comm}(t) = 0$ 表示在时刻 t 用户设备 m 选择将任务卸载到宏蜂窝基站边缘云；否则， $a_{m,i}^{comm}(t) = 1$ 表示在时刻 t 用户设备 m 选择将任务卸载到小蜂窝基站边缘云。

接下来，定义行向量 $a_m^{cache}(t) = [a_{m,1}^{cache}(t), a_{m,2}^{cache}(t), \dots, a_{m,N}^{cache}(t)]$ ，其中 $a_{m,j}^{cache}(t)$ ， $j \in \{1, 2, \dots, N\}$ 表示用户设备 m 的内容文件是否缓存在边缘云中。 $a_{m,j}^{cache}(t)$ 的取值是 $\{0, 1\}$ 。 $a_{m,j}^{cache}(t) = 0$ 表示在时刻 t 用户设备 m 的内容文件没有缓存在宏蜂窝基站边缘云；否则， $a_{m,i}^{cache}(t) = 1$ 表示在时刻 t 用户设备 m 的内容文件缓存在小蜂窝基站边缘云。

(3) 回报函数。这种边缘云系统的回报函数设计目标是最大化通信模型、计算模型和缓存模型的回报值。用户设备 m 的回报函数公式建模如下：

$$R_m(t) = R_m^{comm}(t) + R_m^{cache}(t). \quad (6-12)$$

其中， $R_m^{comm}(t)$ 表示从边缘云中通信模型所获得的回报值，包括宏蜂窝基站边缘云和小蜂窝基站边缘云的通信模型回报值，它的具体公式如下：

$$\begin{aligned} R_m^{comm}(t) &= (1 - a_m^{comm}(t)) R_{m0}^{comm} + a_m^{comm}(t) R_{mn}^{comm} \\ &= (1 - a_m^{comm}(t)) (\alpha_{m0} r_{m0} - \beta_{m0} w_{m0}) + \\ &\quad a_m^{comm}(t) (\alpha_{mn} r_{mn} - \beta_{mn} w_{mn}). \end{aligned} \quad (6-13)$$

$R_m^{cache}(t)$ 表示内容文件是否缓存在边缘云中所获得的回报值，包括宏蜂窝基站边缘云和小蜂窝基站边缘云中的计算模型回报值和缓存模型回报值，其具体公式如 (6-14) 所示。 $a_m^{comm}(t)$ 和 $a_m^{cache}(t)$ 表示任务卸载方式和缓存方式的参数。 $\phi_m q_m$ 表示基站边缘云从用户设备 m 中的收入， $\phi_m e_m$ 表示基站边缘云中服务器使用的花费。 $\psi_m l_{A'_m}$ 表示基站边缘云从用户设备 m 收取到的收入， $\gamma_m A'_m$ 表示基站边缘云为回程带宽支付的使用费用。

$$\begin{aligned} R_m^{cache}(t) &= (1 - a_m^{comm}(t)) a_m^{cache}(t) R_{m0}^{cache} + (1 - a_m^{comm}(t)) \\ &\quad (1 - a_m^{cache}(t)) R_{m0}^{comp} + a_m^{comm}(t) a_m^{cache}(t) R_{mn}^{cache} \\ &\quad + a_m^{comm}(t) (1 - a_m^{cache}(t)) R_{m0}^{comp} \\ &= (1 - a_m^{comm}(t)) a_m^{cache}(t) (\psi_{m0} l_{A'_m} - \gamma_{m0} A'_m) + \\ &\quad (1 - a_m^{comm}(t)) (1 - a_m^{cache}(t)) (\phi_{m0} q_{m0} - \phi_{m0} e_{m0}) + \\ &\quad a_m^{comm}(t) a_m^{cache}(t) (\psi_{mn} l_{A'_m} - \gamma_{mn} A'_m) + \\ &\quad a_m^{comm}(t) (1 - a_m^{cache}(t)) (\phi_{mn} q_{mn} - \phi_{mn} e_{mn}). \end{aligned} \quad (6-14)$$

在计算卸载过程中,如果该任务内容缓存在边缘云中,则策略选择缓存模型公式,否则,需要在边缘云中重新计算该任务内容。

6.5 仿真结果

在本节中,基于深度强化学习的计算卸载和缓存策略将在 Anaconda3 Spyder 上进行了仿真和评估。首先,第 6.5.1 小节介绍了仿真实验场景和参数。然后,为了证明该方案的有效性,在第 6.5.2 小节将实验结果与其他现有算法进行了比较,并讨论了算法的参数影响。

6.5.1 仿真实验参数设置

与第四章的仿真实验场景相同,本章的仿真场景超密集网络由宏蜂窝基站边缘云、小蜂窝基站边缘云和终端用户组成。具体来说,宏蜂窝基站边缘云和小蜂窝基站边缘云能够提供计算服务。宏蜂窝基站边缘云数设为 1,小蜂窝基站边缘云数设为 10,并提前设定在固定位置。终端用户在宏蜂窝基站覆盖范围内随机分布。仿真场景如图 6-4 所示。

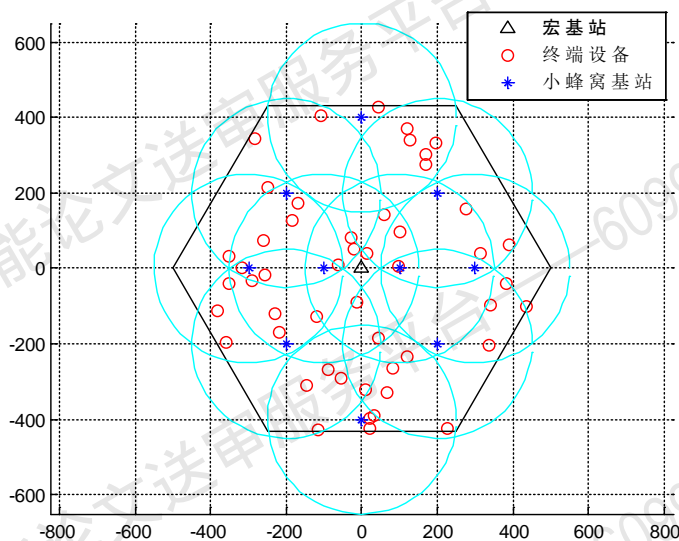


图 6-4 实验仿真场景

Fig. 6-4 Simulation scenario

用户设备到宏蜂窝基站的传输信道带宽设置为 10MHz,用户设备到小蜂窝基站的传输信道带宽设置为 1 MHz。此外,用户设备的传输功率是从 50mW 到 100mW 随机产生的。高斯噪声的方差设为 $\sigma^2 = -100$ dBm。此外,干扰因子为 -100 dBm。通道增益表示为 $h = d^{-\chi}$,其中 χ 表示通过损耗因子,并将其设置为 4。表 6-2 提供了有关此实验中其他参数的更多详细信息。

该方案由一个输入层、两个隐藏层和一个输出层组成。同时，第一隐藏层设置为 120 个隐神经元，第二隐藏层设置为 80 个。训练间隔设置为 10，训练批大小为 128。此外，Adam 优化算法的学习率为 0.005，内存大小为 1024。

表 6-2 参数设置

Table 6-2 Parameter Settings

参数	设定值
宏蜂窝基站边缘云收取的接入费用	3 units/bps
小蜂窝基站边缘云 n 收取的接入费用	1 units/bps
宏蜂窝基站边缘云支付的使用费用	3×10^{-4} units/Hz
小蜂窝基站边缘云 n 支付的使用费用	1×10^{-4} units/Hz
由宏蜂窝基站边缘云收取的计算费用	0.8 units/J
小蜂窝基站边缘云 n 收取的计算费用	0.4 units/J
宏蜂窝基站边缘云支付的计算成本	0.2 units/J
小蜂窝基站边缘云 n 支付的计算成本	0.1 units/J
宏蜂窝基站边缘云收取的存储费用	20 units/byte
小蜂窝基站边缘云收取的存储费用	10 units/byte
宏蜂窝基站边缘云支付的回传成本	0.2 units/bps
小蜂窝基站边缘云支付的回传成本	0.1 units/bps

