

分类号: TN929.5

密 级: 公开

学校代码: 10363

学 号: 2210330115



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 移动边缘计算任务调度
与能耗优化方法研究

论 文 作 者 韩芬

校 内 导 师 余诺 副教授

校 外 导 师 赵义正

专业学位类别 电子信息 (085400)

研究方向 (领域) 人工智能与系统集成

论文提交日期: 2024 年 5 月 31 日

分类号: TN929.5

单位代码: 10363

密 级: 公开

学 号: 2210330115

题 目 移动边缘计算任务调度与能耗优化方法研究

题 目 Research on Task Scheduling and Energy Consumption
Optimization for Mobile Edge Computing

学生姓名: 韩芬

校内导师: 余诺 副教授

校外导师: 赵义正

专业学位类别: 电子信息 (085400)

研究方向: 人工智能与系统集成

论文答辩日期: 2024 年 5 月 19 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：韩芳

日期：2024年5月24日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：韩芳

指导教师签名：余诺

日期：2024 年 5 月 24 日

日期：2024 年 5 月 24 日

移动边缘计算任务调度与能耗优化方法研究

摘要

随着现代通信技术的发展，移动应用程序对计算资源的需求日益增长。然而，受限于移动用户设备的计算资源约束，这些需求往往难以得到充分满足。为解决这一问题，移动边缘计算作为一种新技术应运而生。移动边缘计算通过将计算任务从资源受限的用户设备转移到更靠近用户的基站所部署的边缘计算服务器中，实现计算任务的近端处理。这种方法不仅显著降低用户设备上的计算负载，还有效减少数据传输过程中产生的时延，从而为用户提供了更加流畅的应用体验。

在多用户环境下，为提高计算任务的处理效率，需要对用户提交的计算任务进行合理调度。通过优化计算任务调度策略，在满足计算任务响应时延和带宽资源约束的前提下，实现移动边缘网络资源的高效利用，在保证计算服务质量的同时，减少移动边缘网络能耗，对于推动移动计算的可持续发展具有重要意义。

本文基于移动边缘计算网络架构，建立计算任务调度模型，并明确旨在最大化基站服务用户数量的优化目标。基于此模型动态调整任务分配，可以适应网络中用户负载的不断变化。为求解计算任务调度优化问题，本文将其转化为一个马尔可夫决策过程，其决策目标是在基站服务能力受限的情况下，最大化处理用户计算任务的数量。通过采用深度强化学习算法，本文实现对用户计算任务与基

站之间分配关系的快速决策，有效地提升基站服务用户设备的能力。仿真结果表明，与现有算法相比，本文所提出的计算任务调度方法，可以在相同的响应时延和带宽资源约束条件下，满足更多的边缘计算任务需求。

为减少移动网络在提供边缘计算服务过程中产生的能耗，降低网络运行成本，本文进一步研究移动边缘计算能耗优化问题。本文基于所提出的计算任务调度算法，进一步研究提出移动边缘计算能耗优化算法。该算法通过计算每个基站的资源利用率，并在此基础上做出基站动态开关状态的决策，以优化整体能耗。通过综合考虑基站的资源利用效率和用户设备的连接状态，该算法能够对符合条件的基站进行动态状态调整，并重新分配用户计算任务。实验结果表明，与整数线性规划方法和基于最近邻域分配原则的贪婪启发式方法相比，本文所提出的方法在降低移动边缘计算能耗方面具有明显的优势。

关键词：移动边缘计算，任务调度，能耗优化，时延约束，带宽约束，深度强化学习

RESEARCH ON TASK SCHEDULING AND ENERGY CONSUMPTION OPTIMIZATION FOR MOBILE EDGE COMPUTING

ABSTRACT

With the evolution of modern communication technologies, mobile applications are increasingly demanding in terms of computational resources. However, the computational resource constraints of mobile user devices often prevent these demands from being fully met. To address this challenge, Mobile Edge Computing (MEC) has emerged as a novel technology. MEC facilitates the offloading of computational tasks from resource-constrained user devices to edge computing servers deployed at base stations closer to the users, enabling the near-user processing of tasks. This approach not only significantly reduces the computational load on user devices but also effectively decreases latency in data transmission, thereby providing users with a smoother application experience.

In a multi-user environment, to enhance the efficiency of computational task processing, it is necessary to reasonably schedule the computational tasks submitted by users. By optimizing the task scheduling strategy, the efficient utilization of mobile edge network resources is achieved, while satisfying the response latency and bandwidth resource

constraints of computational tasks. This approach is crucial for ensuring the quality of computational services and reducing the energy consumption of mobile edge networks, which is significant for promoting the sustainable development of mobile computing.

This thesis establishes a computational task scheduling model based on the mobile edge computing network architecture and defines an optimization objective aimed at maximizing the number of users served by base stations. Based on this model, computational tasks can be scheduled adapting to the fluctuating user loads in the network. To solve the optimization problem of computational task scheduling, this thesis formulates it as a Markov Decision Process (MDP), with the decision goal of maximizing the number of computational tasks processed under the constraint of base station service capabilities. By employing a deep reinforcement learning algorithm, this thesis realizes rapid decision-making regarding the relationship between computational tasks and base stations, effectively enhancing the ability of base stations to serve user devices. Simulation results indicate that, compared to existing algorithms, the computational task scheduling method proposed in this thesis can meet more edge computing task demands under the same response latency and bandwidth resource constraints.

To reduce the energy consumption generated by mobile networks in providing edge computing services and to lower network operational

costs, this thesis further investigates the energy optimization issue in mobile edge computing. Building upon the computational task scheduling algorithm proposed in this thesis, an energy optimization algorithm for mobile edge computing is further proposed. This algorithm calculates the resource utilization rate of each base station and makes decisions on the on/off states of base stations to optimize overall energy consumption. By comprehensively considering the resource utilization efficiency of base stations and the connection status of user devices, the algorithm dynamically adjusts the on/off states of base stations and reallocates user computational tasks. Experimental results demonstrate that compared to integer linear programming methods and greedy heuristic methods based on the nearest neighborhood allocation principle, the method proposed in this thesis has a distinct advantage in reducing the energy consumption of mobile edge computing.

KEY WORDS: mobile edge computing, task scheduling, energy consumption optimization, delay constraint, bandwidth constraint, deep reinforcement learning

Han Fen (Electronic information)

Supervised by Professor Yu Nuo

目录

摘要.....	I
ABSTRACT.....	III
第 1 章 绪论.....	1
1.1 课题研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 移动边缘计算任务调度的研究现状.....	2
1.2.2 移动边缘计算能耗优化的研究现状.....	4
1.3 本文主要研究内容.....	7
1.4 本文的结构安排.....	8
第 2 章 移动边缘计算调度优化相关理论与分析.....	10
2.1 移动边缘计算概述.....	10
2.2 移动边缘计算任务调度方法.....	11
2.3 移动边缘计算能耗优化方法.....	13
2.4 基于深度强化学习的调度优化方法.....	14
2.5 本章小结.....	17
第 3 章 基于深度强化学习的移动边缘计算任务调度研究.....	19
3.1 系统模型.....	19
3.1.1 任务调度模型.....	20
3.1.2 带宽资源模型.....	20
3.1.3 计算资源模型.....	21
3.2 移动边缘计算任务调度问题建模.....	22
3.3 移动边缘计算任务调度算法设计.....	23
3.4 仿真实验结果分析.....	26
3.4.1 仿真实验参数设置.....	26
3.4.2 实验结果分析.....	27
3.5 本章小结.....	31

第 4 章 基于深度强化学习的移动边缘计算能耗优化研究.....	32
4.1 系统模型.....	32
4.1.1 能耗优化模型.....	33
4.2 边缘计算网络能耗优化问题建模.....	35
4.3 基于深度强化学习移动边缘计算能耗优化算法设计.....	36
4.4 仿真实验结果分析.....	39
4.4.1 仿真实验参数设置.....	39
4.4.2 仿真实验结果分析.....	39
4.5 本章小结.....	41
第 5 章 结论与展望.....	42
5.1 结论.....	42
5.2 展望.....	43
参考文献.....	44
攻读学位期间取得的学术成果目录.....	50
致 谢.....	51

第1章 绪论

1.1 课题研究背景和意义

随着 5G 时代的快速发展，万物互联逐渐成为现实。同时，丰富的移动应用对计算资源的需求，也给移动用户设备带来了极大的挑战。为了获得足够的计算资源，传统的方法是将需要处理的用户计算任务通过传输到远端的云计算中心，经过计算处理后，将计算的结果回传到用户设备。但是，现在很多新兴的移动应用对计算的实时性提出了更高的要求，传统的云计算服务模式已经不能满足新兴移动应用高标准的计算需求。因此，为了解决用户设备计算资源相对缺乏的问题，移动边缘计算(Mobile Edge Computing, MEC)技术应运而生^[1]。移动边缘计算通过将计算任务从资源受限的用户设备转移到更靠近用户的基站所部署的边缘计算服务器中，可以实现计算任务的近端处理^[2]。这种计算服务模式有效地缓解移动应用对云计算资源的依赖，同时减少远距离数据传输，从而降低了服务用户的响应时延。这些突出优势使得移动边缘计算获得广泛关注。

在移动边缘网络中，用户设备产生的海量数据要求能够被实时处理，并提供低延迟、高可靠性的服务，这使得计算任务调度管理成为关键环节。基站的计算资源和通信资源的有限性，用户设备的移动性，以及用户负载的动态变化，均给移动边缘计算任务调度带来了挑战。

移动边缘网络作为一种新兴的网络架构，旨在将计算能力和存储资源推向网络的边缘，从而更接近数据源和用户。这种架构的优势在于能够显著降低服务延迟，提高数据处理的实时性和响应速度，满足日益增长的移动数据需求。然而，随着用户设备数量的增加和应用种类的多样化，用户设备产生的数据量呈现出爆炸式增长，这些数据往往需要实时处理并提供低延迟、高可靠性的服务。

在这样的背景下，计算任务调度管理成为移动边缘网络中的一个关键环节。任务调度不仅需要考虑如何高效地分配计算资源，以确保任务能够在规定的时间内完成，还需要考虑如何优化通信资源的使用，以减少数据传输过程中的延

迟和能耗。

移动边缘计算任务调度的主要目标是通过合理的任务分配策略和资源分配机制，将用户请求的计算任务高效分配至靠近用户的边缘服务器，以提高移动边缘网络的资源利用率，并降低服务响应延迟。计算资源和带宽资源的有限性要求分配策略能够有效管理这些资源，以满足不同计算任务的需求。同时，许多应用场景对计算任务处理的实时性有着严格的要求，这就需要高效的调度算法来确保任务的及时完成。此外，移动边缘网络中基站之间的无线信号多重覆盖特性为资源节点间的协作提供了可能。因此，研究高效的计算任务调度方法，充分利用网络中资源节点的协作特性，对于提高通信和计算资源的利用效率、提升边缘计算服务质量具有重要意义^[34]。

为了满足用户设备的移动边缘计算服务需求，需要在移动通信网络中部署数量众多的基站和边缘服务器。这些网络设备的运行不仅产生大量能耗，还带来网络运行成本和碳排放问题。因此，对移动边缘计算的能耗进行优化成为了一个迫切的需求。用户设备的移动性和计算服务需求的波动性导致移动计算任务需求分布具有明显的时空变化特性。因此，动态调整网络设备的工作状态，并根据计算任务需求的变化进行任务分配，可以有效提高网络设备的能量效率，减少移动边缘计算服务过程中的能耗。研究相应的计算任务动态调度算法，在满足服务需求的前提下，动态调整网络设备的工作状态，以达到减少网络能耗的目标，具有重要的研究价值。

综上所述，开展移动边缘计算任务调度和能耗优化的研究对于提升服务性能、改善用户体验、降低能源消耗具有重要意义。这不仅促进了移动边缘计算技术的广泛应用和发展，也为实现可持续发展的移动通信网络提供了技术支撑。

1.2 国内外研究现状

1.2.1 移动边缘计算任务调度的研究现状

近年来，MEC 领域取得了显著的研究进展，特别是在任务调度、能耗优化和资源分配等关键技术方面。学者们不仅提出众多创新算法，还对这些算法进行持续的优化和迭代，以期获得更优的性能。这些研究成果不仅展示了 MEC 技

术的快速发展，也为该领域的进一步探索奠定了坚实的基础。

当前，MEC 领域的研究重点主要集中在任务调度优化上。例如，Subrat 等人^[3]研究了移动边缘计算中的用户分配资源分配问题，旨在满足时间延迟约束的条件下，最大化基站服务器能够服务的用户数量。然而，该研究未充分考虑带宽资源的约束，这在实际应用中可能影响服务质量和效率。唐伦等人^[4]研究了一种基于 Lyapunov 理论的资源分配算法，该算法针对多用户多边缘服务系统，建立了动态调度模型，并在考虑用户时延约束的同时，引入了概率约束条件，实现了无线资源和计算资源的联合分配。李燕君等人^[23]通过引入软件定义网络和网络功能虚拟化技术，改进了 MEC 网络，并提出了基于 Q 学习的深度 Q-网络(Deep Q Network, DQN)在线资源分配方法，实验结果显示该方法能有效提高实时任务处理的成功率。詹文翰等人^[24]针对 MEC 中存在依赖关系的任务卸载决策问题，采用深度强化学习方法，构建了序列到序列的深度神经网络模型，并通过近端策略优化方法进行强化学习训练，实验结果表明该算法在任务卸载决策方面展现出良好的性能。

杨天等人^[10]针对 MEC 环境下服务器计算资源受限且计算任务存在时延约束的情况，提出了一种联合优化方法，用于优化卸载决策与资源分配，并验证了该方法的有效性。童钊等人^[11]在考虑数据和计算节点安全性的背景下，提出了一种基于深度强化学习的算法，旨在解决 MEC 中的卸载和资源调度问题，研究结果表明该算法在满足用户服务质量需求方面表现优异。此外，文献[12]提出了一个针对任务分配与移动设备协调关系的优化框架，旨在减少总体任务延迟和网络能耗。Liu 等人^[14]针对复杂的用户服务器关系的 MEC 系统，提出一维搜索算法，采用正交频分多址作为接入技术，优化随机任务调度的策略，在满足功率条件下实现用户设备的时延最小化。王云鹏^[15]深入研究了 MEC 中的实时性和资源分配问题，使用深度强化学习算法，通过技术重构 MEC 网络，有效提升了任务处理速率和传输速率，并在不同场景下验证了算法的有效性和收敛性。

现有研究在移动边缘计算领域的资源调度和任务分配方面已经取得一定的成果，但仍存在一些不足，未来研究需要进一步探讨和完善。

首先，许多研究集中在理论模型的构建和算法的提出，但在实际应用场景

中的可行性和有效性验证相对不足。实际环境中的复杂性和不确定性，如设备异构性、网络动态变化和用户行为的不可预测性等因素。因此，未来的研究需要重点关注算法在真实的环境中的表现，并通过实验或实际部署来验证其有效性。此外，安全性和隐私保护在现有研究中往往没有得到足够的重视。随着数据泄露和隐私侵犯事件的增多，如何在优化资源调度和任务分配的同时，确保数据安全和用户隐私成为亟待解决的问题。未来的研究需要在设计优化算法时，更加注重安全性和隐私保护的考虑。最后，现有研究可能过于关注单一目标的优化，如时延，而忽视了多目标之间的平衡。在实际应用中，可能需要对多个因素指标综合考虑，采用多目标的优化方法，以实现系统性能的全面优化。

