Evaluation Warning: The document was created with Spire.Doc for Python.

****

**2024届**

**本科毕业论文（设计）**

**论文题目： 基于深度强化学习的异构网络算力分配研究**

**学生姓名：**

**所在学院： 计算机与人工智能学院**

**专 业： 计算机科学与技术**

**学 号： 41805054**

**指导教师：**

**成 绩：**

**2024年 03月**

**西南财经大学**

**本科毕业论文原创性及知识产权声明**

本人郑重声明：所呈交的毕业论文是本人在导师的指导下取得的成果，论文写作严格遵循学术规范。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。因本毕业论文引起的法律结果完全由本人承担。

本毕业论文成果归西南财经大学所有。

特此声明

毕业论文作者签名：

作者专业：

作者学号：

年 月 日

# 摘要

随着互联网的技术的迅猛发展，以及大数据、人工智能等技术的飞速创新，应对海量数据传输和处理需求的异构算力网络应时而生。异构算力网络根据业务需要，在云端、边缘、以及终端设备等不同的异构计算资源之间进行任务的分配和资源的优化调度，旨在满足各计算节点的算力服务需求，同时极力提升资源的使用效率。本文在基于无向图概念提出的异构算力网络模型基础上引入深度强化学习算法，通过深度强化学习算法的各项优势提升异构算力网络系统的性能，以适应现代社会对异构算力网络的需求。为了在实际应用中解决这一问题，本文设计了一种基于深度Q网络的异构资源协同优化算法，并且提供了相关理论性能分析。最后，通过实验证明了所提算法理论的实用性，还通过与其他算法的比较分析，展示了其在服务效率方面的性能优越性。

**关键词：异构资源协同优化；深度强化学习算法；DQN算法**

**Abstract**

With the rapid development of Internet technology, along with the swift innovation in big data and artificial intelligence, the emergence of heterogeneous computing power networks has been timely to address the demands of massive data transmission and processing. These networks, based on business needs, allocate tasks and optimize resource scheduling among different heterogeneous computing resources such as cloud, edge, and terminal devices. The goal is to meet the computing service demands of various computing nodes while significantly enhancing the efficiency of resource utilization. This paper introduces a heterogeneous computing power network model based on the concept of undirected graphs and incorporates deep reinforcement learning algorithms to improve the performance of the heterogeneous computing power network system, catering to the modern societal needs for such networks. To address this issue in practical applications, the paper designs a heterogeneous resource collaborative optimization algorithm based on deep Q-networks and provides related theoretical performance analysis. Finally, through experimental validation, the practicality of the proposed theoretical algorithm is demonstrated, and its superior performance in service efficiency is showcased through comparative analysis with other algorithms.

**Keywords：Heterogeneous Resource Coordination Optimization; Deep Reinforce-ment Learning Algorithm;DQN Algorithm**

目录

[**第一章 引言 1**](#_Toc160961926)

[**1.1 研究背景与意义 1**](#_Toc160961927)

[**1.2 国内外研究现状 2**](#_Toc160961928)

[**1.3论文组织架构 3**](#_Toc160961929)

[**第二章 相关理论基础 5**](#_Toc160961930)

[**2.1 强化学习概述 5**](#_Toc160961931)

[**2.2 强化学习的元素 5**](#_Toc160961932)

[**2.3深度强化学习算法 6**](#_Toc160961933)

[**2.3.1马尔可夫决策过程 7**](#_Toc160961934)

[**2.3.2深度 Q 网络（DQN） 7**](#_Toc160961935)

[**第三章 系统模型及问题表述 8**](#_Toc160961936)

[**3.1通用应用模型 8**](#_Toc160961937)

[**3.2异构算力网络模型 9**](#_Toc160961938)

[**3.3问题表述 10**](#_Toc160961939)

[**第四章 算法设计 11**](#_Toc160961940)

[**4.1强化学习要素定义 11**](#_Toc160961941)

[**4.2 DQN框架构建 12**](#_Toc160961942)

[**第五章 实验设置与性能评估 15**](#_Toc160961943)

[**5.1实验设置 15**](#_Toc160961944)

[**5.2性能评估 17**](#_Toc160961945)

[**5.2.1随机分配方法 17**](#_Toc160961946)

[**5.2.2 Baseline方法 18**](#_Toc160961947)

[**5.2.3对比分析 19**](#_Toc160961948)

[**第六章 总结与展望 20**](#_Toc160961949)

[**参考文献 22**](#_Toc160961950)

[**致谢 23**](#_Toc160961951)

# 第一章 引言

## 研究背景与意义

在数字经济时代背景下，随着物联网的兴起，如大数据和人工智能等技术的革新加速了数字经济的增长，并带来了巨量数据处理需求和更高的任务执行标准。面对新时代对数据传输和计算能力的需求，异构算力网络应时而生。异构算力网络在云计算、边缘计算、终端等异构计算资源之间进行有效的任务划分和资源配置，旨在满足各节点的计算服务需求，同时最优化资源的使用效率。

深度强化学习（Deep Reinforcement Learning，DRL）融合了深度学习和强化学习的优势，因而在各个领域中备受瞩目。其突出的优点是适应复杂环境的能力，即能够在高度不确定、复杂且高维度的环境中进行学习。通过深度神经网络，DRL 可以进行端到端学习，无需手动提取特征，实现了更加自动化的学习过程。同时，DRL强大的泛化能力使得模型在未见过的情境中也能表现出色，无需重新设计。此外，DRL 可以处理非线性关系，从而能够在建模和优化策略的同时自动学习任务相关的特征，具备更广泛的适用性。最后，对于需要处理大规模数据和高维动作空间的问题，DRL 的计算能力和深度神经网络的特性使其更为有效；而在需要快速决策和迭代优化的任务中，DRL也有优秀的表现，因为DRL支持在线学习，能够即时更新策略以适应环境的变化。

总体而言，深度强化学习以其适应多样环境、端到端学习、泛化能力强等优势，成为解决复杂任务的有力工具，在游戏、机器人控制、自然语言处理等领域有着广泛应用。因此，深入研究异构网络算力分配，尤其是结合深度强化学习的方法，对于推动异构算力网络的发展，提高其在数字经济时代的应用能力具有重要意义。