6.5.2 仿真实验结果

在本小节中，将验证基于深度强化学习计算卸载和缓存策略的实验效果。此外，还将在接下来的实验中研究参数对仿真结果的影响。图 6-5 和图 6-6 显示了本方案在 $UE=20$ 时的回报函数值和学习率的变化。从图 6-5 可以看出，随着迭代次数的增加，回报函数值在前 400 次迭代时逐渐增加，收敛后保持稳定。如图 6-6 所示，训练损失在第一次 100 次迭代时迅速减少，然后保持不变。综上所述，提出的方案可以通过探索来实现收敛，损失值可以降低到 20% 左右。

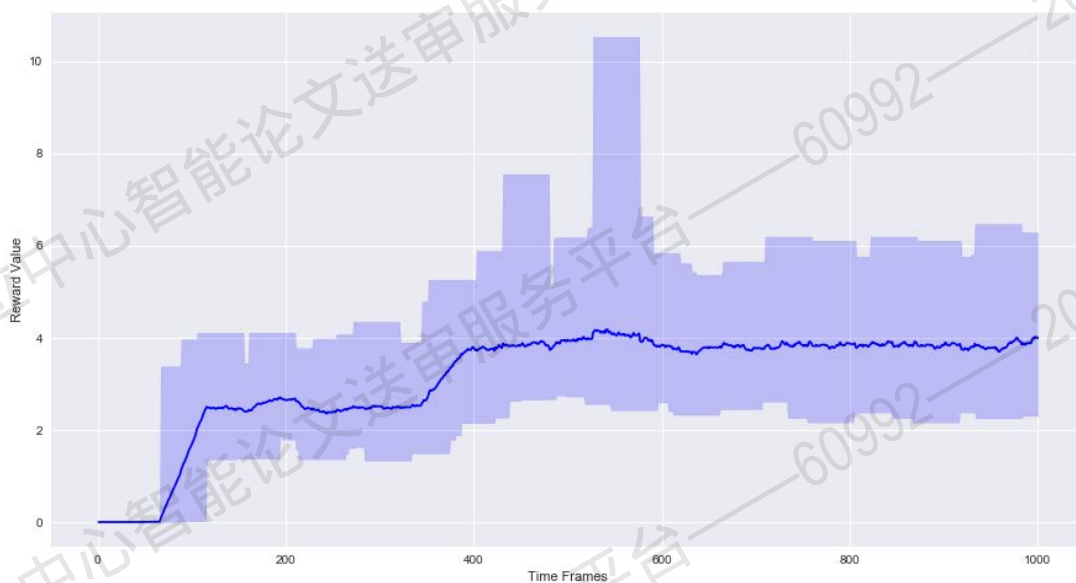


图 6-5 回报函数值变化

Fig. 6-5 Reward value changes versus iterations

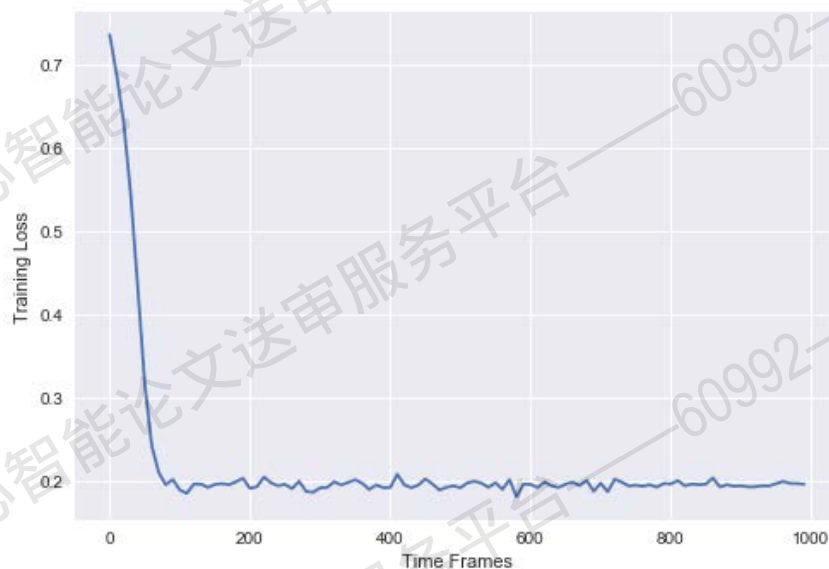


图 6-6 训练损失值变化

Fig. 6-6 Learning rate changes versus iterations

图 6-7 反映了不同设备数量下的回报函数值变化，设备数量分别设为 {5,10,20}。通过实验结果图可以发现，设备数量设置为 5 时，收敛后的回报函数值最小，设备数量设置为 20 时的回报函数值最高，但是收敛速度实在 350 代左右开始迅速增长的，略晚于设备数量为 10 的函数曲线。

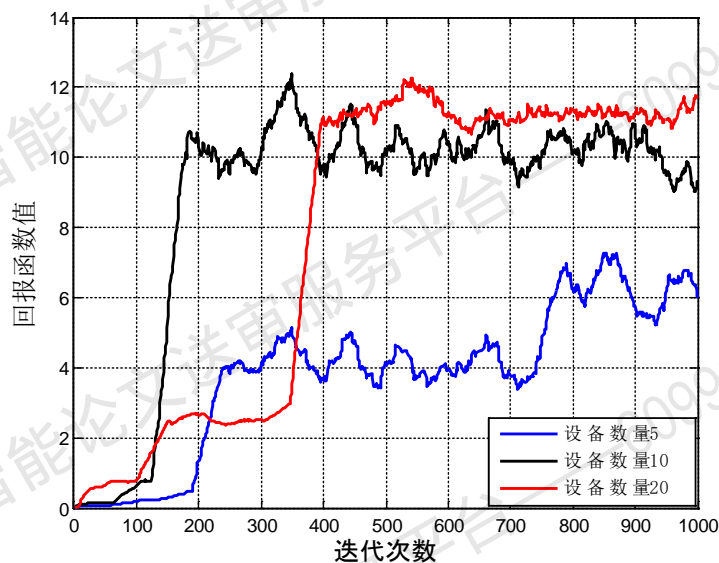


图 6-7 不同设备数量下的回报函数值变化

Fig. 6-7 Reward value changes with different UE numbers

接下来, 提供了深度强化学习算法中三个重要参数(学习率、批量大小和训练间隔)对仿真结果的影响。学习速率控制模型的学习过程, 具体来说, 学习速率取值过高可能导致快速收敛, 而学习速率过低则容易带来过度拟合。批量大小影响模型的优化程度和速度, 如果批量大小的值太小, 则算法不会在 *epoches* 中收敛, 否则, 很容易陷入局部收敛。同样, 训练间隔也是深部神经网络的一个重要参数。

图 6-8 验证了学习率对仿真结果的影响, 学习率设置为{0.005, 0.01, 0.05, 0.1}。具体来说, 当学习率为 0.01 时, 深度强化学习算法所需的回报函数值在整个迭代过程中表现得比其他算法好。学习率为 0.1 不能达到学习率为 0.01 的效果, 但性能优于其他两种情况。另外, 学习率 0.05 的性能最差。