综上所述，这些文献的研究都是基于特定的环境，已知网络中用户和基站的数量情况下，研究如何进行资源的调度，传统的决策优化方法，可以用来解决静态规划问题，例如传统的旅行商问题或者路径优化问题。在这种决策过程中，环境是静态的，环境参数是确定性的，不会因为决策动作而发生变化，即动作不会对环境产生影响^[7]。在实际情况中，网络环境是动态变化的，用户的位置和需求、基站的状态以及网络的流量都是动态变化的，这也是任务调度的难点所在。在调度问题上，确保用户在边缘服务器上分配资源，同时满足覆盖范围、资源可用性、延迟要求等方面的限制。但在实际情况下可能无法预知这些信息，所以上述文献所提出的一些方法在解决动态环境调度问题不太适用。许多学者已经在移动边缘计算的资源调度，任务调度和缓存资源调度等方面展开了深入研究。现有研究为 MEC 领域的发展奠定了基础，但仍需在实际应用验证、算法效率、安全性和隐私保护以及多目标优化等方面进行深入研究，以推动 MEC 技术的进一步发展和成熟。

1.2.2 移动边缘计算能耗优化的研究现状

目前在 MEC 研究领域，能耗优化也已成为众多学者关注的焦点。文献[5]探讨了在规定时间内，如何有效地安排和分配计算与存储资源以实现能耗最小化的问题。该研究在模型建立时简化了问题，忽略任务间的依赖关系，这限制算法的普适性。王秋云^[6]针对移动边缘计算中的资源调度问题，以最小化能耗为目标，采用了传统的启发式算法进行研究。高素林等人^[16]提出一种实时调

度算法，旨在将用户请求的任务调度至靠近用户端的 MEC 服务器上，以应对云计算中心资源受限的挑战。他们在多用户场景下构建了相应的模型，并将其转化为能耗最小化问题，实验结果表明该算法能有效降低能量消耗。文献[17]介绍了一种适用于多服务器 MEC 系统的优化框架，针对系统中目标的复杂性和小区间干扰问题，研究两种匹配算法进行卸载优化，可以有效降低系统的计算成本。

文献[18]针对多址的 MEC 服务器系统，提出将正交频分多址技术应用于上行链路的新思路，有效降低网络能耗。邝祝芳等人^[25]针对 MEC 网络中多用户多任务卸载问题，提出适用于该问题的算法，其核心思想是将该问题转换成贪心策略的流水车间调度问题，并通过深度强化学习算法求解，目标是最小化延迟和能耗。该方法在降低延迟和本地耗能方面表现出了显著的优越性能。高云等人^[26]采用深度 Q 学习方法，针对不同 MEC 网络场景中的卸载和资源分配问题进行了深入研究，模型考虑了多重约束条件，仿真结果表明所提算法能改善系统性能，并提高任务数据的计算速率。卢海峰等人^[27]提出结合长短期记忆网络与 DQN 的方法来处理 MEC 中的任务卸载决策问题，有效降低了任务处理时延和传输能耗。杨戈等人^[28]提出一种基于强化学习算法，考虑任务模型、计算模型和信道状态，以确定最优的计算卸载策略，仿真结果显示该算法能降低系统总成本。丁晨阳^[29]针对移动边缘计算环境下的 workflow 任务调度问题，提出一种考虑任务优先级、资源可用性和卸载节点的多用户任务卸载与调度方案。通过建立数学模型和应用改进的自适应遗传算法，实现满足时延要求的同时最小化系统能耗。此外，该文章还探讨了多用户系统中的能耗优化问题，提出一种任务卸载和资源分配的联合机制，并采用迭代优化算法进行求解。仿真实验验证了所提方案在降低系统能耗方面的有效性，并考虑物理层安全技术以增强数据安全性。

现有研究在 MEC 中的能耗优化方面已经取得了一定的进展，但仍存在一些不足之处，需要未来的研究进一步探讨和完善。

首先，文献[5]在探讨能耗最小化问题时，简化问题模型并忽略任务间的依赖关系。这种简化虽然有助于降低问题的复杂性，但可能无法准确反映实际应用中的复杂场景，限制算法的普适性和实际应用效果。文献[6]采用的传统启发

式算法虽然在某些情况下能够有效地解决资源调度问题，但这些算法可能缺乏对 MEC 环境动态变化的适应能力，且可能无法保证全局最优解。文献[16]提出的实时调度算法虽然在多用户场景下能够有效降低能耗，并在高并发、高时延情况下优于遗传算法，但该算法可能需要进一步的优化以应对更加复杂多变的网络条件，特别是在任务数量和用户规模迅速增长的情况下。文献[17]介绍的多服务器 MEC 系统的优化框架，虽然研究了匹配算法进行卸载优化，有效降低系统的计算成本，但该框架可能需要进一步考虑实际部署中的可行性和算法的实时性，以确保在不断变化的网络环境中保持高效运行。

文献[18]提出将正交频分多址技术应用于上行链路以降低网络能耗的新思路，但该方法可能在实际部署时面临兼容性和实施难度的问题。正交频分多址技术的应用需要与现有的无线通信标准和设备相适应，这可能需要额外的硬件支持和软件升级。邝祝芳等人^[25]提出的适用于多用户多任务卸载问题的算法，虽然通过深度强化学习算法求解，并在降低延迟和本地耗能方面表现出优越性能，但深度学习方法通常需要大量的数据和计算资源进行训练，这可能限制其在资源受限的移动边缘设备上的应用。高云等人^[26]采用的深度 Q 学习方法虽然考虑了多重约束条件，但 Q 学习方法可能在面对复杂多变的网络环境时，难以快速适应和收敛，影响系统性能的实时优化。卢海峰等人^[27]提出的结合长短期记忆网络与 DQN 的方法，虽然有效降低任务处理时延和传输能耗，但该方法可能在处理大规模任务卸载决策时面临计算复杂性的问题。杨戈等人^[28]提出的基于强化学习的算法考虑了任务模型、计算模型和信道状态，但可能需要较长的时间来探索和学习最优策略，这种方法在动态变化的网络环境中可能不太适应。丁晨阳^[29]提出的多用户任务卸载与调度方案虽然在满足时延要求的同时最小化系统能耗，但该方案可能需要进一步的验证和测试，以确保在不同的网络条件和应用场景中的普适性。

综上所述，现有研究在能耗优化方面取得一定的成果，但仍需解决实际部署的兼容性问题、深度学习方法的资源消耗和收敛速度问题、算法的计算复杂性以及策略的实时适应性问题。未来的研究需要在这些方面进行改进，以实现更加高效、智能和可靠的能耗优化策略。

1.3 本文主要研究内容

移动边缘网络的设计和运行面临着用户请求任务量的显著波动性，这种波动性对网络的任务调度，资源管理，能耗优化等方面提出新的挑战。用户请求的随机性和数量的不断变化要求网络能够灵活地适应这些动态变化，以确保用户体验和网络效率的最优化。为解决动态环境带来的挑战，出现了深度强化学习算法，是深度学习和强化学习结合的一种机器学习方法，能够自动学习在不同状态下的最佳行动选择。深度强化学习方法(Deep Reinforcement Learning, DRL)算法通过学习网络的历史状态和行为结果，能够自主地制定出最优的调度策略。从上述文献调研分析可知，深度强化学习算法在处理移动边缘计算相关问题时表现了出色的性能。本文采用深度强化学习算法针对 MEC 中的两个核心问题开展研究：计算任务调度和网络能耗优化。

(1) 基于深度强化学习的移动边缘计算任务调度研究

在移动边缘网络中，基站的计算资源和带宽资源是有限的，从而限制其服务用户的能力。因此，计算任务调度问题成为确保用户体验和网络效率的关键环节。有效的任务调度策略不仅需要确保用户请求得到及时处理，满足用户体验的要求，还需要合理分配带宽资源，以避免资源的过度消耗和网络拥塞，从而提高整个网络的效率和性能。本文针对计算服务响应时延约束和带宽资源约束条件下的 MEC 任务调度问题进行深入研究。本文采用马尔可夫决策过程(Markov Decision Process, MDP)模型来描述用户与基站之间的连接问题。MDP 模型能够捕捉时延和带宽的动态变化，为任务调度提供灵活的解决方案。通过定义适当的状态转移概率和奖励函数，MDP 模型可以将任务调度问题转化为一个寻求最大化长期累积奖励的序列决策问题。

在 MDP 模型的基础上，本文进一步采用深度强化学习算法来制定任务调度策略。在调度过程中，不仅考虑基站的覆盖范围和处理用户任务所需的资源量，还考虑边缘服务器的当前负载情况，这些关键因素的综合考虑，使得调度策略更加精准和高效。DRL 算法通过与环境的交互学习最优策略，能够自适应地调整任务分配方案，以适应网络状态的动态变化。实现用户与基站之间的有效匹配，最大化基站服务的用户数量，同时满足计算服务的响应时延约束和带宽资

源的约束。

(2) 基于深度强化学习的移动边缘计算能耗优化研究

通过实时分析网络状态，深度强化学习算法能够智能地调整用户的连接策略，确保每个处于工作状态的边缘服务器在满足服务质量要求的同时，能够服务尽可能多的用户。这种智能调度策略使得网络能够更加高效地利用资源，提高资源利用率，同时也降低因过度分配或资源浪费导致的能耗。

此外，本文提出的动态调度策略通过有效地将用户集中到部分基站上，在合适的时机关闭一些相对空闲的基站，进一步实现网络能耗的优化。这种动态调度策略不仅提高资源利用率，还改善了用户体验。最后，本文通过一些仿真实验验证所提出方法在任务调度方面的有效性。实验结果表明，相比于传统的静态调度方法，本文提出的基于 DRL 的动态调度策略能耗降低方面都有显著的改进。

综上所述，本文的研究为移动边缘网络中的任务调度和能耗资源管理提供了一种新的解决方案。通过 DRL 算法的引入，不仅提高网络的调度效率和资源利用率，实现在满足基站时延和带宽阈值的同时，有效提高基站服务用户的数量；还实现在满足用户服务质量要求的同时，有效降低网络能耗的目标。基于 DRL 算法的方案对解决移动边缘计算中的任务调度和能耗优化问题具有重要意义。

1.4 本文的结构安排

本论文分为 5 个章节进行撰写，每章的主要内容如下：

第 1 章为绪论。首先，深入介绍 MEC 的基础知识，并提出利用深度强化学习解决 MEC 中任务调度和能耗优化问题的研究思路。这一思路的提出不仅在理论上为后续章节的研究提供了指导，也为解决实际问题提供创新的思路。回顾国内外在 MEC 网络任务调度、资源分配和能耗优化领域的研究进展，本章总结了一些现有研究的不足和局限性。最后，明确本文的研究目标，概述论文的整体结构和各章节的基本内容。

第 2 章为移动边缘计算调度优化相关理论与分析。本章对 MEC 调度优化相关的理论与方法进行分析。首先，介绍现有的计算任务调度的优化方法，并进

行深入分析，包括原理、优点以及应用中的局限性。同时，还探讨 MEC 网络能耗优化的挑战和难点，并介绍几种广泛应用的能耗优化算法，旨在提高 MEC 系统的能效。最后，本章详细介绍了深度强化学习算法的理论基础及其在优化问题求解过程中的应用。深度强化学习作为一种前沿的优化技术，为解决复杂的 MEC 调度优化问题提供了新的思路和方法。这也为后面章节内容和仿真实验奠定了理论基础。

第 3 章主要研究基于深度强化学习的移动边缘计算任务调度。首先，对 MEC 系统模型进行详细说明，包括计算资源模型和带宽资源模型。在此基础上，对 MEC 计算任务调度问题进行建模，将其转化为马尔科夫决策过程，设计一种基于深度强化学习的 MEC 任务调度优化算法进行求解，其目标是在资源有限的情况下最大化基站服务的用户数量，提高系统的整体效率。实验仿真结果表明，文中提出的深度强化学习方法在相同的网络资源条件下能够有效地提高基站所服务的用户数量。由此看来，深度强化学习方法在解决一些复杂的 MEC 任务调度问题上具有很大的潜力，也为未来的研究和应用提供方向。

第 4 章为基于深度强化学习的移动边缘计算能耗优化研究。本章分析移动边缘网络的能耗组成。鉴于能耗优化问题与任务调度问题的紧密关联，本章将任务调度作为能耗优化过程的核心部分。建立目标是网络总能耗最小化的优化模型，并设计相应的深度强化学习算法。这一算法，系统可以在考虑任务调度的同时，优化整个网络的能耗情况。仿真实验和对比分析结果，表明本章所提出的算法在网络负载较低时，可以在满足用户计算服务需求的同时有效降低网络总能耗。

第 5 章为总结与展望。对基于深度强化学习的移动边缘计算任务调度和能耗优化方法研究进行了全面的总结。本章主要概括研究的主要成果，并讨论未来的研究方向和潜在的改进空间。

第2章 移动边缘计算调度优化相关理论与分析

本章主要介绍 MEC 调度优化的相关理论知识与分析。首先，本章对 MEC 进行全面的定义，阐述其核心特征，包括低延迟、高带宽和实时性等，并探讨 MEC 在不同场景下的应用，这些应用对 MEC 的性能提出不同的需求和挑战，为本章后续的讨论奠定了基础。下一步，介绍 MEC 领域的两个研究热点：任务调度和网络能耗问题。任务调度问题关注的是如何在 MEC 环境中高效地分配计算任务到边缘节点，以满足不同任务的时延和带宽要求。网络能耗问题主要关注如何在保证服务质量的前提下，优化网络的能耗，减少能源消耗，实现绿色通信。随后，还介绍本文使用的算法——深度强化学习算法。将阐述 DRL 的基本原理、关键技术、以及如何利用 DRL 算法来解决 MEC 中的任务调度和能耗优化问题。本章还将讨论 DRL 算法的主要优势，如自适应学习、决策连续性，以及如何通过神经网络来近似复杂的策略或价值函数。

2.1 移动边缘计算概述

随着物联网时代的到来，对于数据传输的带宽、延迟以及服务的安全性和可靠性提出了日益增长的需求^[30]。物联网设备的快速增加，对网络的负载和性能提出更高的要求。这不仅挑战现有的网络架构，也催生对 MEC 等新兴技术的需求，以实现更快速、可靠和安全的数据传输与处理。传统的云数据中心计算模式，由于集中化的数据处理方式，可能导致很高的延迟和资源损耗，无法满足新兴行业低延迟，高标准的要求。互联网中这些数据不仅具有巨大的体量^[49]，而且对设备性能和服务时延等方面提出了多样化的需求。相比之下，MEC 作为一种新型的网络计算模式^[2]，能够满足多样化需求，有效减少用户和边缘服务器之间的传输延迟，提高边缘服务器计算用户任务的响应速度。随着互联网、物联网、车联网等概念逐渐普及。通信技术的快速发展提升了数据传输和通信质量，用户体验也得到显著提升^[50]。