## 国内外研究现状

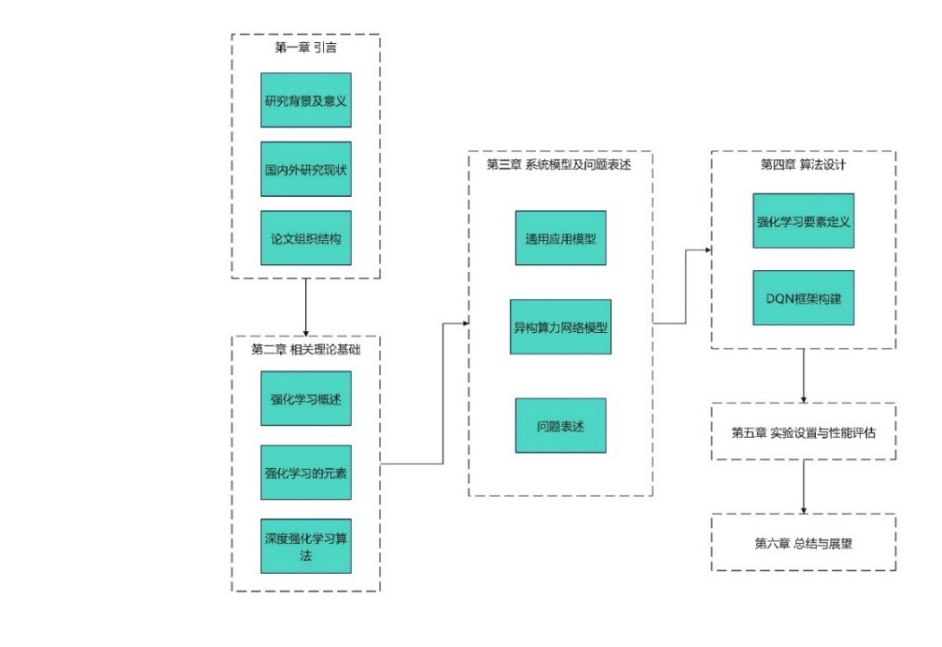
在当今以“计算”为核心、“网络”为基础的时代，面向计算需求的异构算力网络正在迅猛发展。异构算力网络的目标是提供更加高效的网络化计算、存储及传输服务。这就要求在现有的网络功能与技术框架的基础上，异构算力网络需要对云计算、边缘计算以及广泛分布的智能设备中的异构计算资源进行更深层次的整合。然而，由于异构网络中的节点在设备配置、网络状况、地理位置等多方面存在显著差异，这些差异使系统难以克服性能瓶颈节点，导致网络资源的整体利用效率不高，服务质量难以提升。

为此，国内外学者提出了众多关于异构网络资源优化的方案。例如，文献[11]中介绍了一种名为边缘矩阵（EdgeMatrix）的云边协作异构资源优化框架。该框架通过采用网络化的多智能体增强学习技术，对系统内各项任务的资源配置进行智能适配。尽管这种云边协作架构能够满足一部分用户的服务需求，但在大规模用户应用场景下，资源的不足和成本的增加仍是无法解决的问题。雷波等人[1]并构建了一个算力网络管理平台。该平台利用分布式路由协议和算力网络管理平台来分发资源信息，并对网络资源进行集中式调度。刘泽宁及其团队[2]设计了算力网络和计算卸载系统，该系统利用博弈论方法来优化多用户任务的调度。李铭轩等研究者[3]采用了基于“Kubernetes+K3S”的双层结构，以实现边缘侧资源的统一调度管理，促进异构嵌入式计算资源的协同工作。这种异构算力网络通过对资源状态的实时监测和可编程的流量调度，能够将任务有效分配至适宜的节点，以提高系统效能。易昕昕等人[4]主要探讨算力网络可编程服务中路由策略的设计。贾庆民等人[5]则专注于确定性算力网络的技术架构和运作机制的设计。蔡岳平等研究者[6]对确定性网络的标准化进行了研究。陈星延等人[10]设计了基于波利亚重球法（ Polyak heavy-ball method） 的异构资源协同优化算法，提出了一种面向“云—边—端”算力网络的计算和传输联合优化方案。

综上所述，对异构算力网络中资源的协同优化而言，必须全面考虑包括通用化需求、大规模部署需求以及协同并发虚拟化需求在内的多种因素。其目标不仅是实现应用程序的广泛适用性，还需兼顾大规模应用场景的特定需求；不仅需确保数据处理与传输的并行性和可靠性，还应实现硬件与软件的解耦，构建它们之间的逻辑联系，以便于资源能够被灵活地调配。然而，目前的解决方案尚不能全面应对这些挑战，存在一定的局限性。

与此同时，许多研究者也在探索将深度强化学习技术应用于异构算力网络中，以期实现更高效的任务卸载和资源调度。例如邝祝芳等人[7]设计了一种双层的任务卸载和资源调度策略。在这个策略的上层，运用基于贪婪算法的流水线作业调度方法来处理任务卸载问题；而在下层，则通过深度强化学习对服务器资源分配进行了优化，从而实现更高效的资源利用。刘晓宇等人[8]则提出了一个基于深度强化学习的动态优先级并发接入算法，这个算法首先为异构设备分配不同的服务优先级，然后应用强化学习技术进行效用优化，以提高资源调度的效率和效果。喻鹏等人[9]在移动边缘网络中引入深度强化学习算法，通过搭建DDQL框架探究高能效的资源分配方法，解决传统的资源分配方法可能会导致资源利用不足或者性能下降的问题。

## 1.3论文组织架构



**图1-1：论文组织框架图**

本文共包含六个章节，其组织结构如图1-1所示，各章节内容安排如下：

第一章：本章首先探讨了在异构算力网络中应用深度强化学习进行算力分配的研究背景及其重要性。随后，本文综合分析了国内外在该领域的研究进展和现状，指出了现有研究工作在理论深度、应用范围及优化效率方面的不足之处。鉴于这些不足，本文明确了自身的主要研究方向和内容，旨在通过深度强化学习技术，提出更为高效、灵活的异构网络算力分配策略，以优化资源利用率，提高服务质量。

第二章：本章深入介绍强化学习的相关基础理论，涵盖了强化学习的基本概念、关键元素及深度强化学习算法的核心机制，为后续在异构算力网络中构建基于深度Q网络（DQN）的框架提供坚实的理论基础，同时也为第三章和第四章的研究提供理论依据。

第三章：本章主要提出两个模型以及对本文研究问题的表述。首先、针对通用应用服务的特点，对传统的网络服务链表示模型进行了改进，引入通用服务模型。其次，基于无向图概念提出异构算力网络模型。最后，利用这两个模型对异构网络算力分配问题进行形式化表征，为第四章构建DQN框架提供指导。

第四章：基于第三章研究结果，进行强化学习关键元素定义，构建DQN框架，并通过框架图、伪代码等形式对所构建框架进行详细解释，达到异构网络算力分配问题中引入深度强化学习方法的目标。

第五章：对第四章提出的DQN框架进行实验验证，从多个方面对其实验结果进行阐述，并与其他方法（随机分配方法、Baseline方法）进行对比，以验证在异构算力网络中运用DQN算法具有服务效用和资源成本方面的性能优势。

第六章：本章总结本文的核心研究内容，概述本研究所涵盖的关键领域、实施过程以及通过此研究所实现的成果。同时，本章还指出本文的不足之处以及未来可能的改进方向。

# 第二章 相关理论基础

## 2.1 强化学习概述

强化学习是一种机器学习方法，它使智能体能够在与环境互动的过程中学习到最佳行为策略。在这一学习范式中，智能体通过对环境状态的观察，采取行动，并根据环境反馈的奖励或惩罚来调节其行为方式。此过程的核心目标是让智能体通过反复的尝试和错误过程，学习到一个能够最大化其长期累计奖励的行为策略。

## 2.2 强化学习的元素

在强化学习的理论框架中，核心元素可分为状态、动作、奖励以及策略。

状态被定义为对环境的详细描述，作为智能体在决策过程中的观测点，涵盖智能体为做出决策所需的全部信息。状态可以是任何形式的观测数据，例如传感器读数、图像、文本或其他形式的输入。在强化学习框架下，智能体的主要任务是根据其所处的当前状态，选择一个最佳动作，目的是最大化其从当前时刻开始的长期奖励总和。