如图 6-9 所示, 批量大小设置为{32, 64, 128, 256}。具体来说, 当批量大小等于 32 时, 深度强化学习算法所要求的回报函数值在前期阶段的表现不好, 但是后期增长迅速, 并最终取得了最好的性能。批量大小为 64 的性能略低于批量大小为 32 的情况。此外, 批量大小为 128 和批量大小为 256 表现相对较差。

图 6-10 描述了不同训练间隔随着迭代次数的回报函数值变化。具体来说, 训练间隔设置为{5, 10, 15, 20}。显然, 当训练间隔为 10 时, 深度强化学习算法所需的回报函数值在整个训练阶段表现良好。当训练间隔为 5, 15, 20 时, 回报函数值表现相对较差, 尤其是在训练间隔为 5 时算法表现最差。

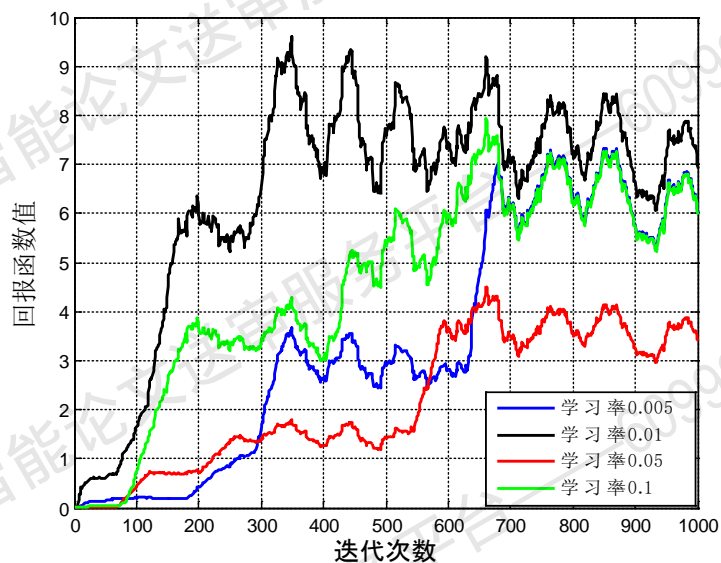


图 6-8 不同学习率下的回报函数值变化

Fig. 6-8 Reward value changes with different learning rate

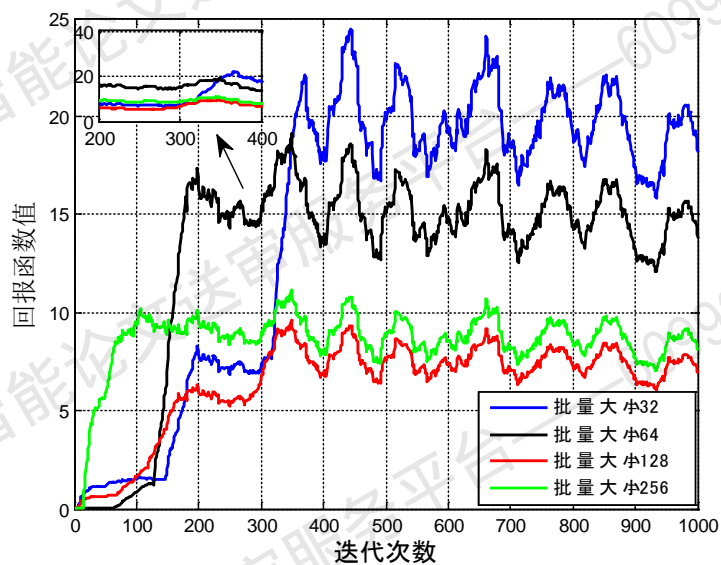


图 6-9 不同批量大小下的回报函数值变化

Fig. 6-9 Reward value changes with different batch size

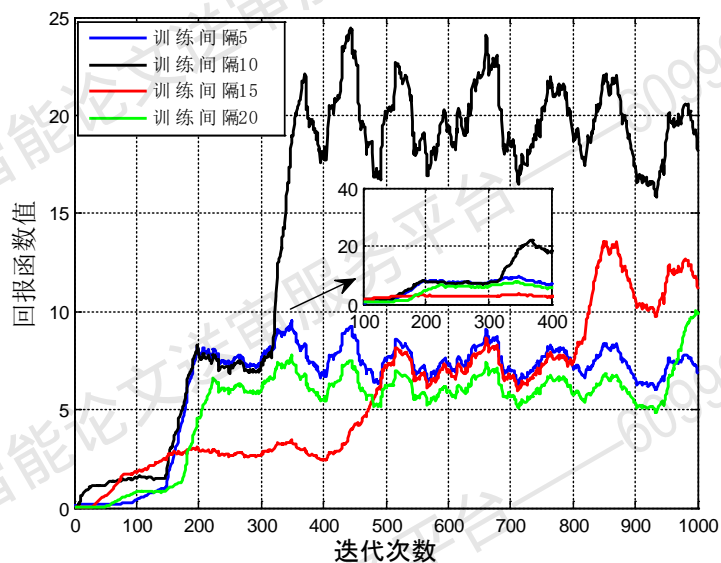


图 6-10 不同训练间隔下的回报函数值变化

Fig. 6-10 Reward value changes with different interval

6.6 本章小结

本章提出了一种在超密集网络中基于深度强化学习算法的计算卸载和缓存策略。在超密集网络的网络架构中,所提出的算法能够根据边缘计算中通信资源、计算资源和缓存资源自适应的提供有效的计算卸载策略。仿真结果表明,该方案可以快速收敛到满意解。最后,验证了该方案的不同参数对实验结果的影响。

第7章 总结与展望

移动互联网与物联网作为传统移动通信的颠覆者和万物互联模式的开拓者,持续驱动着移动通信系统向第五代迈进。5G 移动通信系统承载了诸多的愿景,如超高的连接能力和流量密度、超低时延、业务与用户的智能感知等,而移动边缘计算成为了 5G 移动通信系统的代表性技术之一。移动边缘计算通过将云计算服务下沉至移动边缘,能够为用户设备以超低的时延提供计算、存储等服务。然而,边缘云中的通信资源、计算资源和存储资源有限,如何提高为用户设备卸载任务的效率成为该领域的研究重点之一。本文利用软件定义网络解耦数据平面和控制平面的思想,对整个无线网络进行全局管理,针对软件定义网络中的任务卸载问题做了深入研究,研究内容如下:

本文第二章介绍了软件定义无线网络下的计算卸载策略相关研究概述。首先描述了无线网络技术的演进,分析了从 1G 到 5G 移动通信技术的更迭,并着重介绍了 5G 网络中的超密集蜂窝网络技术,本文后续的研究内容也是基于此网络架构。接下来,详细叙述了软件定义无线网络相关研究,包括软件定义无线网络的基本概念及背景和软件定义无线网络的研究进展。最后,对移动边缘计算研究现状进行概述,并详细介绍了移动边缘计算的基本概念及背景、计算卸载问题研究现状和资源管理问题研究现状。

本文第三章中在软件定义无线网络场景下,提出了面向低能耗的多控制器部署策略。软件定义无线网络的网络状态主要由控制器负责,随着软件定义无线网络元素的增长,单个软件定义无线网络控制器由于自身能力的限制,难以控制所有元素。同时,单个控制器一旦发生故障,会造成整个网络的瘫痪。针对多个控制器难以合理部署问题,利用数据场方法自适应地对网络节点进行分簇,合理部署多个控制器。与粒子群算法等启发式算法相比,仿真实验证明该策略可以有效节省多控制器的传输能耗,降低计算时间复杂度。