移动边缘计算通过在靠近用户一侧部署计算节点^[54]，将用户任务分配到这

些部署的计算节点上进行计算，减少时延并节省资源。可以根据用户需求的多少动态调整部署的计算节点的开关状态^[51]，实现计算资源的分布，更好地满足用户不同的需求^[55]。

近几年，移动边缘计算方面的研究也比较多，从大部分的文献来看研究的问题主要有边缘缓存、计算卸载以及服务迁移等方面^[53]。

随着互联网时代的发展和用户设备的广泛应用，缓存技术已经变得很普遍，成为解决许多问题的一种主要方案。边缘缓存^[60]主要是解决同一区域中的用户设备多次请求相同内容的情况，每次都上传到云端进行处理比较浪费资源。所以将多次请求相同的内容缓存到边缘服务器上，可以快速响应用户请求并降低网络流量负载。缓存技术可以降低用户请求的网络时延，提高用户的网络体验，在面向 5G 的移动边缘计算中被视为关键性技术之一^[31]。

计算卸载技术是一种将计算任务从边缘设备转移到云端或其他更强大的计算设备上进行处理的方法。这种方法在边缘部署的服务器上集中处理本地业务，复杂的计算和存储任务转发到云端或其他计算设备进行处理。有助于提高整体的处理效率，降低能耗和减少延迟^[32]。

随着科技的快速发展，移动边缘计算也正在成为处理和管理海量数据的关键技术。在移动边缘计算的环境中，服务的迁移性也成为一个比较重要的问题。服务迁移不仅可以在一些设备故障或过载时提供弹性支撑，还可以在需求和资源之间创建更好的匹配。服务迁移是指将服务从一个计算节点动态迁移到另一个节点，以提高资源利用率和提高服务质量^[63]。

移动边缘计算技术有效的融合无线网络和互联网技术，提供高效的计算、存储和处理功能。MEC 部署策略充分体现了低延迟和高带宽的优点，同时能够实时获取网络 and 位置信息，为用户提供准确的服务^[58]。在不断发展的互联网时代，MEC 具有广泛的应用前景，例如：车联网中的实时交通管理和虚拟现实等领域^[65]。

2.2 移动边缘计算任务调度方法

在 MEC 中，任务调度是研究的热点问题。针对移动边缘网络环境中的大规模任务处理需求，任务调度策略旨在有效地将任务分配给不同的边缘服务器，

以提供服务。这些策略的核心目标是最大化边缘服务器的资源利用率、降低服务延迟、提高用户体验，并确保系统能够灵活地适应用户需求的变化。在任务调度决策过程中，需要综合考虑多种因素。首先，必须评估边缘基站是否具备满足用户任务请求的能力。其次，需要确定基站需要提供的服务类型，以及哪些用户应当被连接到这些服务。此外，任务调度还需考虑带宽资源的可用性和服务的时延约束，以确保任务能够在规定的时间内得到有效处理。为提高任务处理速率、降低计算能耗和时延，任务调度策略必须能够在系统各计算节点之间平衡负载^[59]。这不仅要求算法能够快速响应用户请求，还要能够适应网络条件的动态变化，以及预测和适应未来的需求波动。通过这种方式，MEC 系统能够更高效地服务于更多用户，满足他们在不同场景下的计算需求。

在 MEC 的研究领域，任务调度问题因其对系统性能和用户体验的重要影响而备受关注。为解决这个问题，很多学者提出一些传统算法求解任务调度问题，根据问题的特点和实际需求，设计一系列启发式规则或策略来指导任务调度决策。

其中，基于负载均衡的调度方法是一种常见的策略，其核心目标是平衡各个边缘服务器的负载，避免一些边缘服务器过载而导致性能下降。通过观察边缘服务器的负载情况，并结合用户任务的特性和服务器的资源状态，进行合理的任务分配。为实现负载均衡，学者们设计了多种算法，如基于最小连接数、资源利用率等策略的调度算法。这些算法通过实时监测服务器的资源使用情况，如 CPU、内存和存储等，以及用户任务的紧急程度和计算需求，动态地将任务分配到最合适的边缘服务器上。

此外，还有一些应用场景主要是考虑对时延比较敏感的任务，如智能物联网、车联网等，对任务处理的实时性要求较高。因此，时延在任务调度算法中是很重要的。传统启发式算法可能在求解过程中需要大量的计算资源和时间，导致效率较低，无法满足实时性的要求。

在 MEC 的研究领域中，除了传统启发式算法外，基于优化算法和机器学习的方法也被广泛探索以解决任务调度问题。例如，遗传算法等其他优化算法。这些算法的核心思想是通过搜索最优解来优化任务调度问题。这些算法倾向于寻找局部最优解而不是全局最优解，但它们在移动边缘计算中发挥着重要作用。

然而，任务调度问题往往是一个复杂的组合优化问题，这些传统启发式算法可能存在一定的限制，无法探索到所有可能的解空间，从而导致无法找到全局最优解。此外，传统启发式算法常常用于解决静态环境的问题，需要预先设定环境，设定规则，确定任务调度方案。启发式算法可能在动态环境中没有很好的性能。

综上所述，尽管传统启发式算法在一定程度上可以解决移动边缘计算中的任务调度问题。但随着 MEC 环境的日益复杂和动态变化，这些算法在处理大规模、高动态性的任务调度问题时可能会遇到性能瓶颈。由此看来，MEC 任务调度是一项复杂而关键的工作，其有效性直接影响到移动边缘计算系统的性能和用户体验。因此，未来的研究需要探索更加高效和自适应的算法，如深度强化学习(DRL)等，以应对 MEC 领域中的新挑战。通过不断创新优化，以满足移动边缘计算系统的需求。

2.3 移动边缘计算能耗优化方法

移动边缘计算能耗优化是指通过合理设计和管理移动边缘计算系统的资源利用方式，实现系统的能源消耗的最小化。鉴于每个边缘服务器的资源有限，能耗优化成为 MEC 中一个重要的研究方向。目标是在有限资源的情况下，提高边缘服务器的资源利用率，以满足更多用户的需求，以增强系统的整体性能。在 MEC 任务调度过程中能源消耗也是网络系统运行的重要成本之一。现在社会主张绿色资源，节约能耗，能耗优化也成为网络系统设计的重要考虑因素。在网络计算中需采用一些策略去实现能耗优化的目的，让移动边缘计算系统节能环保可持续。

传统的启发式算法通过一系列的启发式规则或策略来搜索能耗优化问题的解空间。这些算法在 MEC 的任务调度过程中考虑边缘服务器的能源剩余量，优先调度任务到能源充足的节点，减少节点因能源不足而停机或降低性能的情况。此外，根据系统负载情况和能源供应状态，算法动态调整边缘节点的资源配置，以优化能耗。在能源紧缺时，可以降低一些边缘服务器的计算频率、关闭一些空闲的边缘服务器，以达到减少能耗的目的。建立有效的能源管理系统，监控和管理移动边缘计算系统中的能源使用情况。这包括能源消耗的实时监测、能

源使用数据的分析和预测、定制化的能源管理策略等。能耗优化涉及到多个方面的技术和策略，需要综合考虑系统的实际需求、资源限制和性能目标，以找到最佳的能耗优化策略^[22]。

综上所述，这些方法在一定程度上可以达到减少能耗的目的。能耗优化在移动边缘计算中扮演着至关重要的角色。通过采用先进的算法和策略，不仅可以有效减少能源消耗，降低运营成本，提升用户体验。因此，未来的研究和实践应当不断探索和完善能耗优化方法，移动边缘计算中应当充分考虑能耗优化的问题。

2.4 基于深度强化学习的调度优化方法

深度强化学习是本文用到的方法，结合了强化学习和深度学习各自优势的学习方法^[19]，下面介绍强化学习和深度强化学习的一些基本概念。自提出以来，DRL 因其通用性和有效性而备受研究者关注，已在众多领域得到广泛应用，包括组合优化等一些决策复杂的问题领域，具有重要的研究意义^[64]。目前，已有许多将 DRL 算法应用到任务卸载问题中，并取得一定的进展。与传统方法相比，DRL 在延迟和能耗方面表现出较好的性能^[21]。

深度强化学习的求解方法主要依赖于几个核心概念：环境、定义策略、状态空间、动作空间以及奖励机制。通过这些概念，可以将强化学习的任务抽象为 MDP，并进一步利用贝尔曼方程进行表述^[61]。调度问题的优化过程即是贝尔曼方程的优化，它可以用于表示当前的状态和下一状态的价值之间的关系，也可以表示状态值函数和动作值函数的动态关系，这些关系被综合考虑并输入到代理网络中，代理网络通过计算得出最终的 Q 值，用于指导决策过程^[62]。

强化学习是机器学习的一个重要分支，其独特的学习范式在许多领域展现出了巨大的潜力和价值。其核心目标是通过代理与环境交互来学习并采取合适的策略，以最大化奖励或达到特定的目标^[20]。在初始阶段，代理通常缺乏对环境的充分理解，代理可能无法立即做出正确的决策，需要不断训练、探索、调整。事实上，强化学习是代理通过不断尝试、调整的方式学习，从环境的交互中获取奖励，从而指导动作。随着训练的进行，代理会逐渐积累经验，建立起一个策略(Policy)，用于指导其在面对不同状态时选择行动。代理不仅学习如何

在当前状态下采取行动，还学习如何预测未来的状态变化，从而实现长期规划和决策。强化学习逐渐能够自主的做出正确的决策，实现代理获得最大奖赏的目标。此外，强化学习不需要大体量的数据，而是通过与环境的交互来学习，这使得它在许多实际应用中具有独特的优势。

强化学习的核心模型是基于标准的马尔可夫决策过程，该过程包括代理、环境、状态、动作和奖励五个主要部分组成^[23]。代理作为学习主体，通过感知环境的变化做出决策，并根据环境的反馈进行决策调整。环境构成了代理所处的整体情境和条件，它随着代理采取的动作的影响而发生变化，并且这种改变可以被代理感知。环境会在每次决策后向代理提供相应的奖励值。状态 $S = \{s_1, s_2, \dots, s_t\}$ 是对环境的描述，当代理采取行动后，状态也会相应的变化。状态空间表示系统可能处于的状态集合，这些状态可以是离散的或连续的^[9]。动作 $A = \{a_1, a_2, \dots, a_t\}$ 函数表示代理采取的行为和相应的决策结果。在 MDP 中，动作空间指的是所有可能采取行动的集合，这些行动可以是离散的或连续的。奖励 $R = R\{s_t, a_t, s_{t+1}\}$ 是代理根据当前状态和采取的动作所获取的反馈信号，通常以标量函数的形式表示，考虑当前状态、采取的动作以及下一状态。

环境的动态是由条件概率决定的，作为对当前状态下代理选择的行动的反馈，代理的目的就是找到最优策略，使长期奖励最大化^[20]，即

$$R_t = \sum_{i=t}^T r^{i-t} r(s_i, a_i) \quad (2-1)$$

$r \in [0,1]$ 贴现因子，从状态 s_t 开始的预期贴现收益定义为价值

$$V^\pi(s_t, a_t) = \mathbb{E}_{s_i > t \sim E, a_i \geq t \sim \pi} [R_T | s_t] \quad (2-2)$$

上式中 π 表示策略；状态行动函数表示在执行动作 a_t 后的所期望得到的未来奖励的折现值，即

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_i > t \sim E, a_i \geq t \sim \pi} [R_T | s_t, a_t] \quad (2-3)$$

贝尔曼方程在 MDP 被广泛应用，用于描述价值函数和状态-动作函数之间的关系

$$V^\pi(s_t) = \mathbb{E}_{s_{t+1} \sim E, a_t \sim \pi} [r(s_t, a_t) + \gamma(s_t, a_t) V^\pi(s_{t+1})] \quad (2-4)$$

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim E} \left[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} [Q^\pi(s_{t+1}, a_{t+1})] \right] \quad (2-5)$$

在找到的策略最优状态时 π^* ，价值函数的贝尔曼最优方程可重新表述为：

$$V^*(s_t) = \max_{a_t \in A} \mathbb{E}_{s_{t+1} \sim E, a_t \sim \pi} [r(s_t, a_t) + \gamma V^*(s_{t+1})] \quad (2-6)$$

利用价值迭代算法，可以得到任意状态在最优政策下最优值，表达式如下：

$$V_{k+1}(s) = \max_{a \in A} \sum_{s'} p(s' | s, a) r(s, a) + \gamma V_k(s') \quad (2-7)$$

其中 k 是迭代指数，以策略迭代为例，在 MDP 初始化完成后，每次迭代都会利用贝尔曼方程计算状态值函数，以对当前策略进行评估。然后，根据动作值函数的贝尔曼方程确定当前状态下的最佳行动^[20]，公式如下：

$$\pi^*(s) = \arg \max_{a_t \in A} Q^*(s, a) \quad (2-8)$$

针对给定的初始状态和动作，结合蒙特卡罗方法和时序差分(TD)的学习方法进行学习，其中蒙特卡罗方法又叫做统计模拟方法，使用随机数来解决计算问题，可以在不清楚 MDP 状态转移概率及及时奖励的情况下，直接从经历完整的过程中来学习状态价值。获取的状态行动函数的贝尔曼最优性方程如下：

$$Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim E} [r(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})] \quad (2-9)$$

经过以上计算，后续迭代可以使用代理的经验元组 (s_t, a_t, r_t, s_{t+1}) 去更新状态行动函数，以及其他时间 t 状态下的学习预估如下^[57]：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (2-10)$$

其中 α 是学习率， $r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ 表示为 TD 误差。上述公式就是强化学习中计算 Q 值经典的公式。上述就是强化学习的简单过程。

代理利用深度学习的特征，从环境中提取特征，并将这些特征传递给强化学习模型进行决策和动作执行。在执行动作后，代理接收环境的反馈，其中包括新状态和奖励或惩罚，并相应地更新其决策算法。这一迭代过程持续进行，最终使得代理逐渐学会制定调整获得最大长期奖励的策略。

使用神经网络替代 Q 表，可以实时更新状态-动作对的价值函数值，适

用于连续状态空间。神经网络具有强大的函数逼近特性，DQN 技术就是利用一个由 θ 参数化的神经网络来无限接近 Q 值^[35]。如图 2-1,深度强化学习的框架图，包含 DRL 的核心要素，展示了它们之间进行迭代计算的简易过程。其中， θ 代表参数。函数逼近的不稳定问题，可以使用经验回放方法，将代理计算的每个时间步长 t 的值存储在经验池 U 中。同时，从经验