动作是智能体在特定状态下的反应或响应。在每一状态，智能体都面临着多个可能的动作选择，而它将根据其策略来决定采取哪一种动作。动作可以是离散的（例如，棋子的不同移动方向），也可以是连续的（如机器人的关节角度调整）。通过执行特定的动作，智能体能够对环境状态产生影响，进而改变未来的状态和可能获得的奖励，这是智能体通过学习最优策略以实现长期奖励最大化的关键过程。

奖励是智能体在执行了某个动作之后，从环境接收到的数值反馈。这个奖励信号为智能体在特定状态下所采取的动作提供了即时的反馈，用来引导智能体调整其决策策略，目的是在未来的行为中最大化累计的长期奖励。奖励可以是正数，以奖励智能体采取了某些有益的行动；也可以是负数，用以惩罚智能体的不当或有害行为。

策略在强化学习中是一个核心概念，它表示智能体在面对特定状态时，对于采取各种动作的概率分布。简而言之，策略就是智能体基于当前状态决定其下一步行动的规则或方法。强化学习的目标是发现或学习一个最优策略，即一个能够指导智能体在遇到任何状态时都能做出最佳动作选择的策略，以此来最大化长期奖励的总和。策略可以采取确定性形式，即对于给定的状态总是选择一个特定的动作；也可以采取随机性形式，即对于给定的状态，根据一定的概率分布来选择动作。

强化学习的核心目标是发现或学习一个最优策略，这样智能体就能在遭遇任何状态时做出最有益的动作决策。这个目标的实现依赖于强化学习理论体系中的几个基本元素：状态、动作、奖励、以及策略。这些元素通过它们之间的相互作用，使得智能体能够逐步调整并优化其决策策略，以适应环境的变化，并最大化长期累积的奖励。在这一过程中，智能体持续与环境交互，观察环境状态的变化，根据自身策略执行动作，并通过环境反馈的奖励来评估其行动的效果。通过这样的试错过程，智能体学习如何调整其行为，以提高获得正奖励的概率，减少遭受惩罚的风险。这种学习机制允许智能体在未知或动态变化的环境中做出适应性强、效率高的决策，并随着时间的推移不断提升其性能。

因此，强化学习不仅是关于优化当前决策的理论，它还涉及到如何通过不断的学习和适应，实现智能体在复杂和不确定环境中的长期成功。这种学习和适应的能力，使得强化学习在许多领域，如自动驾驶、游戏、机器人技术等，都显示出了巨大的应用潜力和价值。

## 2.3深度强化学习算法

深度强化学习是一种先进的机器学习方法，它结合了深度学习和强化学习的优点，以处理和解决高维度状态和动作空间的复杂问题。通过利用深度神经网络来近似值函数或策略函数，深度强化学习使得智能体能够在那些传统强化学习技术难以处理的复杂环境中学习最优策略，其中代表性的算法如深度 Q 网络（DQN）、双重深度Q网络（DDQN）以及异步优势演员-评论家（A3C）。

### 2.3.1马尔可夫决策过程

马尔可夫决策过程可被定义为一个五元组 ，其中代表虚拟节点所对应的智能体集合。代表有限的状态空间。代表动作集合，表示为。对于智能体，可用动作集合为。表示当执行动作 时从状态到的转移概率函数。是即时奖励函数。

对于所有的智能体 ，策略集合是一个映射，其目标是得到最大化的累积奖励期望。由此可得长期平均奖励回报的方程如下：

其中是智能体在时刻的奖励。

### 2.3.2深度 Q 网络（DQN）

深度Q网络（DQN）是由DeepMind开发的一种革命性深度强化学习算法，它在处理具有离散动作空间的任务中尤为有效。在传统的强化学习中，智能体通过不断地与环境互动，学习在特定状态下采取何种动作能够最大化其获得的累计奖励。DQN算法通过整合深度神经网络来近似Q值函数，这是一个核心的创新，使得算法能够有效地应对那些状态空间庞大且复杂的问题。与传统的Q-learning相比，DQN的主要优势在于其使用深度神经网络作为一个函数逼近器，来估计状态-动作对的Q值。这种方法使得DQN能够学习到一个从高维感知输入（如原始像素）到动作值评估的直接映射，从而处理以前难以解决的复杂环境。

DQN 的核心思想是运用深度神经网络对Q值函数进行近似，将环境的当前状态输入网络，以获得每个可能动作的Q值输出。通过采用反向传播和梯度下降法，DQN持续优化网络参数，目的是减少Q值函数预测的误差，使智能体得以精确学习各动作的价值估计。同时，为了提升训练的效率与稳定性，DQN引入了经验回放和固定Q目标网络的策略。

经验回放的策略使得DQN能够在训练期间储存既往的经验，并通过随机选取这些经验进行学习，从而破除了数据序列的相关性，增强了样本使用的效率。与此同时，通过定时更新目标网络参数的方法，固定Q目标网络有助于训练过程的稳定化。具体而言，DQN采用了两个结构相同的深度神经网络：一是用于动作选择的行为网络，二是用于估算目标Q值的目标网络。行为网络负责实时做出最佳动作决策，目标网络的参数则定期从行为网络更新，旨在维护目标Q值的一致性。

在训练过程中，DQN 使用ɛ-greedy 策略来平衡探索和利用。具体来说，以ɛ 的概率随机选择动作，以 1-ɛ 的概率选择当前估计 Q 值最大的动作。通过这样的策略，DQN 能够在训练过程中逐步减小探索的概率，从而更加专注地利用已有的知识来提高性能。

总体而言，作为深度强化学习算法的一种，DQN在处理复杂的强化学习挑战上展现出显著成效。它利用了深度神经网络强大的表征能力与强化学习的高效优化机制的结合，使得智能体能从环境互动中学习复杂策略，从而具有卓越的性能。

# 第三章 系统模型及问题表述

## 3.1通用应用模型

网络功能通常以链式模型形式存在，即一系列按照特定顺序排列的网络功能实例。在此模型中，数据流需要按顺序通过各个网络服务功能进行处理后，才能到达最终目标。但是，随着技术进步和应用需求的不断扩大，这种传统的链式模型在某些情况下显得缺乏灵活性和效率。尤其是在工业互联网、元宇宙、自动驾驶等新兴应用领域，网络服务功能面对的需求更加复杂多样，对数据处理速度和处理能力的要求也相应提高。