本文第四章在软件定义无线网络场景下,提出了面向低时延的多用户计算卸载服务策略。大多数无线网络中的基站都在多信道环境中工作,存在多个用户设备之间如何实现有效的无线接入协调以应对计算卸载的挑战。因此,在软件定义无线网络场景中,针对计算卸载的资源受限的挑战,多个用户设备卸载时间冗长的问题,根据多个用户设备能耗、传输时延和计算时延构建资源分配模型,并利用离散布谷鸟搜索算法合理分配通信资源,与其他启发式算法相比,该算法可以有效缩短任务卸载的平均时延。

本文第五章在软件定义无线网络场景下,提出了面向经济效益的多用户计算

卸载服务策略。在计算卸载中,用户设备可以任意地选择不同运营商以及不同的无线接入方式,以获得丰富的应用服务。用户设备之间存在竞争关系,他们会更偏向于接入到服务质量更好、价格更低的接入网络。同时多个无线网络运营商为吸引用户设备,相互之间也存在着竞争关系。运营商希望通过收费来支付投资成本,获得自身收益。运营商在制定价格时,必须考虑到竞争对手的价格策略,因为用户能够自主选择一个服务质量和价格等更优的运营商网络接入。运营商和用户设备彼此之间的这种交互特性是一个典型的斯坦科尔伯格博弈问题。在软件定义无线网络场景中,针对边缘云合理定价实现利润最大化的问题,引入社会交互信息帮助构建边缘云和用户设备的效用函数,构建斯坦科尔伯格博弈模型,并利用动态迭代算法求出满意的纳什均衡解。

本文第六章在软件定义无线网络场景下,提出了面向智能化的多用户计算卸载服务策略。在移动边缘计算环境中,边缘服务器不具备足够的计算能力和存储能力,资源管理问题存在于边缘云服务器中有以下原因:第一,卸载到边缘云的任务具有不同特性、延迟要求、计算要求和能源消耗等,传统的方法不可能准确地对这些问题建模。第二,边缘云无法满足边缘卸载任务的动态变化。在软件定义无线网络场景中,针对边缘服务器的资源管理问题,利用深度强化学习算法,针对不同的环境状态做出最优的决策,考虑边缘云的计算、通信和存储资源,为任务实时分配资源提供了一种满意的解决方案。

本文针对软件定义无线网络下的计算卸载问题进行了深入研究,证明了方法的可行性与有效性,但是仍有很多不足,未来的研究可以从以下几方面进行深入开展:

(1) 本文的研究内容都是基于用户设备位置固定不变的情况,然而实际应用中应该考虑用户设备的移动性影响,构建适应用户设备移动的数学模型,同时应该参考更多的环境因素影响,如动态电压频率调整、自适应带宽与信道调整、频谱变化等;

(2) 考虑多个用户设备在进行计算卸载时,可能会由于信道干扰等原因导致计算卸载失败,因而需要将信道干扰因素进行建模考虑;在对服务器集群进行建模时,应当增加考虑机器间通信时延消耗和能源消耗,以实现在高效的任务调度与处理的同时节约集群的能耗;

(3) 新颖的机器学习方法发展日新月异,不断的尝试最新的机器学习研究成果引入计算卸载策略研究中,以提升计算卸载策略的效率。

参考文献

- [1] Index C V N. Global mobile data traffic forecast update, 2016–2021[J]. white paper, 2017.
- [2] 李德光,郭兵,张瑞玲,马友忠,任祯琴,赵旭鸽,谭庆,李君科.基于视觉显著性的 AMOLED 显示器多区域功耗优化[J/OL]. 软件学报 :1-15[2020-02-25]. <https://doi.org/10.13328/j.cnki.jos.005939>.
- [3] Taleb T, Ksentini A, Chen M, et al. Coping with emerging mobile social media applications through dynamic service function chaining[J]. IEEE Transactions on Wireless Communications, 2015, 15(4): 2859-2871.
- [4] Liang L, Ye H, Li G Y. Spectrum sharing in vehicular networks based on multi-agent reinforcement learning[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(10): 2282-2292.
- [5] Xu F, Li Y, Wang H, et al. Understanding mobile traffic patterns of large scale cellular towers in urban environment[J]. IEEE/ACM transactions on networking, 2016, 25(2): 1147-1161.
- [6] 尹文枫. 移动医疗中数据处理和智能解析方法的研究[D].北京邮电大学,2019.
- [7] Gai K, Wu Y, Zhu L, et al. Privacy-preserving energy trading using consortium blockchain in smart grid[J]. IEEE Transactions on Industrial Informatics, 2019, 15(6): 3548-3558.
- [8] Ge X. Ultra-reliable low-latency communications in autonomous vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 5005-5016.
- [9] Zhang Y, Tu Z, Wang Q. TempoRec: Temporal-topic based recommender for social network services[J]. Mobile Networks and Applications, 2017, 22(6): 1182-1191.
- [10] 尚博东. 高效超密集蜂窝网络理论研究[D].西安电子科技大学,2018.
- [11] Taleb T, Ksentini A, Jantti R. "Anything as a Service" for 5G Mobile Systems[J]. IEEE Network, 2016, 30(6): 84-91.
- [12] Zhang S, Zhang N, Zhou S, et al. Energy-aware traffic offloading for green heterogeneous networks[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5): 1116-1129.
- [13] 卢军球. 移动边缘计算中依赖性任务卸载的时延与能耗优化算法研究[D].华中科技大学,2019.
- [14] 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.
- [15] 司鹏搏,张倩,于非,张延华.异构移动云计算网络中基于 QoS 的动态资源管理研究(英文)[J].中国通信,2014,11(05):144-159.
- [16] 钟杰瑛. 移动边缘计算中内容缓存与资源租赁的联合设计研究[D].华中科技大学,2019.
- [17] 李虎. 基于移动边缘计算的超密集网络任务卸载策略研究[D].重庆邮电大学,2019.
- [18] Kreutz D, Ramos F M V, Verissimo P E, et al. Software-defined networking: A comprehensive survey[J]. Proceedings of the IEEE, 2014, 103(1): 14-76.
- [19] Ksentini A, Bagaa M, Taleb T. On using SDN in 5G: The controller placement problem[C]//2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, 4-8 December 2016.
- [20] Bagaa M, Taleb T, Ksentini A. Service-aware network function placement for efficient traffic