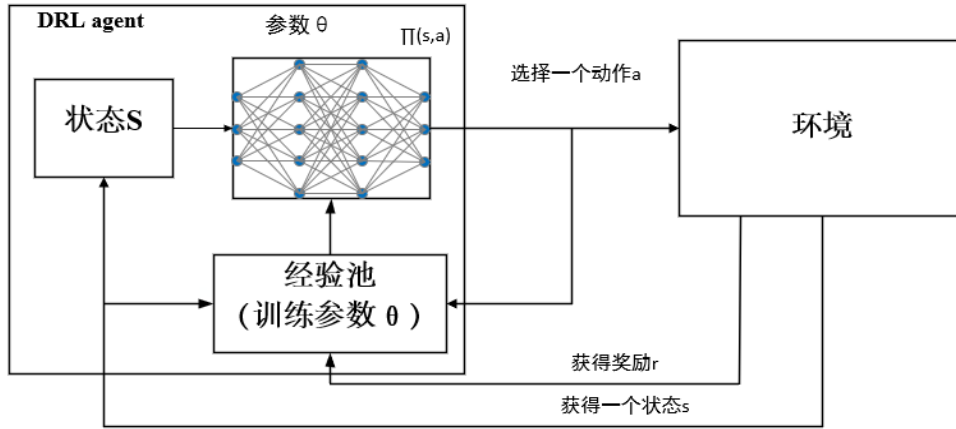


图 2-1 DRL 的框架图

池中随机选取样本，并进行损失函数的计算，公式如下：

$$L(\theta) = \mathbb{E}_{(s,a,r,s') \sim U(B)} [(r + \gamma \max_{a \in A} Q(s', a | \theta') - Q(s, a | \theta))^2] \quad (2-11)$$

神经网络的精度与损失函数成反比，随着损失函数的降低，神经网络的准确性增加。神经网络通过反向传播机制来更新参数 θ ，以最小化损失函数。

2.5 本章小结

本章对移动边缘计算调度优化的相关理论进行简单分析。首先，从 MEC 的定义出发，详细描述其基本概念、核心特性以及与传统云计算的区别。MEC 的特点包括低延迟、实时处理能力。随后，深入分析了 MEC 任务调度问题的必要性，探讨任务调度在提高系统性能、降低延迟和提升用户体验方面的重要性。同时，概述了传统方法在求解任务调度问题中的应用，讨论能耗优化在 MEC 中的关键作用，指出在设计和实施能耗优化策略时需要考虑的因素，如服务器的能源效率、任务的能耗需求、网络的能源管理等。这些方法旨在提高能源效率，

同时满足用户服务质量。最后，介绍了基于深度强化学习的任务调度优化问题，DRL 通过结合深度学习和强化学习的优势，能够在复杂和不确定的环境中学习最优的调度策略。还介绍了 DRL 的基本原理和关键技术，包括状态、动作、奖励等基本概念，以及如何通过学习过程来优化任务调度和能耗管理。此外，还简单说明深度强化学习算法求解调度优化问题的一些基础理论公式和简单计算过程。以便更好的理解 DRL 在 MEC 调度优化和能耗优化方面的研究和应用。

第3章 基于深度强化学习的移动边缘计算任务调度研究

3.1 系统模型

随着新技术的发展，众多应用程序对时间延迟和带宽资源都提出了更高的要求。然而，资源匮乏的移动设备已无法满足这些要求^[36]。为更好的满足这类应用程序，移动边缘计算作为一种新型技术应运而生。通过将动态变化的用户请求调度到 MEC 服务器中进行计算，可以有效缓解设备的资源受限的问题^[37]。MEC 作为一项创新技术，已经提出了多种算法来解决这些调度问题^{[38]-[40]}。在解决应用程序对时间延迟和带宽资源的高需求方面提供了创新性的解决方案。核心策略是通过将动态变化的用户请求的任务调度到 MEC 服务器中计算，从而减少对服务器的资源依赖。在这个过程中，任务调度问题是一个关键性的技术问题，特别是在多用户场景下需要同时考虑时间延迟和带宽资源的限制条件。

具体而言，任务调度问题涉及的是如何有效利用移动边缘服务器的计算资源，以满足用户对低延迟和高带宽的需求。解决这一问题，首先需要考虑时间延迟，确保任务能够在用户请求的时间内得到及时处理。同时，为了避免基站之间的过度拥塞和资源的浪费^[33]，合理分配带宽资源也是至关重要的。

此外，任务调度还需要考虑如何有效地分配用户任务给不同的基站，目的是最大程度地提高基站服务的用户数。这涉及到建立一个用户任务调度模型，考虑用户任务之间的关系、计算需求等因素。针对这一复杂且动态的任务调度模型，提出一种基于深度强化学习的调度算法。深度强化学习算法能够通过学习不同网络环境下的适应性策略^[41]，快速决策用户任务与基站之间的连接，以最大化奖励。通过定义一个考虑用户任务完成时间、带宽利用率、任务调度的效率等多个因素的奖励函数，深度强化学习算法能够自动调整策略，以适应多变的用户需求和网络状态^[34]。总体而言，深度强化学习算法为解决动态任务调度问题提供了有效的解决方案。该方法的优势在于其能够从环境交互中学习，实现智能化的任务调度决策，从而实现基站服务用户数最大化的目标。

3.1.1 任务调度模型

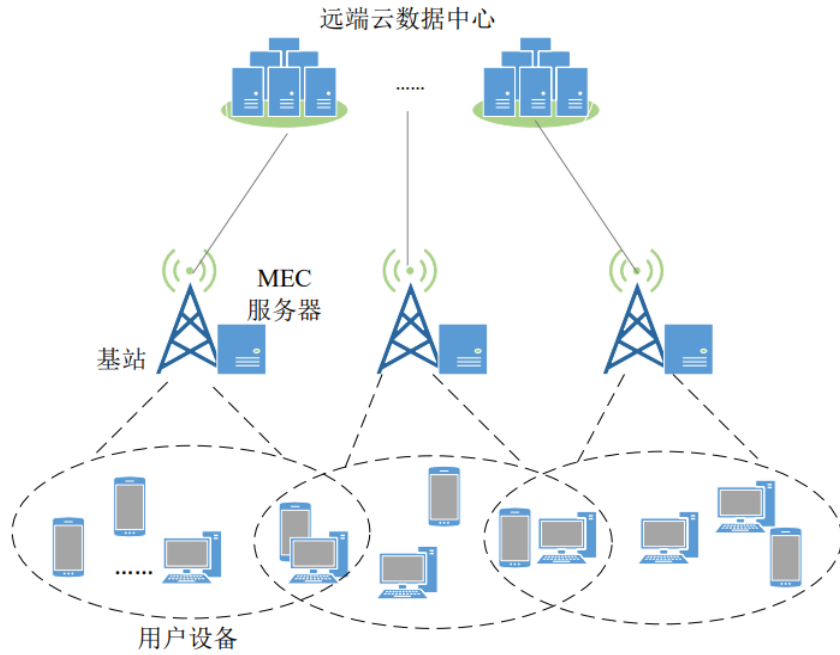


图 3-1 移动边缘计算网络模型

根据上述网络结构模型，在 MEC 网络中部署了很多边缘服务器，还有基站、MEC 服务器和用户设备^[6]。基站与 MEC 服务器一一对应，满足覆盖范围内用户的需求。一开始的时候，所有基站都是处于开放状态。这些基站的集合用 $E \in \{e_1, e_2, e_3 \cdots e_j\}$ 表示，用户的集合用 $U \in \{u_1, u_2, u_3 \cdots u_j\}$ 表示，以及用户请求的服务类型的集合 $S \in \{s_1, s_2, s_3 \cdots s_k\}$ 。但一个用户一次只能连接一个基站，在 MEC 网络中的任务调度过程中，基站主要提供计算资源和带宽资源。计算资源主要用于处理用户请求的任务，通常由服务器消耗。而带宽资源则主要用于基站和用户设备之间数据的传输。这是 MEC 中比较常见的两种资源形式。

3.1.2 带宽资源模型

考虑到实际网络环境的限制，每个基站的计算资源和带宽资源都受到边缘服务器的约束。因此，对于边缘计算服务来说，无线资源的分配变得很重要，尤其是考虑到 MEC 服务输出数据量通常远小于来自用户设备的原始数据量。因此，本章的带宽模型特别注重用户数据传输到服务器时的资源分配。在本章是

假设每个基站可分配的上行总带宽资源为 $m_j^{\max}$ ，将资源分成资源块的形式进行计算^[42]。 $SINR_{i,j}$ 表示基站 j 从每个用户 i 收到的资源块的信噪比，根据自由空间路径损失模型计算得到的信道传输距离相关的信噪比为

$$SINR_{i,j} = 32.44 + 20\log_{10} d_{i,j}(KM) + 20\log_{10} f(MHz) \quad (3-1)$$

上述公式中 $d_{i,j}$ 是表示基站与用户之间的距离， f 表示载波频率。利用香农定理，可计算出用户 i 在基站 j 收到的资源块的数据率，计算公式如下：

$$r_{i,j} = w \log_2(1 + SINR_{i,j}) \quad (3-2)$$

其中 w 表示每个资源块需要的带宽，若 $\lambda_{j,i,k}$ 表示用户 i 从基站 j 到服务 k 所需要的数据率，那么基站 j 在提供服务 k 时分配给用户 i 的总资源块数为

$$m_{i,j,k} = \lambda_{j,i,k} / r_{i,j} \quad (3-3)$$

假设在这段时间里基站服务的用户集合为 U ，则基站为这些用户提供服务所需的带宽总资源数为

$$m_j = \sum_{i \in U} \sum_{k \in S} m_{i,j,k} \quad (3-4)$$

以上是计算带宽资源的过程。

3.1.3 计算资源模型

本文计算资源主要从时间延迟方面来考虑，用时间延迟作为计算资源的消耗。造成 MEC 网络延迟的主要因素有两个：基站在服务用户请求时产生的延迟 $L_j^{service}$ 和网络通信引起的延迟 $L_j^{network}$ 。因此，总延迟由 $L_j^{sum} = L_j^{service} + L_j^{network}$ ^[43] 给出。利用真实世界的环境来提供准确的延迟信息。通过在边缘服务器上重复执行服务，并使用历史数据训练 DRL 代理，经过一定的训练后，可以根据服务器的状态空间预测特定动作的执行时间，并根据其执行时间确定边缘服务器可以服务的用户数量。

3.2 移动边缘计算任务调度问题建模

本章优化的目标是在相同条件下，基站计算资源和带宽资源多种约束限制下，合理制定任务调度策略，使得网络中基站服务用户的数量最大化。因此，移动边缘计算中动态任务调度问题可以转化为每次分配后网络中基站连接的用户总数量。

基站 j ，用户 i ，服务 k ，其中单个服务请求 S_k 的执行时间 $\gamma_{k,j}$ 。 $x_{i,j}$ 表示基站和用户的连接关系， $x_{i,j}=1$ 表示基站 j 与用户 i 连接，为用户 i 提供服务，否则 $x_{i,j}=0$ 。

根据以上分析，边缘计算任务调度的优化问题可以用数学公式表示为：
目标函数和约束条件：

$$\text{Maximize: } \sum_{i=1}^{|U|} \sum_{j=1}^{|E|} x_{i,j} \quad (3-5)$$

$$x_{i,j} = \begin{cases} 1, & \text{if user } u_i \text{ is allocated to server } e_j \\ 0, & \text{otherwise} \end{cases} \quad (3-6)$$

$$d_{i,j} \leq b_j \quad (3-7)$$

$$m_j = \sum_{i \in U} \sum_{k \in S} m_{i,j,k} \leq m_j^{\max} \quad \forall_j \in \{1, \dots, |E|\} \quad (3-8)$$

$$L_j^{nwk} + \sum_{i=1}^{|U|} (x_{i,j} \times \gamma_{k,j}) \leq T \quad \forall_j \in \{1, \dots, |E|\} \quad (3-9)$$

$$\sum_{j=1}^{|E|} x_{i,j} \leq 1 \quad \forall_i \in \{1, \dots, |U|\} \quad (3-10)$$

$$x_{i,j} \in \{0,1\} \quad \forall_i \in \{1, \dots, |U|\}, \forall_j \in \{1, \dots, |E|\} \quad (3-11)$$

公式(3-5)表示目标函数；(3-6)是 $x_{i,j}$ 取值的说明；(3-7)是覆盖半径约束，根据基站覆盖半径 b_j 有限的原则，提供服务的用户必须在覆盖半径内；(3-8)表示带宽约束，一个基站设置的时间阈值和分配的带宽资源不能超过其设定的最大值；公式(3-9)表示时延约束条件；(3-10)表示一个用户只能和一个基站连接；(3-11)表示 $x_{i,j}$ 整数约束。

3.3 移动边缘计算任务调度算法设计

在实际网络环境中，用户和任务的动态变化给任务调度带来了额外的复杂性。使用确定性的任务调度方法可能会导致对边缘服务器的用户过度或不足的分配。本章的目的是制定任务调度策略，这些策略是在严格遵守基站服务用户过程中的时间延迟阈值 T 和基站带宽资源约束 M 的同时，最大化边缘服务器服务的用户数。因此，必须根据基站和用户请求服务的状态动态决定如何进行任务调度。

在任务调度决策的过程中，主要挑战在于每个边缘服务器的资源会受到限制，服务的用户数也因此受到限制。此外，用户的移动性和多基站覆盖范围的重叠也无形中增加任务调度的复杂性。为解决这些难点，本章提出基于深度强化学习的方法。这种方法通过从边缘服务器的经验中学习服务执行模式，以获得同时符合时间延迟阈值和带宽资源约束的用户任务和基站服务器的绑定决策。在深度强化学习框架中，代理通过不断探索学习并选择更优的行动，以最大化从环境中获得的奖励，如图 3-2，强化学习的主要过程图解。

在深度强化学习框架中，状态由边缘服务器的资源向量以及服务请求的数量表示，这反映当前系统的运行状况。动作则表示等待在边缘服务器上执行的一组服务请求，即任务调度的决策。代理通过在不同状态下采取不同的动作，并通过观察奖励来更新策略，强化学习代理能够逐步学习到在复杂、动态网络环境下的最佳任务调度决策。

这种方法的优势在于其灵活性和适应性，它能够适应实时变化的网络条件和用户需求。通过不断优化任务调度决策，系统能够在保持服务质量的同时，优化边缘服务器的资源利用率，实现最大化服务的用户数量的目标。使用马尔可夫决策过程对需要解决的问题进行建模。

状态空间 Σ ：状态空间由 MEC、用户及其所在环境共同决定，由六个属性表示，分别为：(R :RAM(MB), C :内核数, CW :CPU 后台工作负载, GU :GPU 利用率, $Ns1$: 服务 $s1$ 的服务请求数量, Nsk : 服务请求数 sk)。

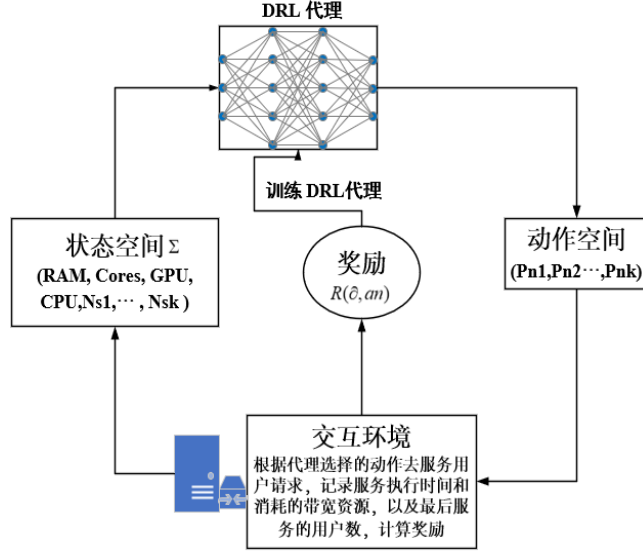


图 3-2 强化学习图解

行动空间 A ：边缘服务器上执行的用户服务请求数表示的一组操作。

$a_n \in A$ 由一个元组表示 $(Pn1, Pn2, \dots, Pnk)$ 其中 k 是边缘服务器上托管服务总数， Pnk 表示要在边缘服务器上执行的服务 sk 的服务请求的数量。