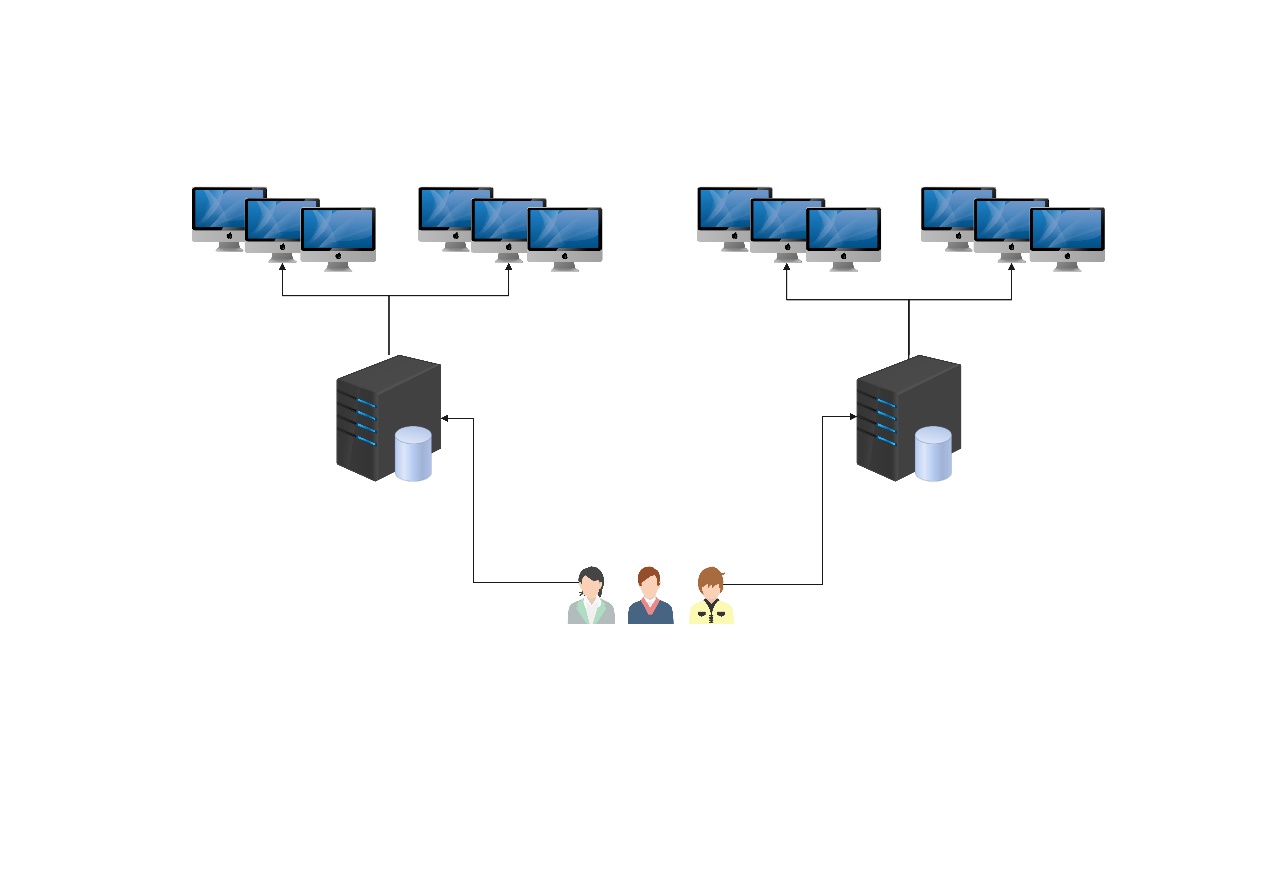
针对这一背景，优化和改进链式模型变得非常关键。通过采用更加灵活的网络服务架构，能够更有效地应对复杂和多变的应用需求。比如，在元宇宙场景中，虚拟数字人的生成就是一个具体的应用实例，该任务可以细分为语音生成与视频生成两个子任务。假如网络节点能够提供充足的计算资源，则这两个子任务便可分别卸载至不同的网络节点上，实现并行处理，而非遵循传统链式模型的顺序执行。这种任务的分解与并行执行极大地提升了数据处理的效率与速度。

更进一步，单个子任务还可以细分为更小的任务单元，这些更细小的任务单元能够被分散至网络中的多个节点上，实现协同工作。这样的多节点协作处理模式，不仅显著提高了处理效率，也增强了系统的容错性和灵活性。例如，在遇到某个节点由于出现故障而不能完成其分配任务的情况下，该任务可以迅速被重新分配到其他节点上，以此确保整个系统的稳定与可靠运行。

此外，这种经过改良的模型还能够更加有效地适应异构算力网络中的动态变化，例如网络负载的波动和计算资源分布的不均衡。通过采用智能化的任务调度与资源分配策略，能够确保网络中的每份计算资源被最大化利用，从而在更大程度上提升算力网络的整体表现和效率。

## 3.2异构算力网络模型

如图3-1所示，本文将异构算力网络系统建模为一个无向图 ，其中和分别代表网络中节点和链路的集合，具体表示为和。定义系统中的网络节点为三种不同的角色：①用户（User），即向特定服务提供者发起请求的节点，是任务请求的发起点；②主机（Host），作为数据传输的中介节点，在异构网络中负责将任务数据流重定向到合适的工作节点进行处理；③工作节点（Worker），具有计算和传输能力，负责处理由主机节点分配过来的任务。假设系统工作在离散的时隙集合 中，并且异构算力网络系统中的每个网络节点都具备计算和链路的转发能力。由于运行中的网络节点可能受到其他应用程序的干扰，因此节点在时刻的可用计算资源被定义为一个时变随机变量，其中 表示节点能够提供的最大计算资源。同样地，将节点在时刻的可用传输资源表示为时变随机变量，其中 表示节点能够提供的最大传输资源。

**

**图3-1：异构算力网络框架图**

除此之外，如表3-1所示，对每个节点都维护两个列表，分别为计算任务列表和传输任务列表，表中元素为任务，表示分配给该节点的计算、传输任务。同时，对每个节点设置计算负载和传输负载两个属性，分别表示节点此时处理所分配计算任务和传输任务消耗的资源。因此，节点在时刻的计算负载和传输负载表示为时变随机变量，。综上所述，可以将节点在时刻的状态表示为4维向量。

**表3-1：节点担负计算、传输任务列表**

|  |  |
| --- | --- |
| Task Compute List | … |
| Task Transmit List | *…* |

设定任务集合为，其中代表需处理的任务数量。对每个任务维护两个属性，分别为计算资源需求和传输资源需求（简写为，下同），表示该任务分配到节点后需要多少计算资源和传输资源才能处理完毕。综上所述，可以将每个任务表示为2维向量。

## 3.3问题表述

本文的目标是在满足内容消费者服务需求的前提下，探索异构网络系统中最优的计算和传输资源协同方案，以提高系统的效率和性能。首先，定义任务实际计算时间为，任务实际传输时间为，任务周转时间为任务实际计算时间与任务实际传输时间之和：

因此，本文的目标就是使每个任务通过异构算力网络系统的分配，得到尽可能小的周转时间，即：

式为约束条件，其中为任务分配给节点处理的时刻，表示此时系统中至少存在一个节点可以满足该任务的计算、传输资源需求，否则系统无法处理该任务。若不存在这样的节点，可能有以下两种情况：①任务资源需求过大，需要通过通用应用模型将任务拆分成若干小任务，再进行处理；②此时系统所有节点的负载都很大，没有足够资源来处理该任务，需要将该任务排后，延期处理。