- handling in carrier cloud[C]//2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, 6-9 April 2014.
- [21] 鲁焱光,王兴伟,李福亮,黄敏.软件定义网络中的动态负载均衡与节能机制[J/OL].计算机报,2019:1-15[2020-02-29].<http://kns.cnki.net/kcms/detail/11.1826.TP.20191216.1554.006.html>.
- [22] 程超,滕俊杰,赵艳领,宋梅.一种基于多智能体强化学习的流量分配算法[J/OL].北京邮电大学学报:1-7[2020-02-29].<https://doi.org/10.13190/j.bupt.2019-140>.
- [23] M. A. Adedoyin and O. E. Falowo, Combination of Ultra-Dense Networks and Other 5G Enabling Technologies: A Survey[J]. IEEE Access. 2020(8): 22893-22932.
- [24] Del Peral-Rosado J A, Raulefs R, Lopez-Salcedo J A, et al. Survey of Cellular Mobile Radio Localization Methods: from 1G to 5G[J]. IEEE Communications Surveys & Tutorials, 2017:1-1.
- [25] Bojkovic Z S, Bakmaz M R, Bakmaz B M. Research challenges for 5G cellular architecture[C]// 2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Nis, 14-17 October 2015.
- [26] 朱金奇,孙华志,黄永鑫,刘明.软件定义网络中延迟满足的路由选择与实时调度更新[J].软件学报,2019,30(11):3440-3456.
- [27] Velasco L, Asensio A, Castro A, et al. Cross-stratum orchestration and flexgrid optical networks for data center federations[J]. IEEE Network, 2013, 27(6): 23-30.
- [28] 钟旭霞. 网络虚拟化环境下服务功能链编排机制[D].北京邮电大学,2019.
- [29] 费宁,徐力杰,成小惠.基于软件定义无线网络的协作式路由实现和性能分析[J].计算机研究与发展,2019,56(05):967-976.
- [30] Sharma R K, Rawat D B. Advances on security threats and countermeasures for cognitive radio networks: A survey[J]. IEEE Communications Surveys & Tutorials, 2014, 17(2): 1023-1043.
- [31] Feamster N, Rexford J, Zegura E. The road to SDN[J]. Queue, 2013, 11(12): 20-40.
- [32] 林耘森箫,毕军,周禹,张程,吴建平,刘争争,张乙然.基于 P4 的可编程数据平面研究及其应用[J].计算机学报,2019,42(11):2539-2560.
- [33] 孙晨,毕军,郑智隆,王舒鹤,胡宏新.MicroNF:基于微服务的异构网络功能虚拟化框架[J].通信学报,2019,40(08):54-59.
- [34] Feamster N, Balakrishnan H, Rexford J, et al. The case for separating routing from routers[C]//Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, Karlsruhe Germany August, 2003.
- [35] Iqbal H, Znati T. Distributed control plane for 4D architecture[C]//IEEE Global Telecommunications Conference, Washington, DC, 26-30 November, 2007.
- [36] Casado M, Freedman M J, Pettit J, et al. Ethane: Taking control of the enterprise[J]. ACM SIGCOMM computer communication review, 2007, 37(4): 1-12.
- [37] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
- [38] 俞洋. 基于服务感知的 5G 动态移动性管理方案设计[D].南京邮电大学,2019.
- [39] 贺琛.基于 5G 的超密集无线网络规划与部署方案的探讨[J].信息通信,2019(12):115-118.
- [40] Akyildiz I F, Wang P, Lin S C. SoftAir: A software defined networking architecture for 5G

- wireless systems[J]. Computer Networks, 2015, 85: 1-18.
- [41] Namal S, Ahmad I, Gurtov A, et al. SDN based inter-technology load balancing leveraged by flow admission control[C]//2013 IEEE SDN for Future Networks and Services (SDN4FNS), Trento, 11-13 November, 2013.
- [42] Usman M, Gebremariam A A, Raza U, et al. A software-defined device-to-device communication architecture for public safety applications in 5G networks[J]. IEEE Access, 2015, 3: 1649-1654.
- [43] Erran L, Morley L Z, Rexford M J. CellSDN: software-defined cellular networks[J]. 2012.
- [44] Mi X, Tian Z, Xu X, et al. NO stack: A SDN-based framework for future cellular networks[C]//2014 International Symposium on Wireless Personal Multimedia Communications (WPMC), Sydney, 7-10 September, 2014.
- [45] Lei T, Lu Z, Wen X, et al. SWAN: An SDN based campus WLAN framework[C]//2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Aalborg, 11-14 May, 2014.
- [46] Syrivelis D, Paschos G S, Tassiulas L. Virtuemanager: A software-defined network architecture for wifi-based metropolitan applications[C]//2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, 1-3 December, 2014.
- [47] Zhao D, Zhu M, Xu M. Supporting "One Big AP" illusion in enterprise WLAN: An SDN-based solution[C]//2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP), Hefei, 23-25 October, 2014.
- [48] Han Z, Ren W. A novel wireless sensor networks structure based on the SDN[J]. International Journal of Distributed Sensor Networks, 2014, 10(3): 874047.
- [49] Sayyed R, Kundu S, Warty C, et al. Resource optimization using software defined networking for smart grid wireless sensor network[C]//2014 3rd International Conference on Eco-friendly Computing and Communication Systems, Mangalore, 18-21 December, 2014.
- [50] De Gante A, Aslan M, Matrawy A. Smart wireless sensor network management based on software-defined networking[C]//2014 27th Biennial Symposium on Communications (QBSC), Kingston, 1-4 June 2014.
- [51] Heller B, Sherwood R, McKeown N. The controller placement problem[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 473-478.
- [52] Gao C, Wang H, Zhu F, et al. A particle swarm optimization algorithm for controller placement problem in software defined network[C]//International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 16 December.
- [53] Hock D, Hartmann M, Gebert S, et al. Pareto-optimal resilient controller placement in SDN-based core networks[C]//Proceedings of the 2013 25th International Teletraffic Congress (ITC), Shanghai, 10-12 September, 2013.
- [54] ul Huque M T I, Si W, Jourjon G, et al. Large-scale dynamic controller placement[J]. IEEE Transactions on Network and Service Management, 2017, 14(1): 63-76.
- [55] Hu Y, Wendong W, Gong X, et al. Reliability-aware controller placement for software-defined networks[C]//2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Zhangjiajie, 10-12 August, 2013.
- [56] Liao J, Sun H, Wang J, et al. Density cluster based approach for controller placement problem in large-scale software defined networkings[J]. Computer Networks, 2017, 112: 24-35.

- [57] 高先明, 王宝生, 邓文平, 陶静. SDN 网络中控制器放置问题综述[J]. 通信学报, 2017, 38(07): 155-164.
- [58] 夏永峰. 面向可靠性和延迟优化的软件定义网络中控制器放置问题的研究[D]. 合肥工业大学, 2017.
- [59] Wang C, Mei W, Qin X, et al. Quantum entropy based tabu search algorithm for energy saving in SDWN[J]. Science China Information Sciences, 2017, 60(4): 040307.
- [60] Andrews J G, Buzzi S, Wan C, et al. What Will 5G Be?[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(6): 1065-1082.
- [61] 夏雪. 5G系统中的超密集网络技术研究[D]. 南京邮电大学, 2019.
- [62] Palattella M R, Dohler M, Grieco A, et al. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(3): 510-527.
- [63] 张平, 陶运铮, 张治. 5G若干关键技术评述[J]. 通信学报, 2016, 37(7): 15-29.
- [64] 谢永, 吴黎兵, 何炎祥, 等. 无缝隙的车联网协助下载方法[J]. 通信学报, 2016, 37(1): 180-190.
- [65] Yu R, Ding J, Huang X, et al. Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach[J]. IEEE Transactions on Vehicular Technology, 2016, 65(10): 7844-7856.
- [66] Anjum A, Abdullah T, Tariq M, et al. Video Stream Analysis in Clouds: An Object Detection and Classification Framework for High Performance Video Analytics[J]. IEEE Transactions on Cloud Computing, 2016, 7(4): 1152-1167.
- [67] 王佳顺. 移动增强现实关键技术研究[D]. 南京邮电大学, 2018.
- [68] Al-Shuwaili A, Simeone O. Optimal resource allocation for mobile edge computing-based augmented reality applications[J]. arXiv preprint arXiv:1611.09243, 2016.
- [69] 李一倩, 刘留, 李慧婷, 张琨, 袁泽. 工业物联网无线信道特性研究[J]. 物联网学报, 2019, 3(04): 34-47.
- [70] 周鹏, 徐金城, 杨博. 工业物联网中基于边缘计算的跨域计算资源分配与任务卸载[J/OL]. 物联网学报: 1-9[2020-02-29]. <http://kns.cnki.net/kcms/detail/10.1491.TP.20200217.1707.004.html>.
- [71] Al-Rubaye S, Kadhum E, Ni Q, et al. Industrial internet of things driven by SDN platform for smart grid resiliency[J]. IEEE Internet of Things Journal, 2017, 6(1): 267-277.
- [72] Navarro-Ortiz J, Sendra S, Ameigeiras P, et al. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things[J]. IEEE Communications Magazine, 2018, 56(2): 60-67.
- [73] CISCO, "The Internet of Things how the next evolution of the Internet is changing everything," White paper, Apr. 2011. [Online]. Available: http://www.cisco.com/c/dam/en/us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- [74] You C, Huang K, Chae H. Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5): 1757-1771.
- [75] 宋富洪, 邢焕来, 潘炜. 一种面向移动云计算的多目标任务卸载算法[J]. 物联网学报, 2019, 3(03): 41-49.
- [76] 崔勇, 宋健, 缪葱葱. 移动云计算研究进展与趋势[J]. 计算机学报, 2017, 40(2): 273-295.