奖励函数 $R(\partial, an)$ ：($an \in A$ at state $\partial \in \Sigma$) 强化学习代理根据边缘服务器状态空间选择动作的过程，产生的总时间延迟和带宽资源的消耗。在边缘服务器 e_j 上，给定一个状态 $\partial = (R, C, CW, GU, Ns1, \dots, Nsk) \in \Sigma$ 和一个动作 $an = (Pn1, Pn2, \dots, Pnk) \in A$ ，提供 k 个服务。在时间延迟阈值和带宽资源限制下，则对应的状态动作获得的奖励为所容纳服务的总和，乘以阻尼因子 η 。此处如果所选的用户任务无法服务，奖励则为零。形式上

$$R_{lin}(\partial, an) = \begin{cases} \eta \sum_{i=1}^k Pni & \text{如果时间延迟 } L < T \text{ 带宽 } m < M \\ 0 & \text{其他} \end{cases} \quad (3-12)$$

在 MDP 中，代理学习最佳动作，即边缘服务器上可部署的服务请求的最大数量； L_j^{sum} 不超过 T ，其中 $L_j^{sum} = L_j^{service} + L_j^{network}$ ，服务执行时间 $L_j^{service}$ 和网络延迟 $L_j^{network}$ [43]；边缘服务器 e_j 的网络延迟 $L_j^{network}$ 最大可能是由于网络通信延迟而产生的延迟。在延迟阈值 T 和带宽资源阈值 M 下执行的服务请求数越

多的动作，返回的奖励值就越高，然而，如果选择的动作未满足服务延迟 T 和带宽资源 M ，则奖励值越低。代理人通过探索和利用环境来学习最佳行动^[44]。调度问题运行模拟的环境数据是真实数据^[45]，是由澳大利亚墨尔本市中央商务区(CBD)的用户和边缘服务器的位置组成，它提供了系统的真实延迟 L_j^{sum} 和状态。通过在边缘服务器 e_j 上执行多个服务，可以直接从系统中获得由特定动作产生的延迟； L_j^{sum} 用于返回边缘服务器 e_j 上状态为 ∂ 的操作的奖励 $R(\partial, an)$ 。

给定一个边缘服务器 e_j 上的初始资源状态向量，该特定边缘服务器上单个服务请求 Sk 的执行时间 $\gamma_{k,j}$ 通常通过系统对历史服务执行数据进行学习然后计算确定。根据服务执行时间可以确定边缘服务器服务的用户数量。通过向神经网络输入状态 $\partial = (R, C, CW, GU, Ns1, \dots, Nsk) \in \Sigma$ ，输入到神经网络中，计算 Q 值。计算过程主要由 DRL agent 完成，它使用深度 Q 学习范式进行训练。根据方程式(3-13)，计算每个状态下可选动作的 Q 值^[44]。选择与该状态下的最大 Q 值相对应的动作，该动作是具有最大服务数量的元组。这里 α 是学习率， β 是折扣系数。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \{R(s_t, a_t) + \beta \max_{a_{t+1} \in A} [Q(s_{t+1}, a_{t+1})] - Q(s_t, a_t)\} \quad (3-13)$$

在 MDP 中，代理的学习目标是通过选择最佳动作来最大化边缘服务器上的服务请求数量。使用真实系统获取的数据集，代理通过探索和利用这些数据集环境来不断学习，挑选出每个状态下的最佳行动。使用强化学习算法求解用户任务调度的目标是最大化服务器服务的用户数。

算法 3-1，描述的是基于 DRL 算法求解任务调度的整个过程。初始化基站边缘服务器的资源，每个基站根据边缘服务器的资源状态，输入到经过训练 DRL 后代理获得当前状态下该基站可以服务的用户请求数量元组的预测。对于每个用户，可能被多个基站覆盖，用户从被覆盖的多个基站服务器中选择一个用户请求数量元组预测值最大的基站服务器，将其分配给该基站。通过这样的方式完成任务调度问题，最终获得基站和用户的连接列表。

算法 3-1: DRL 任务调度算法**Input:** U —用户, E —服务器, DRL 代理**Output:** 返回用户-服务器调度连接列表 X

```

1 初始化 DRL 代理的神经网络, 用户和服务器的连接状态, 设置训练步
   数、带宽资源限制和时延阈值
2 for  $e \in E$  do
3    $S$ —单个边缘服务器  $e$  的状态
4    $U_e[k]$ —用于预测边缘服务器  $e$  服务的用户数量的元组
5   给定边缘服务器  $e$  的状态  $\partial \in \Sigma$ , 经过训练的 DRL 代理获得预测的
      服务请求
6 end
7 for  $u \in U$  do
8    $e_{list}$ —覆盖用户  $u$  的服务器列表
9    $e_{chosen}$ —在  $e_{list}$  列表中选择具有最大  $U_e[k]$  元组的边缘服务器  $e$ 
10   $X \leftarrow (u, e_{chosen})$  将用户  $u$  分配给所选的边缘服务器  $e_{chosen}$ 
11  对于一个边缘服务器的待服务的用户请求状态  $S$  减少 1
12 end

```

3.4 仿真实验结果分析

3.4.1 仿真实验参数设置

本章中使用的实验平台是在 Window10 系统上的 Anaconda3(64bit)中 python10.0 版本。实验中边缘服务器上部署的服务类型以 YOLO 和 MobileNetV2 为对象进行仿真实验。YOLO^[46]和 MobileNetV2^[47]服务广泛用于图像和视频中的对象检测。这两者都使用 CPU 和 GPU 进行计算, 应用程序的计算量相对比较大。因此, 上述服务被用作本文 MEC 环境中的代表性服务。实验中使用软件库 Stable-Baseline3^[48]进行强化学习代理的培训。最后, 通过比较边缘服务器容纳的用户数量, 证明了该方法的优势。

实验中的一些参数设置如表 3-1 所示。(r[yolo]和 r[mnet]分别表示用户 i 向基

站请求 yolo 和 mobileNetV2 服务所需的数据速率)。本文使用文献[45]中的边缘服务器位置数据集, 边缘服务器的覆盖区域设置为 150 米的半径。奖励函数中的因子设置为 $\eta=0.01$, 学习率设置为 $\alpha = 0.0001$ 。

表 3-1 实验中一些参数符号及数值

参数	参数值
内核 RAM	3000-11000MB
载波频率(f)	2 GHz
r[yolo]	1.3Mbps
r[mnet]	5Mbps
每个资源块的带宽 w	5kHz
CPU 的核数	2-5
初始 GPU 利用率	1-10%
初始 CPU 的工作量	40-60%
阻尼因子 η	0.01
神经网络学习率 α	0.0001
衰减系数 β	1
时间步数	30000
用户数量	150-400
基站数量	30-100
延迟阈值 T	20-50s
带宽资源块阈值 M	10-20,50

3.4.2 实验结果分析

本章在验证基于深度强化学习的移动边缘计算任务调度算法的有效性的同时, 也与其他相关算法进行了比较。

(1) ILP——确定性整数线性规划方法。ILP 模型是一个传统的分配方法, 在一定的服务执行时间下运行。给定边缘服务器的初始资源状态, 通过历史服务执行时间数据的平均值来计算单个用户在服务器上请求服务的执行时间, 最后计算出基站服务的用户数量^[52]。

(2) Greedy——基于最近邻分配原理的简单贪婪启发式方法。Greedy 方法的主要思想是将请求服务的用户分配给具有最近剩余资源的边缘服务器, 并在同时满足时间阈值和带宽约束时向用户提供服务, 最后得到基站服务的用户数量^[56]。

仿真实验初始化默认总共有 400 个用户, 边缘服务器的数量为 30-100, 带

宽阈值为 50，延迟阈值为 20 秒。进行了重复实验，比较了不同带宽资源和时延阈值对用户任务调度结果的影响。还比较了每个实验的训练损失、平均回报、分配的用户数量以及不同方法的执行时间。

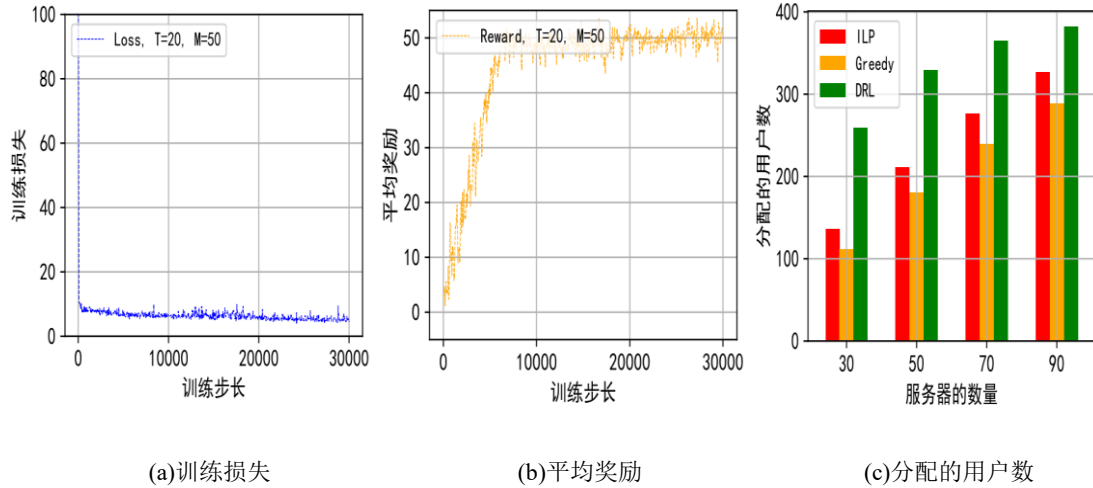


图 3-3 $T=20s$, $M=50$ 下仿真结果图

图 3-3 显示了在边缘服务器的数量为 30-100，带宽阈值为 50，延迟阈值为 20 秒的条件下，训练损失、平均奖励以及与基准方法对比的分配用户数的结果图。从这些图中还可以看出，在相同配置条件下，DRL 在用户任务调度分配方面具有一定的优势。

下面是任务调度问题在不同带宽和时延约束下的结果对比图。

1) 不同延迟阈值和边缘服务器数量下的比较

仿真中设置用户数为 400，带宽资源为 50，时延阈值为 20-50s，边缘服务器数量为 30-100。从图 3-4 中，可以看到这三种方法在用户任务调度中的性能。当时延阈值相对较低时，DRL 方法获得的结果与其他确定性方法显著不同。随着时间阈值的增加，三种方法分配的用户数量之间的差距逐渐缩小。确定性方法很难在严格的延迟内完成用户分配。在网络中，随着服务器的增加，分配用户变得更加容易。在这种情况下，确定性方法等效于 DRL 方法。

2) 不同延迟阈值和用户数量下的比较

仿真中设置边缘服务器的数量为 50，带宽资源为 50，时延阈值为 20-50s，用户数量为 150-400。从图 3-5 中可以看出，在延迟阈值相对较低的情况下，DRL 方法调度分配的结果更好，不会随着延迟阈值的变化而波动很大，更稳定。随着时间阈值的增加，三种方法分配的用户数量之间的差距逐渐缩小。随着用

户数量的增加，DRL 和确定性方法调度的结果是可比较的，差距越明显。DRL

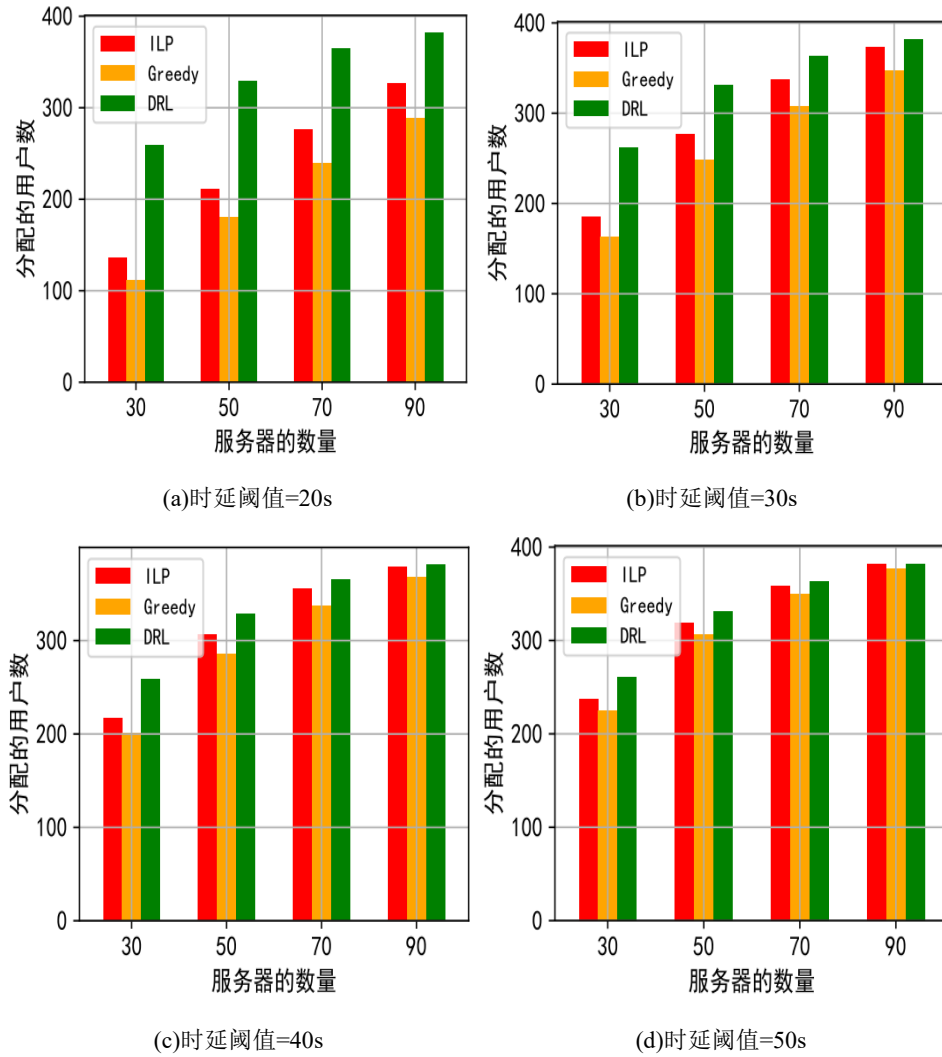


图 3-4 不同延迟阈值和边缘服务器数量下的比较

在更严格的延迟约束下表现更好。

3) 不同带宽和边缘服务器数量下的比较

不同带宽和边缘服务器数量下的比较，设置用户数量为 400，时间阈值为 40s，带宽阈值为 10-20，边缘服务器为 30-100。从图 3-6 可以看出，在带宽资源约束严格的情况下，DRL 与其他两种方法相比有很大的差距。随着带宽资源的增加，这种差距逐渐缩小。DRL 方法在带宽资源严格限制和时延相对宽松的情况下显示出良好的优势。但是，在一定程度上，增加带宽资源之间并没有明显的差距，因为，在这种调度分配过程中，它不仅受到带宽资源的限制，还受到延迟阈值的约束。