# 第四章 算法设计

## 4.1强化学习要素定义

根据第二章的相关理论基础，结合本文的优化模型，现对强化学习的要素进行定义。

状态：，为一个联合状态空间，表示时刻整个异构网络算力系统所有节点的状态，其中表示当前需要处理的任务。因此为4·+2维向量。

动作：，为一个联合动作集合，表示当任务来临时分配给哪一个节点处理。因此 为维向量。

奖励：与式中的优化目标相对应。鉴于DQN算法要最大化累积奖励，而模型的优化目标是最小化任务周转时间，并且根据3.2节所述，可定义任务资源需求为计算资源需求与传输资源需求之和：

因此，在任务所需资源恒定的情况下，可将奖励定义为：

即任务周转时间越小，奖励越大。

## 4.2 DQN框架构建

Q值，也称为状态-动作价值函数，体现了在特定状态，并遵循策略执行时，所能获得的预期总奖励。其定义表达式为：

在Q-learning算法中，必须将各个状态-动作对的Q值记录在表格中，但当状态或动作的可能性极为庞大时，这种方法变得不切实际。深度Q网络（DQN）利用深度神经网络（DNN）来近似最优策略下的Q值作为解决方案，表示为 ，如式所示。

其中，参数代表神经网络的权重，在迭代中通过调整参数来训练神经网络。在这个过程中，被用于估计价值函数的神经网络称为Q网络（Q-network)。

本文所使用的深度神经网络属于多层前馈神经网络（FNN）类型，其特点是神经元被分层排列，每一层的神经元与下一层的神经元之间形成全面的连接。该网络通过反向传播算法进行参数的调整。深度神经网络接受状态作为输入，并针对所有潜在动作输出相应的值。此外，该网络采用ReLU函数作为其激活函数，该函数的定义式为：

在DQN算法中，部署了两个结构一致的深度神经网络。一个是当前Q网络，记为，其参数为，负责评估当前状态-动作对的Q值；另一个是目标Q网络，记为，其参数为，用于生成目标Q值。

DQN算法采用的损失函数是均方误差，定义为：

其中，为目标Q值，为：

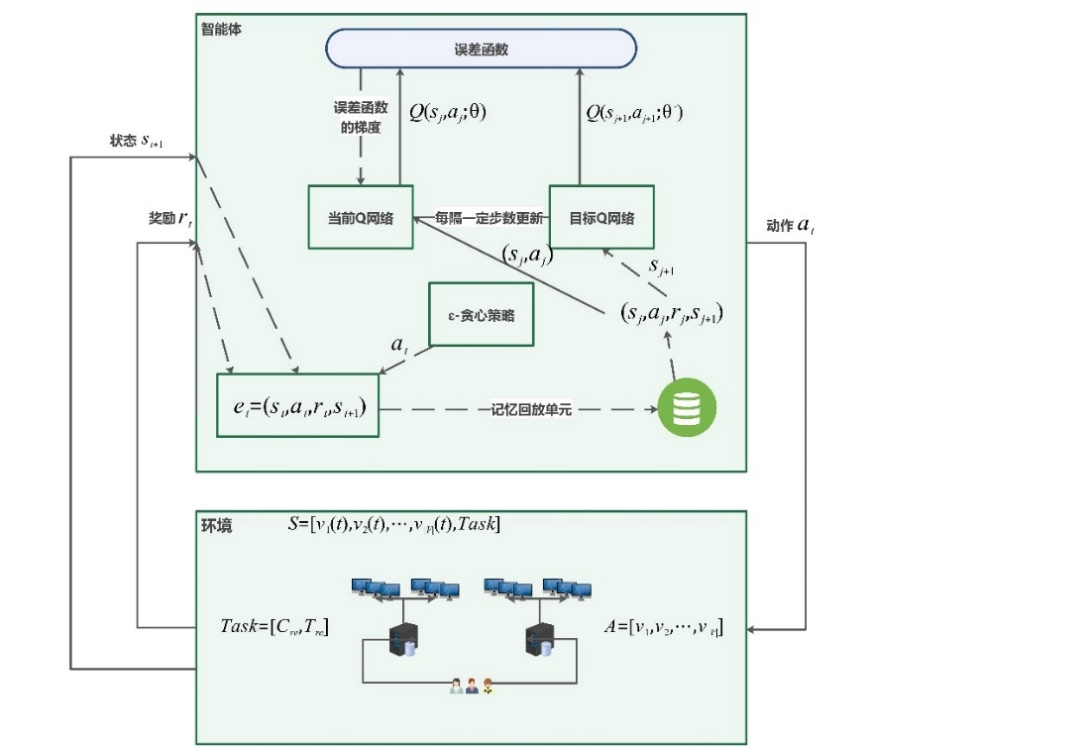
其中，为在状态执行动作后的下一个状态，为状态下可选择的动作。

DQN算法通过采用固定Q目标机制以及两个结构相同的深度神经网络来避免在单一网络中计算误差以及更新参数时引起的训练不稳定性。这种不稳定性主要来自更新网络时持续变化的Q值。因此，DQN算法使用了结构相同的两个网络：一个是当前Q网络，它在每步操作中都通过随机梯度下降法更新以减少误差；另一个是目标Q网络，它的参数在固定步骤后才更新一次，更新时将当前Q网络的参数直接赋值给目标Q网络，从而提高了训练的稳定性。

在DQN算法中为了增强学习效率和稳定性还采用了经验回放机制。具体来说，每当智能体在时刻获得经验时，这个经验就会被保存进回放记忆库中。回放记忆库的容量是有限的，因此当它满载时，加入新经验会随机地移除一些旧经验。在训练阶段，算法会从这个记忆库中随机抽取一批经验样本，然后用这些样本来训练网络并更新参数。这种做法能够打破经验之间的时间相关性，提供更稳定的学习环境。

如2.3.2节所述，在DQN算法训练Q网络的过程中，采取ɛ-贪心策略来选择动作。ɛ-贪心策略定义了一个探索与利用的平衡参数ɛ，其值介于0到1之间。根据这个策略，智能体以ɛ的概率进行随机动作选择，目的是探索未知的状态空间；而以的概率选择当前已知的最大Q值对应的动作，以此利用已有的知识。

DQN算法框架图如图4-1所示。



**图4-1：****DQN算法框架图**

DQN算法训练流程具体伪代码如表4-2所示。

**表4-2：DQN算法训练流程**

|  |
| --- |
| 算法1：DQN算法训练流程 |
| 输入：系统环境参数、任务参数和 DQN算法参数  输出：当前Q网络参数   1. 初始化回放记忆单元 2. 用随机参数初始化当前Q网络，用参数=初始化目标Q网络 3. for episode=1:MaxEpisode 4. 初始化状态s 5. for =1:MaxStep 6. 用ɛ-贪心策略选择动作，并执行 7. 观察下一状态 8. 判断状态是否满足约束条件，计算奖励 9. 将状态转化存入回放记忆单元 10. 从回放记忆单元中随机抽取一批状态转化 11. 对其中状态转化根据式()计算目标Q值，根据 式()计算误差，更新参数 12. 每隔C步更新目标Q网络，即 = 13. end for 14. end for |

# 第五章 实验设置与性能评估

本文对构建的DQN模型进行实验，首先说明实验设置的各项参数，然后对实验结果进行分析，并从各个尺度与随机分配方法、基线方法进行比较，进一步论证DQN模型的性能优势。