- [77] 李继蕊, 李小勇, 高云全, 等. 5G网络下移动云计算节能措施研究[J]. 计算机学报, 2017, 40(7):1491-1516.
- [78] 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.
- [79] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [80] 王月. 面向移动边缘网络的资源管理研究[D]. 北京邮电大学, 2019.
- [81] ETSI, "Mobile-edge computing introductory technical white paper," White Paper, Mobile-edge Computing Industry Initiative. [Online]. Available: <https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edgecomputing-introductorytechnicalwhitepaper-v1>
- [82] Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]// Edition of the Mcc Workshop on Mobile Cloud Computing, 13-16 August, 2012.
- [83] 董文远, 丁任霜, 黄文艳, 张光华. 车载社交网络中基于雾计算的信任评估算法[J]. 小型微型计算机系统, 2019, 40(06):1291-1298.
- [84] 汪成亮, 黄心田. 智能环境下基于雾计算的推理节点优化分配研究[J]. 电子学报, 2020, 48(01):35-43.
- [85] Wood T, Ramakrishnan K K, Shenoy P, et al. CloudNet:dynamic pooling of cloud resources by live WAN migration of virtual machines[J]. IEEE/ACM Transactions on Networking, 2015, 23(5):1568-1583.
- [86] Papadimitratos P, La Fortelle A, Evenssen K, et al. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation[J]. IEEE Communications Magazine, 2009, 47(11):84-95.
- [87] Lu N, Cheng N, Zhang N, et al. Connected Vehicles: Solutions and Challenges[J]. IEEE Internet of Things Journal, 2014, 1(4):289-299.
- [88] Uhlemann E. Introducing Connected Vehicles [Connected Vehicles][J]. IEEE Vehicular Technology Magazine, 2015, 10(1):23-31.
- [89] 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.
- [90] Kutscher D, Kutscher D, Kutscher D, et al. Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions[C]// The Workshop on Mobile Edge Communications, 7-12 August, 2017.
- [91] Li L, Li Y, Hou R. A Novel Mobile Edge Computing-Based Architecture for Future Cellular Vehicular Networks[C]// IEEE Wireless Communications and Networking Conference, San Francisco, 19-22 March, 2017.
- [92] 李孟涵. 面向已部署 802.11 无线局域网的视频组播性能优化研究[D]. 清华大学, 2016.
- [93] 郭富强. 智能视频监控系统中的前景目标检测技术研究[D]. 西安电子科技大学, 2019.
- [94] 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.
- [95] 王畅. 基于增强现实的智能运维技术研究与应用[D]. 山东大学, 2019.
- [96] 王雄宇. 基于增强现实的实时公交系统 Android 客户端的设计与实现[D]. 北京邮电大学, 2019.

- [97] Kaur K, Garg S, Aujla G S, et al. Edge Computing in the Industrial Internet of Things Environment: Software-Defined-Networks-Based Edge-Cloud Interplay[J]. IEEE Communications Magazine, 2018, 56(2):44-51.
- [98] 张琪, 胡宇鹏, 嵇存,等. 边缘计算应用:传感数据异常实时检测算法[J]. 计算机研究与发展, 2018, 55(3):524-536.
- [99] Liu J, Mao Y, Zhang J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems[C]//2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, 10-15 July, 2016.
- [100] Mao Y, Zhang J, Letaief K B. Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12):3590-3605.
- [101] Kamoun M, Labidi W, Sarkiss M. Joint resource allocation and offloading strategies in cloud enabled cellular networks[C]// IEEE International Conference on Communications, London, 8-12 June, 2015.
- [102] Labidi W, Sarkiss M, Kamoun M. Energy-optimal resource scheduling and computation offloading in small cell networks[C]// IEEE International Conference on Telecommunications, Sydney, 27-29 April, 2015.
- [103] Chen M H, Liang B, Dong M. A semidefinite relaxation approach to mobile cloud offloading with computing access point[C]// IEEE, International Workshop on Signal Processing Advances in Wireless Communications, Stockholm, 28 June-1 July, 2015.
- [104] Muñoz O, Pascual-Iserte A, Vidal J. Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading[J]. IEEE Transactions on Vehicular Technology, 2015, 64(10):4738-4755.
- [105] Barbarossa S, Sardellitti S, Lorenzo P D. Joint allocation of computation and communication resources in multiuser mobile cloud computing[C]// IEEE Signal Processing Advances in Wireless Communications, Darmstadt, 16-19 June, 2013.
- [106] Sardellitti S, Scutari G, Barbarossa S. Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing[J]. IEEE Transactions on Signal & Information Processing Over Networks, 2015, 1(2):89-103.
- [107] Tao X, Hou Y, et al. An energy-optimal offloading algorithm of mobile computing based on HetNets[C]// IEEE International Conference on Connected Vehicles and Expo, Shenzhen, 19-23 October, 2015.
- [108] Zhao Y, Zhou S, Zhao T, et al. Energy-efficient task offloading for multiuser mobile cloud computing[C]// IEEE International Conference on Communications in China, Shenzhen, 2-4 November, 2015.
- [109] Yang L, Zhang H, Li M, et al. Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G[J]. IEEE Transactions on Vehicular Technology, 2018, PP(99):1-1.
- [110] Wang Y, Sheng M, Wang X, et al. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling[J]. IEEE Transactions on Communications, 2016, 64(10):4268-4282.
- [111] Chen X, Jiao L, Li W, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing[J]. IEEE/ACM Transactions on Networking, 2015, 24(5):2795-2808.
- [112] Muñoz O, Iserte A P, Vidal J, et al. Energy-latency trade-off for multiuser wireless computation offloading[C]// IEEE Wireless Communications and Networking Conference