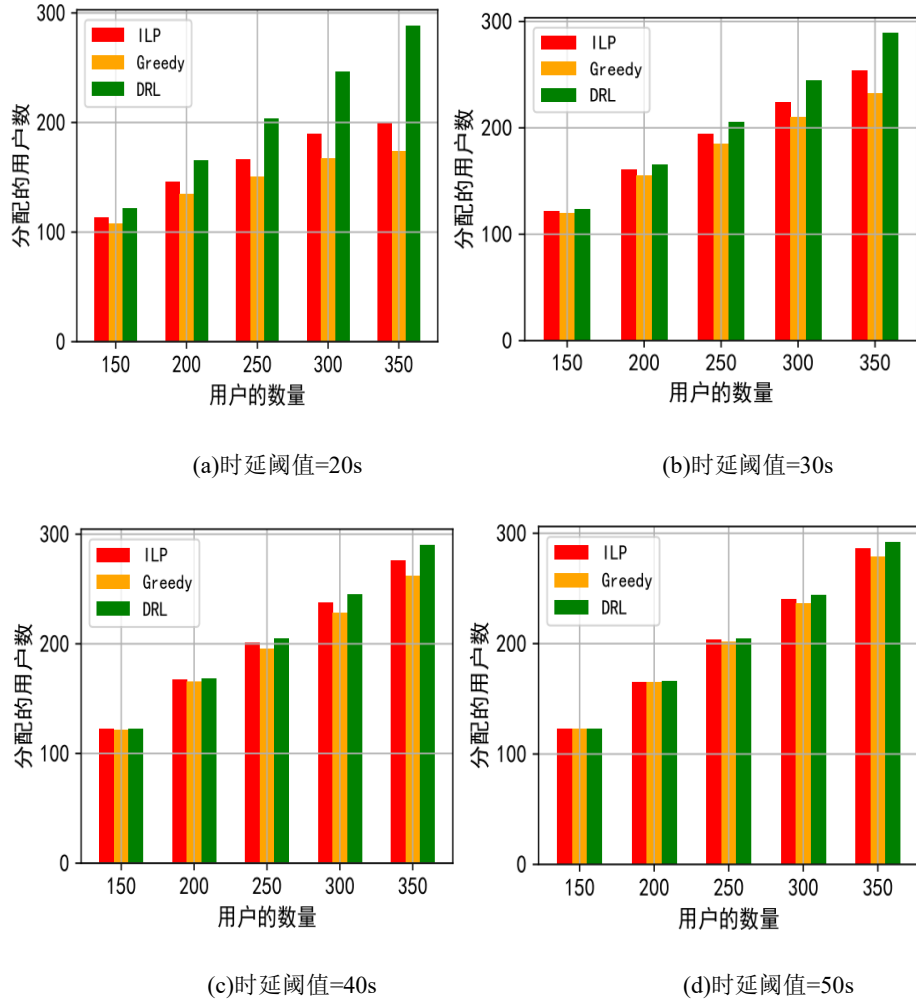


图 3-5 不同延迟阈值和用户数量下的比较

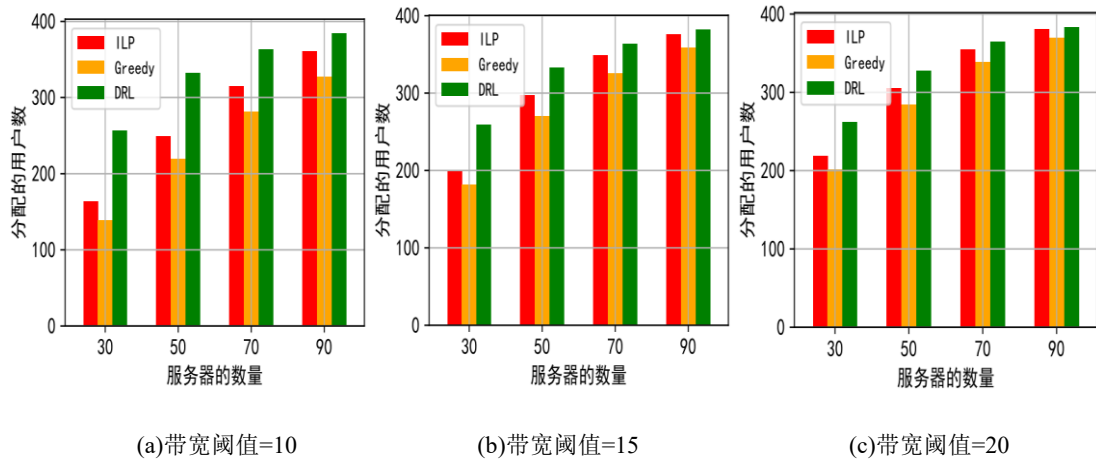


图 3-6 不同带宽和边缘服务器数量下的比较

4) 不同带宽和用户数量下的比较

仿真实验中设置边缘服务器的数量为 50，时间阈值为 40 秒，带宽阈值为 10-20，用户数量为 150-400。从图 3-7 中可以看出，当用户数量相对较大时，

DRL 方法比其他两种方法分配更多的用户。当用户数量相对较少时，三种方法的结果有差距。随着带宽资源的增加，三种方法之间的差距逐渐缩小。由于网络中的用户数量较少，可分配的用户很少，因此分配结果之间的差距不是很明显。

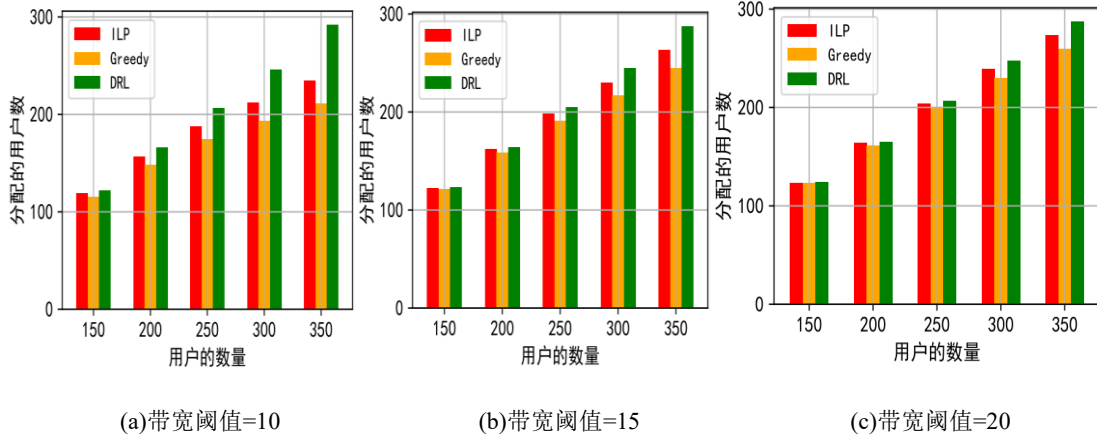


图 3-7 不同带宽和用户数量下的比较

上述仿真实验中，实验 1)和 2)时延阈值约束起主导作用，所以在设置带宽阈值时选择相对宽松的阈值 $M=50$ ；实验 3)和 4)带宽阈值的限制约束起了主导作用，所以时延阈值就选择相对宽松的 $T=40s$ 。

基于上述实验结果，DRL 方法在求解多约束任务调度方面具有一定的优势。

3.5 本章小结

在本章中，使用了 DRL 方法来解决移动边缘计算环境中各种约束条件下用户任务的调度问题。这种方法克服了传统方法只能对确定的服务执行时间进行建模的局限性。考虑到实际网络中计算和带宽资源的限制，使用真实世界的数据集和服务执行数据进行实验，在边缘服务器上执行服务来自动推断资源利用关系，并在严格遵守时延阈值和用户消耗的带宽资源约束的情况下将请求服务的用户分配到边缘服务器。实验结果表明，提出的基于 DRL 的方法优于传统的确定性方法。

第4章 基于深度强化学习的移动边缘计算能耗优化研究

4.1 系统模型

在第3章中，深入讨论了如何利用深度强化学习方法解决 MEC 中的任务调度问题。在实际网络环境中，用户请求服务的特性是动态变化的，考虑网络中用户任务请求的随机性，并提出一种优化基站用户连接的分配策略，以使每个基站在有限的计算资源和带宽资源内服务更多的用户。

在这个过程中，注意到在某些时刻，即网络中的用户请求服务的低谷时期。如果所有基站都一直处于开启状态，这可能导致一些基站资源的浪费。因此，在本章中，考虑在网络中用户流量较少的时候，将用户集中到较少的基站上，利用最少的边缘服务器满足这些用户的需求。通过关闭一些基站服务器，可以有效减少能耗，从而实现能耗的优化目标。通过在低流量时段关闭一些基站，系统可以更加灵活地适应实际的用户需求，减少资源浪费并最大化满足用户需求。

在移动边缘计算的背景下，深度强化学习成为了一个有效的工具，它能够适应动态变化的网络环境，并智能地调整边缘服务器的开关状态。通过代理与环境的交互学习，DRL 算法能够自适应地调整策略。以确保在用户请求服务的低谷期间最大程度地降低能耗。代理根据当前环境的状态，包括用户数量、流量状况、基站资源参数和基站的资源利用率等因素，来决策是否关闭某些基站，以达到能耗最优化的目标。最终实现了可以根据网络中用户数量的多少动态调整边缘服务器的开关。

总的来说，本章综合考虑用户随机性、网络流量动态性以及能耗优化的需求，提出了一种基于 DRL 的方法，以实现移动边缘计算边缘服务器的智能开关。这种根据网络流量动态调整边缘服务器开关状态的策略可以在实际网络中有效解决能耗优化问题。

4.1.1 能耗优化模型

结合实际应用场景，对边缘计算网路中能耗资源的计算，只考虑用户和基站之间的信息传输的传输能耗、基站服务用户的实际计算能耗以及关闭基站用户切换产生的能耗这三部分组成^[6]。具体的能耗模型如下：

(1) 基站传输能耗 p_j^b, p_j^b 主要包括两部分，第一部分是基础能耗 $p_j^{b,op}$ ，基站运行会有基础能耗，转换器，冷却器所产生的能耗称为基本能耗，一般设为固定值；另一部分是实际传输过程所产生能耗 $p_j^{b,tx}$ ，根据第 3 章的公式(3-4)可以计算出每个基站的实际传输能耗：

$$p_j^{b,tx} = \rho^{-1} p_j^{tx} m_j \quad (4-1)$$

其中 ρ 表示功率放大器能效系数，影响基站的传输能耗。基站 j 接收的资源块的传输能耗用 p_j^{tx} 来表示。基站的传输能耗可通过以下公式表示：

$$p_j^b = p_j^{b,op} + p_j^{b,tx} \quad (4-2)$$

(2) 基站计算能耗 p_j^s ，计算能耗主要是有两部分组成，第一部分 $p_j^{s,op}$ 是服务器公共基础的计算能耗，一般为固定值。第二部分 $p_j^{s,tx}$ 是服务器为用户提供任务服务所产生的实际计算能耗。其中每个基站服务器最大的计算容量 $c_j^{\max}$ ；基站处理每个服务 k 所消耗的基础能耗 c_k^b ，基站服务器处理用户 i 的计算任务所需的资源 c_i ，基站 j 在这个状态下对请求服务 k 的用户所需要的资源可以表示为：

$$c_j^k = c_k^b + \sum_{i \in U_l} a_{i,k} c_i \quad (4-3)$$

其中 $a_{i,k}$ 表示用户 i 此时请求的服务类型是否为 k ， $a_{i,k} = 1$ 表示用户 i 在这个状态下请求的服务类型为 k ；相反， $a_{i,k} = 0$ 表示用户 i 在这个状态下请求的服务类型不是 k 。因此，基站 j 的边缘服务器在这个状态下提供的总计算资源可以表示为：

$$c_j = \sum_{k \in S} c_{j,k} \quad (4-4)$$

基站 j 的边缘服务器的最大计算功率用 $p_j^{s,\max}$ 表示，这样就可以计算出对应基站边缘服务器的实际计算功率：

$$p_j^{s,tx} = (p_j^{s,\max} - p_j^{s,op}) \frac{c_j}{c_j^{\max}} \quad (4-5)$$

因此，基站 j 的计算能耗：

$$p_j^s = a_j^s (p_j^{s,op} + p_j^{s,tx}) \quad (4-6)$$

其中 a_j^s 表示基站 j 的服务器的开关状态， $a_j^s = 1$ 表示在这个状态下基站 j 是开启状态， $a_j^s = 0$ 则表示基站 j 是关闭状态，关闭状态下是不提供服务。

(3) 关闭基站用户切换产生的切换能耗 p_j^{sw} ， a_j^s 表示在这个状态下基站 j 的服务器的开关状态， $a_j^s = 1$ 表示基站 j 是开启状态， $a_j^s = 0$ 则表示基站 j 是关闭状态； $a_j^{s-plus_one}$ 表示在下一状态下基站 j 对应的服务器的开关状态， $a_j^{s-plus_one} = 1$ 表示下一状态下基站 j 对应的服务器是开启状态， $a_j^{s-plus_one} = 0$ 则表示下一状态下基站 j 对应的服务器是关闭状态。在网络中用户和基站都是动态变化的，每次变化都称为一个状态。每个状态的切换用户不相同，因此切换能耗也不相同，切换能耗可以表示为，

$$p_j^{sw} = \begin{cases} \psi |U_j^s| & a_j^s = 1, a_j^{s-plus_one} = 0 \\ \psi |U_j^{s-plus_one}| & a_j^s = 0, a_j^{s-plus_one} = 1 \\ 0 & a_j^s = a_j^{s-plus_one} \end{cases} \quad (4-7)$$

其中 ψ 表示当个用户的切换能耗， $|U_j^s|$ 表示基站 j 在状态 s 时连接的用户集合， $U_j^{s-plus_one}$ 表示基站 j 在状态 s_plus_one 时连接的用户集合。

因此，网络中能耗模型公式如下：

$$p_j = p_j^b + p_j^s + |a_j^{s-plus_one} - a_j^s| p_j^{sw} \quad (4-8)$$

本章还涉及到一个基站资源利用率的计算，公式如下：

$$\theta_j = \frac{c_j}{c_j^{\max}} \quad (4-9)$$

c_j 表示基站 j 的边缘服务器在这个状态下为用户提供服务所消耗的总计算资源， $c_j^{\max}$ 表示基站 j 的边缘服务器的资源。

4.2 边缘计算网络能耗优化问题建模

根据上述的分析，为了有效地解决网络中的能耗问题，首先需要深入了解整个网络结构中的能耗模型。能耗模型涉及到每个组成部分的功耗特性，包括边缘服务器、基站设备等。理解这些信息可为建立合适的马尔可夫决策过程提供基础。只有充分理解整个网络结构中的能耗模型，才能更好的设置马尔可夫决策过程的模型参数，以便使用户和基站更好的连接，以及决策哪些 MEC 基站开放服务用户。通过这种方式，可以确保网络在满足用户服务质量要求的同时，实现能耗的最优化。