## 5.1实验设置

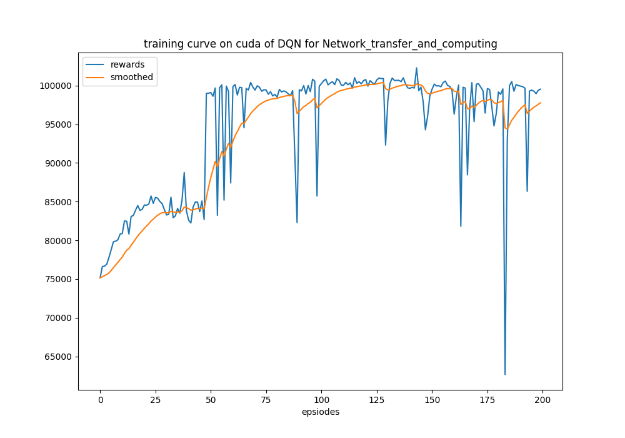
在实验中，设置节点数量为20，节点任务计算和传输处理策略均为FIFO（先来先服务），生成任务的资源需求满足式的约束。DQN算法的各参数设置如表5-1所示。

**表5-1：DQN算法参数设置**

|  |  |
| --- | --- |
| 参数 | 数值 |
| 训练回合Training epochs | 200 |
| 奖励折扣因子Reward discount factor | 0.99 |
| 批次量Batch Size | 32 |
| 隐层维度Hidden layer dimension | 256 |
| 学习率Learning Rate | 0.0003 |
| 衰减率Learning Rate Decay Rate | 400000 |
| 缓冲池Experience Replay Buffer | 100000 |

其中，奖励折扣因子γ是一个介于 0 到 1 之间的值，用于表示未来奖励的衰减程度。γ 决定了我们在计算 Q 值时应该对未来奖励的多大程度进行考虑。具体地说，未来奖励按照 γ 的幂次递减。γ 较接近 0，那么智能体更倾向于考虑即时奖励，而忽略长期奖励；而如果 γ 接近 1，智能体更加重视长期奖励。批次量（Batch Size）即在经验回放中每次从经验池（Experience Replay Memory）中随机选择的样本数量。隐层维度（Hidden layer dimension）指的是神经网络中隐藏层的维度，也就是隐藏层中神经元的数量。学习率（Learning Rate）定义了参数更新时沿着梯度下降方向移动的步长大小。该参数的设定直接影响着参数调整的速度及方向，从而控制了学习进程的快慢和效率。过高的学习率可能导致学习过程中的震荡，而过低的学习率则可能使得学习进度缓慢，难以在合理时间内收敛到最优解。衰减率是指学习率衰减率（Learning Rate Decay Rate）。学习率衰减是指在训练过程中逐渐减小学习率的过程。在训练神经网络时，初始时较大的学习率可能有助于快速收敛，但随着训练的进行，可能会引起参数的摆动或者不稳定。通过衰减学习率，可以逐渐减小参数更新的步长，让优化过程更加平缓。参数缓冲池指的是经验回放缓冲区（Experience Replay Buffer），是一个存储先前经验的固定大小的缓冲区，其作用是保存智能体在环境中执行动作时观察到的状态、采取的动作、获得的奖励以及下一个状态的信息。

DQN模型的奖励曲线如图5-1所示：



**图5-1：DQN算法奖励曲线**

从图5-1可以看出，奖励曲线随回合数增大而上升，至100回合后达到较高水平并且趋于平稳。这有力地说明整个异构算力网络系统的任务处理速度随着模型训练的进行越来越快，系统效率越来越高。

除此之外，本文还在20个节点中选取2个节点，并观察这两个节点在训练前、后期的计算、传输负载，从更微观的角度证明DQN算法性能优势。

|  |  |
| --- | --- |
|  |  |
|  |  |

**图5-2：节点计算、传输负载情况**

如图5-2所示，0号节点的计算能力为15，传输能力9；4号节点的计算能力为20，传输能力9。

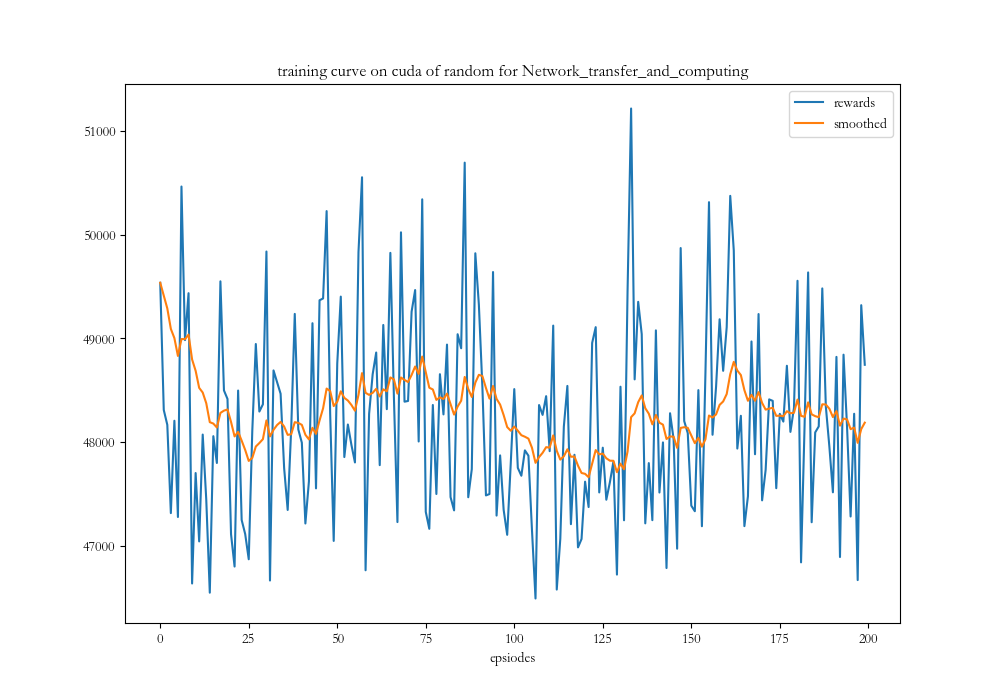
在训练前期，0号节点和4号节点计算负载和传输负载曲线呈现剧烈波动，并且计算负载累积均值小于传输负载累积均值，不符合两节点计算能力、传输能力的实际情况。这说明此时系统分配任务时没有以这两个节点当前的计算、传输能力为依据（即没有给这两个节点分配更为“擅长”的计算任务），不仅造成资源的极大浪费，还导致任务处理时间长，系统效率低。

在训练后期，可以看到两个节点的计算、传输负载趋于均衡，计算负载累积均值大于传输负载累积均值，与两节点计算能力、传输能力的实际情况相符。这说明在DQN算法的作用下，系统根据当前节点的计算、传输能力来分配任务，实现了资源的充分利用，系统效率得到提高。