- Workshops, Istanbul, 6-9 April, 2014.
- [113] Wang C, Yu F R, Liang C, et al. Joint Computation Offloading and Interference Management in Wireless Cellular Networks with Mobile Edge Computing[J]. IEEE Transactions on Vehicular Technology, 2017, 66(8):7432-7445.
- [114] Zhao T, Zhou S, Guo X, et al. A Cooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-Aware Mobile Cloud Computing[C]// IEEE Globecom Workshops, San Diego, 6-10 December, 2015.
- [115] Guo X, Singh R, Zhao T, et al. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems[C]// IEEE International Conference on Communications, Kuala Lumpur, 22-27 May, 2016.
- [116] 于博文, 蒲凌君, 谢玉婷, 徐敬东, 张建忠. 移动边缘计算任务卸载和基站关联协同决策问题研究[J]. 计算机研究与发展, 2018, 55(03):537-550.
- [117] Wang C, Liang C, Yu F R, et al. Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing [J]. IEEE Transactions on Wireless Communications, 2017, 16(8):4924-4938.
- [118] Ranadheera S, Maghsudi S, Hossain E. Mobile Edge Computation Offloading Using Game Theory and Reinforcement Learning[J]. arXiv preprint arXiv:1711.09012, 2017.
- [119] You C, Huang K, Chae H, et al. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3):1397-1411.
- [120] Tanzil S M S, Gharehshiran O N, Krishnamurthy V. Femto-Cloud Formation: A Coalitional Game-Theoretic Approach[C]// IEEE Global Communications Conference, San Diego, 6-10 December, 2015.
- [121] Oueis J, Calvanese-Strinati E, De Domenico A, et al. On the impact of backhaul network on distributed cloud computing[C]// IEEE Wireless Communications and Networking Conference Workshops, Istanbul, 6-9 April, 2014.
- [122] Oueis J, Strinati E C, Barbarossa S. Small cell clustering for efficient distributed cloud computing[C]// IEEE, International Symposium on Personal, Indoor, and Mobile Radio Communication, Boston, 6-9 September, 2015.
- [123] Oueis J, Strinati E C, Sardellitti S, et al. Small Cell Clustering for Efficient Distributed Fog Computing: A Multi-User Case[C]// IEEE Vehicular Technology Conference, Boston, 6-9 September, 2015.
- [124] Oueis J, Strinati E C, Barbarossa S. The Fog Balancing: Load Distribution for Small Cell Cloud Computing[C]// IEEE Vehicular Technology Conference, Glasgow, 11-14 May, 2015.
- [125] Vondra M, Becvar Z. QoS-ensuring distribution of computation load among cloud-enabled small cells[C]// IEEE, International Conference on Cloud Networking, Luxembourg, 8-10 October, 2014.
- [126] Rawat D B, Reddy S. Recent advances on software defined wireless networking[C]// SoutheastCon 2016, IEEE, Norfolk, 30 March-3 April, 2016.
- [127] Wang Y, Tao X, He Q, et al. A dynamic load balancing method of cloud-center based on SDN[J]. China Communications, 2016, 13(2): 130-137.
- [128] 齐月震. SDN 控制器放置优化技术与系统实现[D]. 北京邮电大学, 2019.
- [129] Lange S, Gebert S, Zinner T, et al. Heuristic approaches to the controller placement problem in large scale SDN networks[J]. IEEE Transactions on Network and Service Management,

- 2015, 12(1): 4-17.
- [130] 黄子桦. 软件定义网络中多控制器部署问题研究[D]. 电子科技大学, 2019.
- [131] Abdel-Rahman M J, Mazied E D A, Mac Kenzie A, et al. On stochastic controller placement in software-defined wireless networks[C]//2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, 19-22 March, 2017.
- [132] Knight S, Nguyen H X, Falkner N, et al. The internet topology zoo[J]. IEEE Journal on Selected Areas in Communications, 2011, 29(9): 1765-1775.
- [133] 王潇, 杨金民. 一种基于 IBPSO 的 SDN 控制器放置优化方案[J]. 计算机应用研究, 2019, 36(10): 3069-3071+3075.
- [134] Wang S, Gan W, Li D, et al. Data field for hierarchical clustering[J]. International Journal of Data Warehousing and Mining, 2011, 7(4): 43-63.
- [135] Shaw J A. Radiometry and the Friis transmission equation[J]. American journal of physics, 2013, 81(1): 33-37.
- [136] Halim M A, Razak M S, Yasin M N M, et al. Optimization on wireless power transfer[J]. Jurnal Teknologi, 2015, 77(28): pp. 17-21.
- [137] Berger A L, Pietra V J D, Pietra S A D. A maximum entropy approach to natural language processing[J]. Computational linguistics, 1996, 22(1): 39-71.
- [138] Mirjalili S, Dong J S, Lewis A, et al. Particle swarm optimization: theory, literature review, and application in airfoil design[M]//Nature-inspired optimizers. Springer, Cham, 2020: 167-184.
- [139] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique[J]. Pattern recognition, 2000, 33(9): 1455-1465.
- [140] Schnass K, Vandergheynst P. Dictionary preconditioning for greedy algorithms[J]. IEEE Transactions on Signal Processing, 2008, 56(5): 1994-2002.
- [141] 高云全. 物联网环境下数据聚合关键技术研究[D]. 北京邮电大学, 2019.
- [142] Hossain M S, Muhammad G. Cloud-assisted industrial internet of things (iiot)-enabled framework for health monitoring[J]. Computer Networks, 2016, 101: 192-202.
- [143] Simsek M, Aijaz A, Dohler M, et al. 5G-enabled tactile internet[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(3): 460-473.
- [144] 贾庆民. 5G 移动通信网络中缓存与计算关键技术研究[D]. 北京邮电大学, 2019.
- [145] Cui Z, Sun B, Wang G, et al. A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems[J]. Journal of Parallel and Distributed Computing, 2017, 103: 42-52.
- [146] 龙笑, 周良, 郑洪源. 基于布谷鸟算法的 Storm 集群动态负载均衡策略[J]. 计算机技术与发展, 2019, 29(10): 164-169+190.
- [147] Li X, Yin M. Modified cuckoo search algorithm with self adaptive parameter method[J]. Information Sciences, 2015, 298: 80-97.
- [148] Iosifescu M. Finite Markov processes and their applications[M]. Courier Corporation, 2014.
- [149] Sisinni E, Saifullah A, Han S, et al. Industrial internet of things: Challenges, opportunities, and directions[J]. IEEE Transactions on Industrial Informatics, 2018, 14(11): 4724-4734.
- [150] Zhao N, Li D, Liu M, et al. Secure transmission via joint precoding optimization for downlink MISO NOMA[J]. IEEE Transactions on Vehicular Technology, 2019, 68(8): 7603-7615.

- [151] Ning Z, Kong X, Xia F, et al. Green and sustainable cloud of things: Enabling collaborative edge computing[J]. IEEE Communications Magazine, 2018, 57(1): 72-78.
- [152] Zhenyu Z, Haijun L, Bo G, et al. Robust Mobile Crowd Sensing: When Deep Learning Meets Edge Computing[J]. IEEE Network, 2018, 32(4):54-60.
- [153] J. Du, C. Jiang, A. Benslimane, S. Guo, and Y. Ren, Stackelberg differential game based resource sharing in hierarchical fog-cloud computing[C]. IEEE Global Communications Conference. Waikoloa, Hawaii, USA, 9-13 Dec. 2019
- [154] Zheng Z, Song L, Han Z, et al. A Stackelberg game approach to proactive caching in large-scale mobile edge networks[J]. IEEE Transactions on Wireless Communications, 2018, 17(8): 5198-5211.
- [155] 高宏超,陈晓江,徐丹,彭瑶,汤战勇,房鼎益.无源感知网络中能耗和延迟平衡的机会路由协议[J].软件学报,2019,30(08):2528-2544.
- [156] Jeffrey R M C D J. The Wright ω Function[M]// Artificial Intelligence, Automated Reasoning, and Symbolic Computation. 2002.
- [157] C. V. N. Index, "Forecast and methodology 2017-2022," Cisco: San Jose, CA, USA, 2019.
- [158] Tran T X, Hajisami A, Pandey P, et al. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges[J]. IEEE Communications Magazine, 2017, 55(4):54-61.
- [159] Teng Y, Liu M, Yu F R, et al. Resource allocation for ultra-dense networks: A survey, some research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2018, 21(3): 2134-2168.
- [160] 王月. 面向移动边缘网络的资源管理研究[D].北京邮电大学,2019.
- [161] 卢军球. 移动边缘计算中依赖性任务卸载的时延与能耗优化算法研究[D].华中科技大学,2019.
- [162] 王月. 面向移动边缘网络的资源管理研究[D].北京邮电大学,2019.
- [163] 朱晓强. 考虑竞争制造商供应链的零售商定价策略研究[D].电子科技大学,2015.
- [164] 左超,武继刚,史雯隼.移动边缘计算中的端到端任务分配算法[J/OL].计算机应用研究:1-7[2020-02-25].<https://doi.org/10.19734/j.issn.1001-3695.2019.01.0013>.
- [165] Yao H, Li M, Du J, et al. Artificial Intelligence for Information-Centric Networks [J]. IEEE Communications Magazine, 2019, 57(6): 47-53.
- [166] Yao H, Mai T, Jiang C, et al. AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing[J]. IEEE Computational Intelligence Magazine, 2019, 14(4): 21-30.