基站 j ，用户 i ，服务 k ，其中单个服务请求 Sk 的执行时间 $\gamma_{k,j}$ 。 $x_{i,j}$ 表示基站和用户的连接关系， $x_{i,j}=1$ 表示基站 j 与用户 i 连接，为用户 i 提供服务，否则 $x_{i,j}=0$ 。

根据以上分析，移动边缘计算中多重能耗资源优化问题可以用数学公式表示为：

目标函数：

$$\min \sum_{j=1}^{|E|} p_j \quad (4-10)$$

$$d_{i,j} \leq b_j \quad (4-11)$$

$$m_j = \sum_{i \in U} \sum_{k \in S} m_{i,j,k} \leq m_j^{\max} \quad \forall_j \in \{1, \dots, |E|\} \quad (4-12)$$

$$L_j^{nwk} + \sum_{i=1}^{|U|} (x_{i,j} \times \gamma_{k,j}) \leq T \quad \forall_j \in \{1, \dots, |E|\} \quad (4-13)$$

$$\sum_{j=1}^{|E|} x_{i,j} \leq 1 \quad \forall_i \in \{1, \dots, |U|\} \quad (4-14)$$

$$x_{i,j} \in \{0,1\} \quad \forall_i \in \{1, \dots, |U|\}, \forall_j \in \{1, \dots, |E|\} \quad (4-15)$$

在第 3 章的基础上进行能耗优化，所以任务调度的约束都需要满足。因此，根据基站覆盖半径 b_j 有限的原则，提供服务的用户必须在覆盖半径内；且一个基站设置的时间阈值和分配的带宽资源不能超过其设定的最大值；公式(4-10)表示目标函数；公式(4-11)表示的覆盖半径约束条件；公式(4-12)表示带宽资源约束条件；公式(4-13)表示边缘服务器的时间延迟约束；(4-14)表示一个用户只能和一个基站连接；(4-15)表示 $x_{i,j}$ 只能取整数。

4.3 基于深度强化学习移动边缘计算能耗优化算法设计

在实际的任务分配过程中，传统的分配策略可能会将用户过度或不足地分配给边缘服务器。为解决这个问题，本章提出一个基于 DRL 的学习框架，该框架旨在通过从边缘服务器上的经验中学习服务执行模式来获取更好的分配策略。在这个 DRL 框架中，强化学习代理充当学习者的角色，通过与环境交互来不断学习。代理观察边缘服务器在不同条件下的运行状态和任务分配情况，执行相应的动作，并根据获得的奖励来调整自身的策略。通过这种学习过程代理可以发现对于特定条件下哪些策略能够最大程度地满足用户需求并优化系统性能。

DRL 框架中，代理通过探索环境来学习环境以选择更好的行动，并从动作接收反馈。代理持续地与边缘服务器交互以采取行动，即执行若干服务请求，并根据执行足迹获得相应的奖励。通过这种方式，代理能够识别并优化资源的配置，以实现动态调整资源利用率较低的边缘服务器的开关状态，从而减少能耗浪费。

算法 4-1 详细描述了深度强化学习如何通过动态关闭一些基站来减少网络总能耗。具体过程如下：

算法 4-1: DRL 能耗优化算法

Input: U — 用户, E — 服务器, 基站 j 与用户 i 之间的连接状态, 基站当前的开关状态 a_j^s 与下一开关状态 $a_j^{s-plus-one}$, DRL 代理

Output: 返回用户-服务器调度连接列表 X

1 初始化 DRL 代理的神经网络, 用户和服务器的连接状态, 设置基站开启的数量、网络中用户的变化范围、训练步数、带宽资源限制和

时延阈值

```

2  while true do
3    for  $j \in E$  do
4      用资源利用率公式计算每个状态下的边缘服务器  $j$  的利用率  $\theta_j$ 
5    end
6    while true do
7      for  $j \in E$  do
8        从上述计算的  $\theta_j$  中挑选出利用率最低的服务器并将其标记为
           $j^*$ 
9      end
10     利用能耗公式计算此时网络中总能耗  $\Phi$ 
11     更新标记基站  $j^*$  的状态后, 代入并运行 3-1 算法, 获取重新分
        配的连接状态  $X_c$ 
12     for  $j \in E$  do
13       用资源利用率公式计算每个状态下的边缘服务器  $j$  的利用率
           $\theta_j$ 
14     end
15     while true do
16       for  $j \in E$  do
17         从上述计算的  $\theta_j$  中挑选出利用率最低的服务器并将其标记为
             $j^*$ 
18       end
19       利用能耗公式计算此时网络中总能耗  $\Phi_c$ 
20       if 更新前 3-1 算法运行结果用户数小于等于更新后的用户
          数, 并且计算的  $\Phi_c \leq \Phi$  成立 then
21         保留更新后的基站服务器的状态
22       else
23         Break
24       end

```

25	end
26	end
27	end

在算法 4-1 中，首先利用第 3 章中的算法来获取初始条件下用户和基站之间的连接关系。这是很重要的一步，在后续能耗优化工作中提供基础数据。接下来，在这个基础上，利用公式(4-9)计算了每个基站在每个状态下的资源利用率。这一计算过程包括基站在服务用户请求、空闲等不同状态下的资源利用情况。通过这些计算，可以获得每个基站在不同运行状态下的资源利用率。然后，根据上一步计算的基站资源利用率，从其中挑选出每个基站资源利用率最低的状态，这些利用率最低的基站会被标记，用符号 j^* 来表示。这为后续的优化策略提供了重要的参考依据，以更好地适应动态变化的网络条件和用户需求。

同时，使用能耗计算公式(4-8)计算当前网络的总能耗，并将其标记为 Φ 。计算完当前状态下的网络能耗后，将前面标记的基站 j^* 的开关状态动态调整为关闭。这一步主要是模拟关闭资源利用率相对较低的基站，评估关闭该基站对系统能耗的影响。更新基站的开关状态后，将新的基站开关状态输入到算法 3-1 中，以进行用户和基站连接的重新分配。

再次利用公式(4-9)计算更新后的每个基站的资源利用率，并在其中挑选出基站资源利用率最低的状态，标记为 j^* ，同时利用公式(4-8)计算更新后的网络总能耗，并将其标记为 Φ_c 。

此时比较 Φ 与 Φ_c 的大小，以及更新前后算法 3-1 获取的分配结果中，网络中基站服务器服务的用户数的大小。当 $\Phi_c \leq \Phi$ ，且更新后的基站服务器服务的用户数大于等于更新前的用户数时，才满足条件，将继续循环。否则就跳出循环，将更新前的状态，能耗作为优化后最终的结果。

4.4 仿真实验结果分析

4.4.1 仿真实验参数设置

本章中使用的实验平台是在 Window10 系统上的 Anaconda3(64bit)中 python10.0 版本。实验中使用软件库 Stable-Baseline3^[45]进行 DRL 代理的培训。最后，通过比较每种方法计算的能耗，证明了该方法的优点。

实验中的一些参数设置如表 4-1 所示。使用第 3 章算法运行的用户分配的结果，计算网络中的基站利用率以及网络总能耗。该网络部署了 80 个边缘基站，边缘服务器的覆盖区域设置为 150 米的半径。该网络中每个基站配置数据是一样的，每个基站都有一个 MEC 服务器与之对应，基站与基站之间的距离为 300 米。其他的参数和第 3 章一样。

表 4-1 实验参数

参数	值
每个基站的最大时延	1s
每台边缘服务器最大的 CPU 频率	10(Megacycles/second)
MEC 服务请求上传的数据包大小	250bits - 1500bits
每台边缘服务器的计算资源(标准化)	1
每个用户的传输功率	10dBm
边缘服务器硬件系统相关常量(μ)	0.01
每个用户消耗的切换功率 Ψ	4W
每个基站的基础功率	100W
每个资源块的传输功率	0.2W
功率放大器效率 ρ	50%
每台边缘服务器的基础功率	60W
每台边缘服务器最大的计算功率	200W
用户数量	150-300
基站数量	30-100

4.4.2 仿真实验结果分析

在本章的实验中，主要考虑 DRL 方法计算的总能耗，设置了不同的网络用户数量，对比网络总能耗的变化。与其他算法计算的结果进行对比验证算法的有效性，对比算法与第 3 章的算法一致，确定性整数线性规划方法(ILP)和基于最近邻分配原理的简单贪婪启发式方法(Greedy)。因此，在本章的实验中，

设置模拟了以上三种方法分别计算的总网络能耗的结果。

1) 不同用户数量下的能耗比较

仿真实验中设置边缘服务器的数量为 80，带宽资源为 50，时延阈值为 30s，用户数为 150-250。从图 4-1 中可以看出，在分配的用户数量差别不大的时候，可以在能耗图中看出此时深度强化学习表现出较好的优势，与确定性方法计算的总网络能耗差距明显，能耗更低。这种优势的体现可能源自于深度强化学习算法在网络资源管理方面的灵活性和智能性。深度强化学习能够根据当前的网络状态和环境情况动态调整决策，使得网络资源的分配更加高效和灵活。相比之下，基于确定性方法可能会受到网络状态变化的限制，难以适应复杂多变的网络环境。

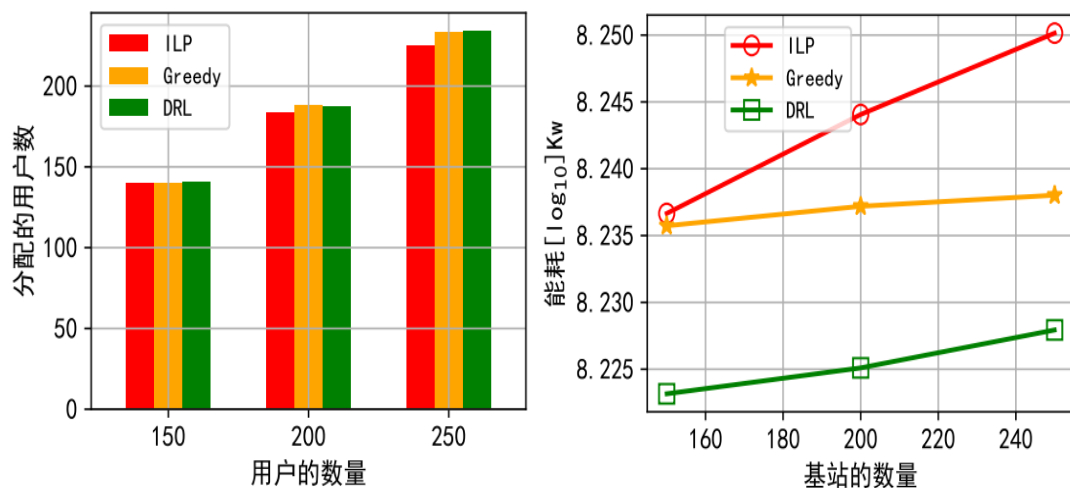


图 4-1 不同用户数下的能耗和分配的用户数对比图

2) 不同边缘服务器数量下的能耗比较

仿真实验中设置用户数为 300，带宽资源为 50，时延阈值为 30s，边缘服务器数量为 30-70。从图 4-2 中可以看出，在分配的用户数相近的情况下，深度强化学习展现出了显著的优势。特别是在这一范围内，深度强化学习算法表现出了较低的网络能耗，相较于其他方法，其优势更为明显。以最大程度地降低网络能耗，同时保证用户服务质量。其他方法可能会受到网络资源分配的限制或者无法适应网络环境的变化，从而导致网络能耗较高。

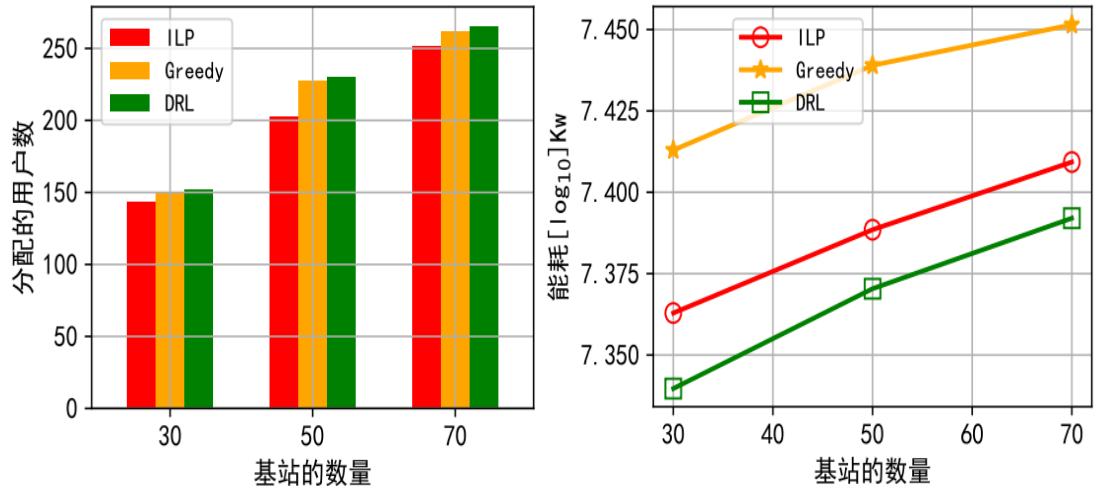


图 4-2 不同基站数量下的能耗和分配的用户数对比图

因此，通过在仿真实验中观察深度强化学习算法的表现，可以得出结论：在移动边缘计算环境中，采用深度强化学习进行任务调度和能耗优化是一种有效的策略，能够显著降低网络的总能耗，提高系统的性能和效率。

4.5 本章小结

在本章中，使用 DRL 方法解决移动边缘计算环境中多重基站能耗优化问题。在网络流量较少时，将这些较少的用户集中到部分基站上进行计算服务。根据每次迭代过程中基站资源利用率的高低来尝试关闭一台资源利用率最低的基站，以优化基站能耗问题。考虑到实际网络中计算和带宽资源的限制，使用真实世界的数据集和服务执行数据进行实验。利用 DRL 算法，可以在严格遵守时延阈值和用户消耗的带宽资源约束的情况下将用户分配给基站服务器，并在分配的过程中学习网络状态和用户请求，不断优化任务分配策略，以减少网络能耗。实验结果表明，深度强化学习方法在降低基站总能耗方面优于传统的确定性方法。

第5章 结论与展望

5.1 结论

移动边缘计算作为 5G 网络的核心技术之一，通过将计算和存储资源部署于网络边缘，即接近用户的位置，有效缩短了数据传输的路径。这一策略提升了移动通信系统在数据传输延迟和带宽利用方面的表现，并为物联网设备执行计算密集型和对延迟敏感的应用提供了支持。边缘服务器所提供的服务，加快了数据处理速度，减轻了网络拥堵，从而为用户带来更迅速、更稳定的服务体验。在 MEC 系统中，任务调度和能耗优化是两个关键问题，它们对于提升系统整体性能至关重要。本文针对 MEC 中的任务调度和能耗优化问题进行了深入研究，提出了一种基于深度强化学习的解决方案。本文首先将任务调度问题建模为马尔科夫决策过程，并运用 DRL 算法进行有效求解。在此基础上，进一步考虑了网络能耗问题，并提出了一种动态调整边缘服务器运行状态的策略。该策略依据实时网络流量情况，智能决策关闭部分边缘服务器，以优化能耗。本文的主要研究工作总结如下：