## 5.2性能评估

### 5.2.1随机分配方法

随机分配方法即对每个任务随机选择节点分配处理。为控制实验变量，设置其他条件均与DQN算法相同。实验结果如图5-3所示。

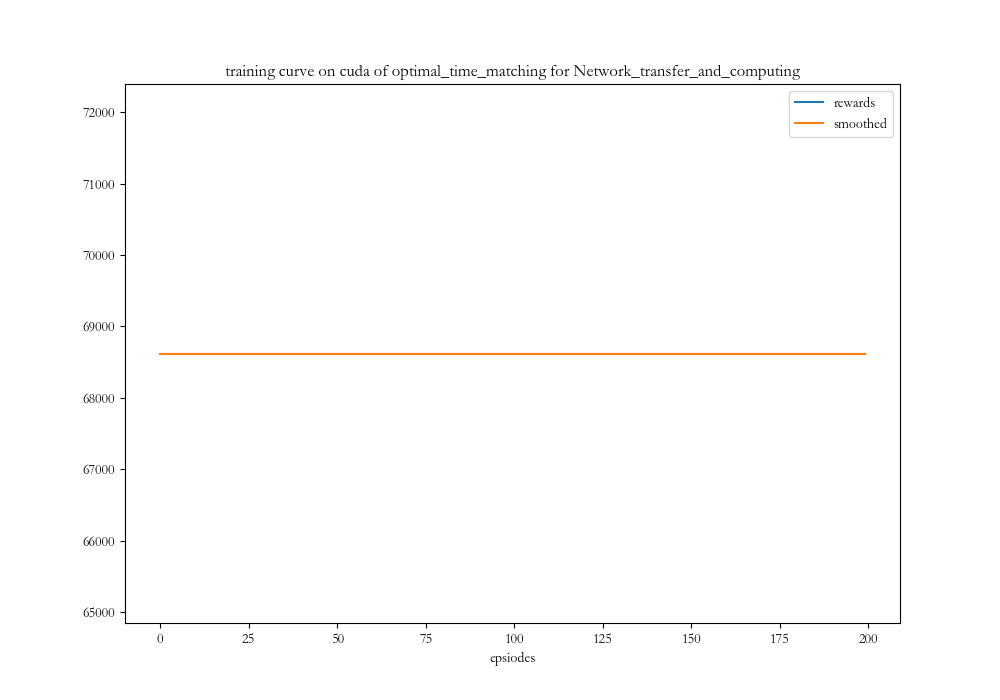


**图5-3：随机分配算法奖励曲线**

从图5-3中可以看出，随机分配算法的奖励曲线随回合数增大而不断上下波动，不会趋于平稳，并且奖励水平较低。这说明该方法不会提升异构算力网络系统的任务处理速度，系统整体效率较低。

### 5.2.2 Baseline方法

本文实验的Baseline方法也称optimal time matching方法，即选择负载最先处理完的节点分配任务。为控制实验变量，设置其他条件均与DQN算法相同。



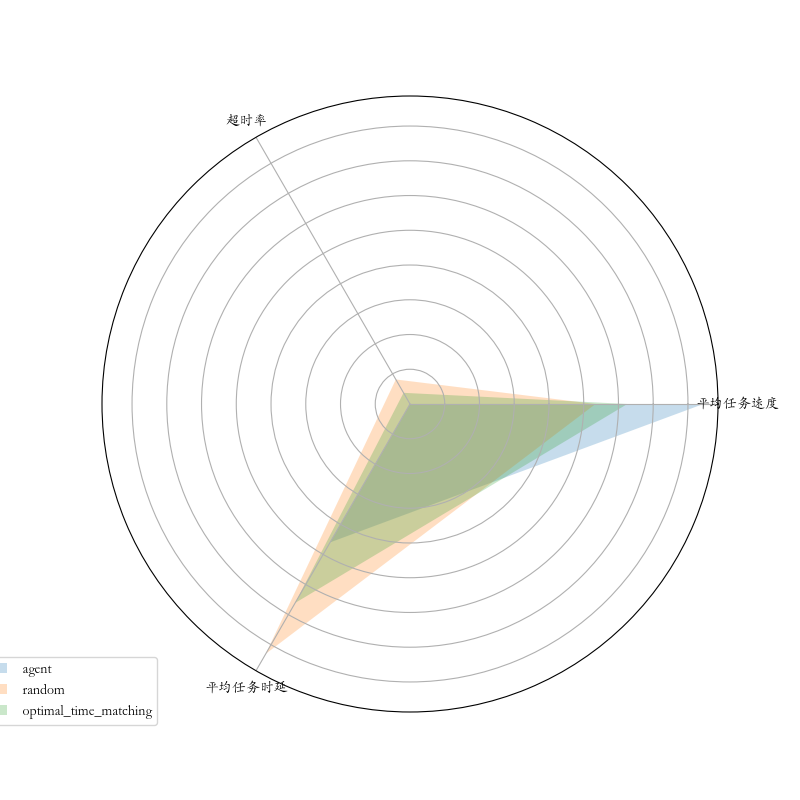
**图5-4：Baseline算法奖励曲线**

从图5-4中可以看出，Baseline算法的奖励曲线是一条直线，所有任务的处理时间都相同（因为任务生成的时间都相同），即该方法不会提升异构算力网络系统的任务处理速度。

### 5.2.3对比分析

对比三种算法的奖励曲线可得知，训练结束时，DQN算法的任务处理速度（即奖励水平）远高于随机分配算法和Baseline算法，Baseline算法的任务处理速度略高于随机分配算法。这说明DQN算法在性能方面确有较大优势，证明在异构算力网络系统中引入深度强化学习算法可以提高系统任务处理速度，改善系统效率。

同时，将总的任务处理时间除以总任务数，即可得到任务平均处理时延，进而可得到任务平均处理速度；将任务处理时间超过4秒的任务标记为超时任务，将超时任务数除以总任务数，即可得到任务超时率。由此可得到DQN、随机分配、Baseline三种算法的任务平均处理时延、任务平均处理速度以及任务超时率雷达图，如图5-5所示。



**图5-5：三种算法性能雷达图**

从图5-5中可以看出，在任务平均处理时延方面，随机分配算法最大、Baseline算法次之，DQN算法最小；在任务平均处理速度方面，随机分配算法最慢、Baseline算法次之，DQN算法最快；在任务超时率方面，随机分配算法最大、Baseline算法次之，DQN算法最小。

综上所述，DQN算法在各项性能指标上均强于随机分配算法和Baseline算法，进一步证实了DQN算法性能方面确实具有较大优势，在异构算力网络系统中引入深度强化学习算法确有益处。

# 第六章 总结与展望

在互联网迅速发展的今天，大数据、人工智能等技术正在飞速创新，这种变革加速了海量数据的产生和传输，也带来了对异构算力网络的更高要求。

本文在已有研究基础上，基于无向图概念提出了一种新的异构算力网络模型，并将深度强化学习算法引入其中，以提升系统性能。通过引入深度强化学习算法，我们利用其在处理复杂环境和大规模数据方面的优势，对异构算力网络系统进行优化，以更好地适应现代社会对算力网络的需求。

本文提出了一种基于深度 Q 网络的异构资源协同优化算法，并对其进行了相关理论性能分析。实验结果表明，本文的算法在服务效用方面表现出明显的性能优势。此外，在与其他算法的对比中也验证了算法的有效性和优越性。