附录 中英文缩略词对照表

缩写	英文全称	中文
3DTV	3 Dimension Television	三维电视
5G	5th generation mobile networks	第五代移动通信技术
AR	Augmented Reality	增强现实技术
CPP	Controller Placement Problem	控制器部署问题
CPU	Central Processing Unit	中央处理器
CSA	Cuckoo Search Algorithm	布谷鸟搜索算法
CSG	Closed Subscriber Group	闭合用户组
D2D	Device-to-Device	终端直通
GA	Genetic Algorithm	遗传算法
GL	Greedy Algorithm	贪婪算法
GSM	Global System for Mobile Communications	全球移动通信系统
HDTV	High Definition Television	高清晰度电视
IIoT	Industrial Internet of Things	工业物联网
IoT	Internet of Things	物联网
IPTV	Internet Protocol Television	交互式网络电视
LTE	Long Term Evolution	全球通用标准
MBS	Macro-cell Base Station	宏蜂窝基站
MCC	Mobile Cloud Computing	移动云计算
MEC	Mobile Edge Computing	移动边缘计算
MIMO	Multi Input Multi Output	多输入多输出
OFDMA	Orthogonal Frequency Division Multiple Access	正交频分多址
PSO	Particle Swarm Algorithm	粒子群算法
QoS	Quality of Service	服务质量

RRH	Remote Radio Head	射频拉远头
SBS	Small-cell Base Station	小蜂窝基站
SDN	Software Defined Networking	软件定义网络
SDWN	Software Defined Wireless Networks	软件定义无线网络
TDMA	Time Division Multiple Access	时分多址
UDN	Ultra Dense Network	超密集网络
UE	User Equipment	用户设备
V2I	Vehicle to Infrastructure	车辆对基础设施
V2V	Vehicle to Vehicle	车辆对车辆
VoIP	Voice over Internet Protocol	基于 IP 的语音传输
VPN	Virtual Private Network	虚拟专用网络
WLAN	Wireless Local Area Network	无线局域网
WSN	Wireless Sensor Network	无线传感器网络

攻读博士学位期间所发表的成果

学术论文

- 1 **Feixiang, Li**, Haipeng Yao, Jun Du, Chunxiao Jiang and Yi Qian. Stackelberg Game Based Computation Offloading in Social and Cognitive IIoT [J]. IEEE Transactions on Industrial Informatics. (Early access, SCI 源刊, 中科院一区, IF=7.377)
- 2 **Feixiang, Li**, Xiaobin Xu, Haipeng Yao, Jingjing Wang, Chunxiao Jiang and Song Guo. Multi-Controller Resource Management for Software Defined Wireless Networks [J]. IEEE Communications Letters. 2019, 3(23):506-509. (SCI: 000461240300030, 中科院三区, IF=3.457)
- 3 **Feixiang Li**, Xiaobin Xu, Xiao Han, Shengxin Gao, Yupeng Wang. Adaptive Controller Deployment in Software Defined Wireless Networks [J]. China Communications. 2019, 11(16): 81-92. (SCI: 000498158600007, 中科院三区, IF=1.882)
- 4 **Feixiang Li**, Haipeng Yao, Jun Du, Chunxiao Jiang, Richard. Yu. Green communication and computation offloading in Ultra-dense networks [C]. IEEE 2019 Global Communications Conference. (CCFC 类推荐会议)
- 5 **Feixiang Li**, Xiaobin Xu. A Discrete Cuckoo Search Algorithm for the Controller Placement Problem in Software Defined Networks [C]. 2018 IEEE Annual Information Technology, Electronics and Mobile Communication Conference. (EI 会议)
- 6 **Feixiang Li**, Haipeng Yao, Jun Du, Chunxiao Jiang, Zhu Han, Yunjie Liu. Auction Design for Edge Computation Offloading in SDN-based Ultra Dense Networks[J]. IEEE Transactions on Mobile Computing. (SCI 在审)
- 7 Tianle Mai, Haipeng Yao, **Feixiang Li**, et al. Computing Resource Allocation in LEO Satellites System: A Stackelberg Game Approach[C]//2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, 2019: 919-924. (EI 会议)
- 8 Yinyang Liu, Xiaobin Xu, **Feixiang Li**. Image and Text Correlation Judgement Based on Deep Learning[C]//2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2018: 844-847. (EI 会议)
- 9 Yinyang Liu, Xiaobin Xu, **Feixiang Li**. Image Feature Matching Based on Deep Learning[C]//2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, 2018: 1752-1756. (EI 会议)
- 10 Siyuan He, Tao Li, Yuxin Duan, Zhenning Yang and **Feixiang Li**. VAE Based-NCF for Recommendation of Implicit Feedback[C]//2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). IEEE, 2019: 512-516. (EI 会议)

发明专利

- 1 一种基于遥感卫星管理协议的卫星管理系统, **李飞翔**, 徐晓斌, 李晓伟, 姚海鹏, 已受理。
- 2 一种自适应的多控制器部署方法, **李飞翔**, 徐晓斌, 李晓伟, 姚海鹏, 已受理

致 谢

时光飞逝，转眼四年的博士研究生生活已经到了尾声，这既是一个终点，也是人生下一个阶段的起点。在此，谨向曾经帮助、关心、指导过我的老师、同学、朋友和亲人表示诚挚的谢意和衷心的感谢。

首先，感谢我的导师刘韵洁院士。您以大家的气度、深邃的思维、广阔的视野给我指明了学习方向，同时也提供了良好的学习与研究环境，这些都是伴随我一生的财富。

其次，感谢指导我的姚海鹏老师、姜春晓老师、钱毅老师、杜军博士和王景璟博士。从小论文的发表到大论文的开题、中期和终稿完成，各位老师都倾注了大量的心血和汗水，教会了我如何写规范的学术论文、如何严谨的进行学术研究，带我真正的进入了学术的殿堂，体会到了研究的乐趣。同时，各位老师平易近人态度无论对我以后工作和生活都将产生深远影响。

感谢徐晓斌老师、王露瑶老师、靳晓宁老师以及中心的其他老师们。各位老师具有深厚的专业知识和宽阔的视野，总能在我迷茫的时候给我提出相应给建议和可行的解决方案。感谢你们为我在学术上和生活上提供的帮助与指导，祝愿各位老师工作顺利。

感谢实验室的已经毕业和在校的师弟师妹们，感谢 UNL 交流访学期间遇到的朋友们，感谢一路走来在学术和生活上的帮助，真心的祝愿你们学业有成，工作顺利。

感谢我的父母，一直默默支持着我，为了我花费了毕生的精力和心血，养育之恩永生难忘。在我离开校园走向社会之际，衷心祝福父母安康，幸福。

此外，谨向在百忙之中参加论文评审和答辩的所有老师致以崇高的敬意和衷心的感谢。

完成本论文之时，正值新冠病毒疫情肆虐之际，在此向奋斗在祖国各地的广大抗疫一线工作者致以崇高的敬意！