1) 对于移动边缘计算环境中存在的多种约束条件，本文提出了一种新的用户任务调度方法。该方法充分考虑了边缘服务器的带宽和计算资源限制，并利用 DRL 算法预测在不同状态下，各个边缘服务器能够服务的最大用户数。基于这些预测，算法能够为用户分配最合适的边缘服务器，并确定用户与基站之间的连接状态，从而最大化资源利用率。此外，所提出的算法还考虑了网络的动态变化，如用户请求的波动和服务器负载的变化。本文所提出的方法可以通过实时调整任务分配策略，确保在变化的环境中维持高效的资源分配和优质的用户体验。仿真实验结果证明，与基准算法相比，本研究提出的 DRL 算法在任务调度方面表现更优，能够更好的服务用户。

2) 在深入研究用户任务调度问题的基础上，本文还研究了在网络流量较低时的能耗优化问题。综合考虑了边缘服务器的计算、传输等多种能耗因素，并提出了一种动态能耗优化策略。该策略在满足网络需求的前提下，通过尝试关

闭边缘服务器来实现能耗最小化。仿真实验结果表明，本文所提出的方法在满足网络需求的前提下，有效降低了能耗。

综上所述，本文针对移动边缘计算中的计算任务调度和网络能耗优化问题，采用 DRL 算法进行求解，实现了网络资源的高效利用和网络能耗的优化。

5.2 展望

尽管本文在 MEC 系统中的任务调度和网络能耗优化方面取得了一定成果，但仍可以在以下两方面开展进一步的研究工作：

1) 实际的移动边缘计算环境中存在众多用户和任务，且基站服务器部署的服务类型多样，但在本文的仿真实验中，假设基站服务器只部署了两种典型的服务类型。未来的研究可以着重于设计多任务类型和多用户场景下的 DRL 算法，以实现更高效的资源分配。

2) 本文主要采用仿真实验验证所提出的深度强化学习算法的有效性。未来，可以搭建移动边缘计算测试平台，在实际网络环境下进一步验证和测试本文所提出算法的性能。

参考文献

- [1] 于滨, 李学华, 潘春雨, 李娜. 基于深度强化学习的边云协同资源分配算法[J]. 计算机科学, 2022, 49(07):248-253.
- [2] Fang J., Zhang M. Y., Wei J. H., etc. Smart collaborative optimizations strategy for mobile edge computing based on deep reinforcement learning[J]. Computers and Electrical Engineering, 2021, 96.
- [3] Panda S. P., Banerjee A., Bhattacharya A. User allocation in mobile edge computing: A deep reinforcement learning approach[C]//2021 IEEE International Conference on Web Services (ICWS). IEEE, 2021: 447-458.
- [4] 唐伦, 胡彦娟, 刘通, 等. 移动边缘计算中基于 Lyapunov 的任务卸载与资源分配算法[J]. 计算机工程, 2021, 47(03):29-36.
- [5] We Z. Q., Ma H., Zhang Q. X., etc. Challenge and Trend of Sensing, Communication and Computing Integrated Intelligent Internet of Vehicles[J]. ZTE Technology Journal, 2020, 26(1): 45-49.
- [6] 王秋云. 移动边缘计算网络能耗优化算法研究[D]. 哈尔滨工业大学, 2020.
- [7] Wu J., Bao Y. A., Zhou S., etc. Base-station sleeping control and power matching for energy–delay tradeoffs with bursty traffic[J]. IEEE Transactions on Vehicular Technology, 2015, 65(5): 3657-3675.
- [8] Shuai H., Li F., Pulgar-Painemal H., etc. Branching dueling Q-network-based online scheduling of a microgrid with distributed energy storage systems[J]. IEEE Transactions on Smart Grid, 2021, 12(6): 5479-5482.
- [9] 杨波. 旋翼无人机编队智能检测与对抗技术研究[D]. 杭州电子科技大学, 2024.
- [10] 杨天, 杨军. MEC 中卸载决策与资源分配的深度强化学习方法[J]. 计算机工程, 2021, 47(08):37-44.
- [11] 李强, 杜婷婷, 童钊, 等. 移动边缘计算中基于深度强化学习的依赖任务卸载研究[J]. 小型微型计算机系统:1-8.

- [12] Dinh T. Q., Tang J. H., La Q. D., etc. Offloading in mobile edge computing: Task allocation and computational frequency scaling[J]. IEEE Transactions on Communications, 2017, 65(8): 3571-3584.
- [13] Liu J. W., Wei X. L., Fan J. H. Tolerable Data Transmission of Mobile Edge Computing under Internet of Things[J]. IEEE Access, 2019, 7: 71859-71871.
- [14] Xiao S., Liu C., Li K., etc. System delay optimization for Mobile Edge Computing[J]. Future Generation Computer Systems, 2020, 109: 17-28.
- [15] 王云鹏. 基于深度强化学习的移动边缘计算的资源优化方法研究[D].北京交通大学,2021.
- [16] 高素林,张盈希,任奕菲,等. 基于移动边缘计算的任务调度算法设计[J]. 无线电通信技术,2022,48(05):822-829.
- [17] Pham Q. V., Leanh T., Tran N. H., etc. Decentralized Computation Offloading and Resource Allocation for Mobile-Edge Computing: A Matching Game Approach[J]. IEEE Access, 2018, 6: 75868-75885.
- [18] Yang X. T., Yu X. Y., Huang H., etc. Energy Efficiency Based Joint Computation Offloading and Resource Allocation in Multi-Access MEC Systems[J]. IEEE Access, 2019, 7: 117054-117062.
- [19] 肖海林,黄天义,代秋香,等. 基于轨迹预测的安全强化学习自动变道决策方法[J/OL]. 计算机应用, 2023,1-8.
- [20] 尹晓光. 基于深度强化学习的多用户移动边缘计算任务卸载策略优化[D].北京交通大学,2021.
- [21] 耿小芬. 移动边缘计算技术综述[J]. 山西电子技术,2020,(02):94-96.
- [22] Yu N., Wang Z. Y., Jia X. H., etc. Minimal Energy Broadcast for Delay-Bounded Applications in Satellite Networks[J]. TGCN,2018,2(3).
- [23] 李燕君,蒋华同,高美惠. 基于强化学习的边缘计算网络资源在线分配方法[J]. 控制与决策,2022,37(11):2880-2886.
- [24] 詹文翰,王瑾,朱清新,等. 移动边缘计算中基于深度强化学习的计算卸载调度方法[J]. 计算机应用研究,2021,38(01):241-245+263.
- [25] 邝祝芳,陈清林,李林峰,等. 基于深度强化学习的多用户边缘计算任务卸载调

- 度与资源分配算法[J]. 计算机学报,2022,45(04):812-824.
- [26] 高云,郭艳艳,卫霞. 移动边缘计算网络中基于 DQN 的能效性卸载决策及无线资源分配[J]. 测试技术学报,2022,36(01):54-59.
- [27] 卢海峰,顾春华,罗飞,等. 基于深度强化学习的移动边缘计算任务卸载研究[J]. 计算机研究与发展,2020,57(07):1539-1554.
- [28] 杨戈,张衡. 面向移动边缘计算基于强化学习的计算卸载算法[J]. 电子技术应用,2021,47(02):36-40+48.
- [29] 丁晨阳. 移动边缘计算网络中能耗最小化的任务卸载方法研究[D].黑龙江八一农垦大学,2024.
- [30] 李旭冉. 基于移动边缘计算的蜂窝用户任务卸载计算能耗优化技术研究[D].南京邮电大学,2023.
- [31] 孙景余. 基于边缘计算的卫星网络资源调度研究[D].北京邮电大学,2024.
- [32] Mach P., Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. IEEE Communications Surveys & Tutorials, 2017, PP(3):1-1.
- [33] 周继鹏,李祥. 基于深度强化学习的边缘网络内容协作缓存与传输方案研究[J/OL].计算机应用研究:1-10[2024-03-12].
- [34] 谢人超,廉晓飞,贾庆民,等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 039(011):138-155.
- [35] Ouyang T., Li R., Chen X., etc. Adaptive User-managed Service Placement for Mobile Edge Computing: An Online Learning Approach[C]// IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. IEEE, 2019.
- [36] 胡世红. 边缘计算中资源动态调度的 QoS 优化技术研究[D].江南大学,2021.3
- [37] Vaishnavi A. S., Naresh E., Vijay G. R. Performance evaluation of mobile edge computing using 5G networks[C]//2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, 2021: 1-6.
- [38] Zhang J., Jiang B., Zhao H., etc. Time-Sensitive Multi-User Oriented Mobile Edge Computing Task Scheduling Algorithm[C]//2020 2nd International Conference on Computer Communication and the Internet (ICCCI). IEEE, 2020: 145-149.

- [39] Ma X., Zhou A., Zhang S., etc. Dynamic task scheduling in cloud-assisted mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2021, 22(4): 2116-2130.
- [40] Liu T., Zhang Y., Zhu Y., etc. Online computation offloading and resource scheduling in mobile-edge computing[J]. IEEE Internet of Things Journal, 2021, 8(8): 6649-6664.
- [41] Fang J., Zhang M., Ye Z., etc. Smart collaborative optimizations strategy for mobile edge computing based on deep reinforcement learning[J]. Computers & Electrical Engineering, 2021, 96: 107539.
- [42] Xie Q., Wang Q., Yu N., etc. Dynamic service caching in mobile edge networks[C]//2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2018: 73-79.
- [43] Panda S. P., Banerjee A., Bhattacharya A. User allocation in mobile edge computing: A deep reinforcement learning approach[C]//2021 IEEE International Conference on Web Services (ICWS). IEEE, 2021: 447-458.
- [44] Sutton R.S. and Barto A.G. Reinforcement Learning: An Introduction. MIT press, 2018.
- [45] Lai P., He Q., Abdelrazek M., etc. Optimal edge user allocation in edge computing with variable sized vector bin packing[C]//Service-Oriented Computing: 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings 16. Springer International Publishing, 2018: 230-245.
- [46] Redmon J., Divvala S.K., Girshick R.B., etc. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [47] Sandler M., Howard A.G., Zhmoginov A., etc. Mobilenetv2: Inverted residuals and linear bottlenecks, in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2018:4510-4520.
- [48] Raffin A., Hill A., Ernestus M., etc. Stable baselines3,<https://github.com/DLR-RM/stable-baselines3>, 2019.

- [49] 李智威,莫王昊,李北,等.基于资源约束多目标优化的物联网边缘计算资源分配[J].现代电子技术,2024,47(02):59-66.
- [50] 李长祥. 多接入边缘计算网络中卸载决策和资源分配的智能优化研究[D].南京邮电大学,2023.
- [51] 姚惠娟,陆璐,段晓东. 算力感知网络架构与关键技术 [J]. 中兴通讯技术, 2021, 27 (03): 7-11.
- [52] 刘大畅,丁浩,曾晶. 5G 核心网网络架构及关键技术分析 [J]. 中国新通信, 2021, 23 (01): 81-82.
- [53] 柴卓, 杨卉. 5G 网络的移动边缘计算架构及应用研究 [J]. 信息与电脑(理论版), 2020, 32 (09): 170-172.
- [54] 刘春林,秦进. 面向 5G 网络的移动边缘计算节点部署算法设计 [J]. 计算机仿真, 2022, 39 (12): 436-439+473.
- [55] 甘钊宇. 基于强化学习的 5G 移动边缘计算任务调度系统研究与实现[D]. 北京邮电大学, 2022.
- [56] 黑显冬. 基于云计算技术的 5G 网络优化 [J]. 数字通信世界, 2024, (01): 80-82.
- [57] 黎志忠. 云计算技术在计算机网络安全存储中的实施策略 [J]. 数字技术与应用, 2023, 41 (11): 240-242.
- [58] Chen J., Chen S., Chen W., etc. Design of Typical Application Scenarios for Integrated Demand Response in Multi-energy Collaborative System[C]//2021 11th International Conference on Power and Energy Systems (ICPES). IEEE, 2021: 813-817.
- [59] Oikonomou E., Rouskas A. Selection of service nodes in edge computing environments[C]//2020 7th international conference on internet of things: systems, management and security (IOTSMS). IEEE, 2020: 1-6.
- [60] Ghosh S., Agrawal D. P. A high performance hierarchical caching framework for mobile edge computing environments[C]//2021 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2021: 1-6.
- [61] Sun Y., Chen W., Yuan Y. Research of Caching Methods in Mobile Edge Networks[C]//2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2021: 521-524.

- [62] Zhu X., Yuan S., Zhao P. Research and application on key technologies of 5G and C-V2X intelligent converged network based on MEC[C]//2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). IEEE, 2021: 175-179.
- [63] Khamari S., Rachedi A., Ahmed T., etc. Adaptive Deep Reinforcement Learning Approach for Service Migration in MEC-Enabled Vehicular Networks[C]//2023 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2023: 1075-1079.
- [64] Guo Z., Zhang P., Xia J. Design of virtual reality education platform based on 5G MEC[C]//2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS). IEEE, 2021: 572-578.
- [65] Siriwardhana Y., Porambage P., Liyanage M., etc. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects[J]. IEEE Communications Surveys & Tutorials, 2021, 23(2): 1160-1192.

攻读学位期间取得的学术成果目录

- (1) F. Han, N. Yu, J. Gong, Y. Ge and X. Gao, "Task Scheduling for Mobile Edge Computing Leveraging Deep Reinforcement Learning," 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 2023, pp. 1921-1926;
- (2) J. Gong, N. Yu, F. Han, B. Tang and Y. Ge, "Scheduling Strategy for Energy Storage System in Microgrids Employing Deep Reinforcement Learning," 2023 China Automation Congress (CAC), Chongqing, China, 2023, pp. 3778-3783;
- (3) 申请论文相关发明专利一项：基于深度强化学习的移动边缘计算任务调度方法（申请号：2023111233926）。

致 谢

在校三年的时光感觉过得很快转眼就过去了，在这三年的时光里，遇见了很多挫折，但更多是收获和成长。记得刚入学时，自己从来没有接触过学术科研工作，对科研既憧憬又陌生。三年来，感谢陪伴我一起度过快乐又充实的读研时光的每位尊敬的老师和亲爱的同学们。热心的帮助和认真的指导，才让我勇敢面对挫折，解决问题，直到学业顺利完成。

首先要感谢我的导师余诺老师，余老师有着深厚的学术底蕴、严谨的学术作风、认真负责的态度，在科研上给了我很多建议和指导，每次指导总是一针见血，学术的熏陶，每次指导都能学习到很多知识，这也让我有了方向和动力。每次例会也会认真询问最近的学习情况，认真细心的帮助学生解决困惑和难题，及时给予学生正确的指导。

还要感谢 404 实验室的小伙伴们，请教问题总是会耐心认真的给予建议，一起解决问题。包括一起入学的好同学、好朋友苗宇彤，总是会在我遇到困难的时候开导我，给我支持和鼓励，我们一起成长和收获。另外还要感谢父母家人的关心和陪伴，才能顺利完成学业。

最后，向各位评审老师旨意最真挚的感谢！

尚德敏学 唯实惟新