本文的不足之处在于其他对比算法的数量较少，无法从更多的层面验证本文算法的性能优势。同时，由于时间以及自身能力等因素的限制，没有将更多强化学习算法引入到异构算力网络中来，例如PPO算法、双重深度学习算法（DDQN算法）、SAC算法等。这也是未来需要完善的方向。

展望未来，我将继续完善提出的深度强化学习算法，验证其他强化学习算法在异构算力网络中应用的可能性，探索更多的实际应用场景，并进一步优化算法的性能。我将着重关注算法在边缘计算、物联网等领域的应用。我相信，通过不断的研究和实践，我能够为构建更高效、更智能的异构算力网络系统做出更多的贡献。

# 参考文献

1. 雷波，赵倩颖. CPN：一种计算/网络资源联合优化方案探讨[J]. 数据与计算发展前沿，2020，2（4）：55−64
2. 刘泽宁，李凯，吴连涛等. 多层次算力网络中代价感知任务调度算法[J]. 计算机研究与发展，2020，57（9）：1810−1822
3. 李铭轩，曹畅，唐雄燕等. 面向算力网络的边缘资源调度解决方案研究[J]. 数据与计算发展前沿，2020，2（4）：80−91
4. 易昕昕，马贺荣，曹畅等. 算力网络可编程服务路由策略的分析与探讨[J]. 数据与计算发展前沿，2022，4（5）：23−32
5. 贾庆民，胡玉姣，张华宇等. 确定性算力网络研究[J]. 通信学报，2022
6. 蔡岳平，李天驰等. 面向算力匹配调度的泛在确定性网络研究[J]. 信息通信技术，2020
7. 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（4）：812−824
8. 刘晓宇，许驰，曾鹏等. 面向异构工业任务高并发计算卸载的深度强化学习算法[J]. 计算机学报，2021，44（12）：2367−2381
9. 喻鹏，张俊也，李文璟等.移动边缘网络中基于双深度Q学习的高能效资源分配方法[J].通信学报，2020，41（12）：148-161
10. 陈星延，张雪松等. 面向“云—边—端”算力系统的计算和传输联合优化方法[J].计算机研究与发展，2023，60（4）：719−734
11. Ren Yuanming, Shen Shihao, Ju Yanli, et al. EdgeMatrix: A resources redefined edge-cloud system for prioritized services[C] Piscataway, NJ: IEEE, 2022: 610−619

# 致谢

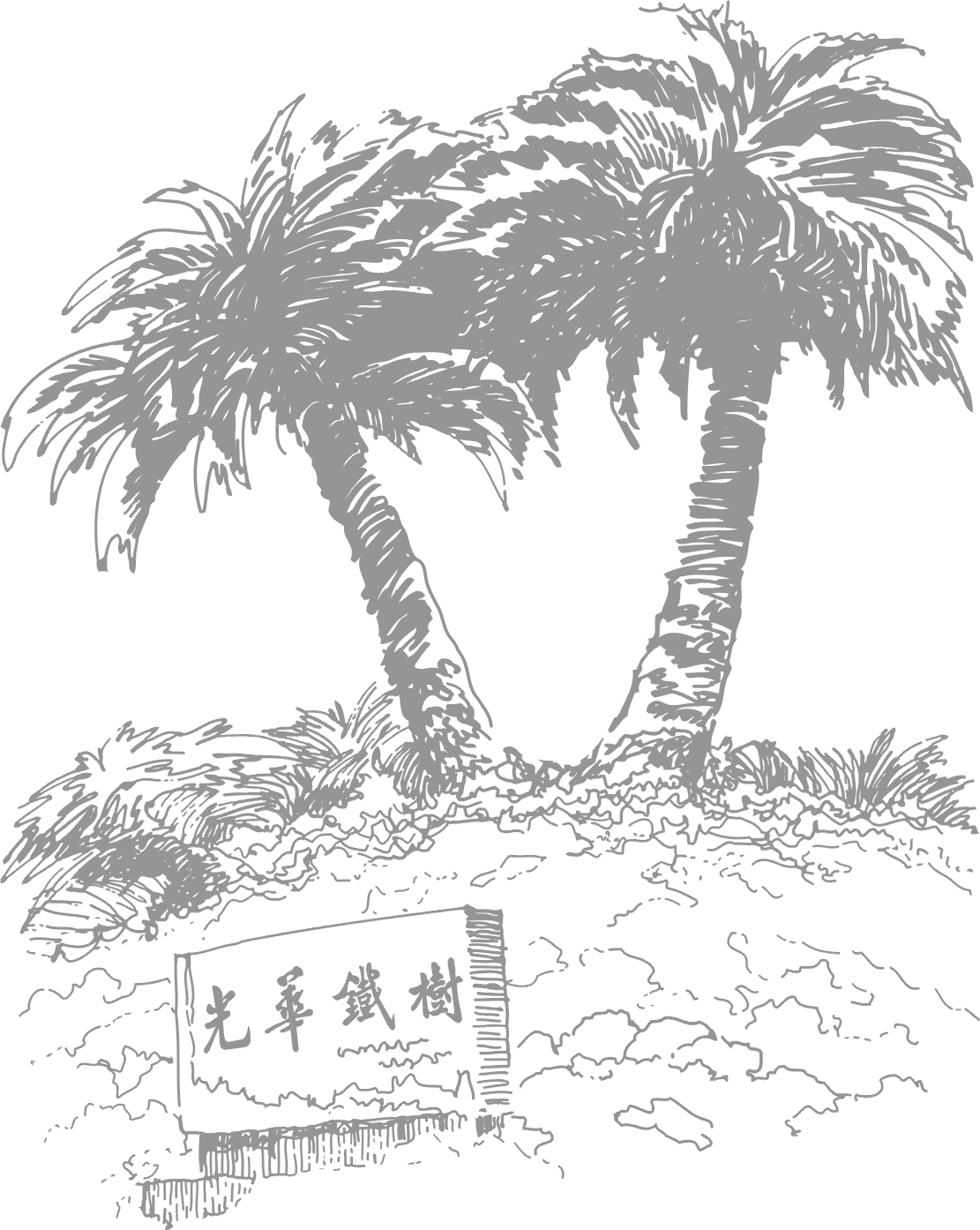
随着大学四年的学习生活渐渐走向终点，毕业季悄然而至。在这段时光的学习与成长中，我深受老师、同学及朋友们的帮助与支持，借此论文完成之际，我想向所有在这个过程中给予我支持和帮助的人表达我深深的感激和最好的祝愿。

首先，我要特别感谢我的论文指导老师XXX老师。我在大二时期的XXXX课程上首次遇见X老师，他认真细致、温和耐心以及极具责任心的教学态度给我留下了深刻的印象。在此次毕业论文的撰写过程中，X老师不仅给予了我鼓励与支持，还在选题、研究方向、模型搭建和实验设计等方面提供了极为宝贵的指导，他的付出对我而言意义重大。同时，我也要对XXX学长表达我的感激之情，在我遇到难题时，他总能指引我前进的方向，并耐心解答我的疑惑，帮助我顺利推进研究工作。在此，向XXX学长致以诚挚的谢意和良好的祝愿！

接下来，我要感谢我的同学和好朋友XX。在我面临困难和压力时，他总是及时地给我鼓励和支持，帮助我缓解压力。感谢你一直以来的陪伴和支持！

最后，我要感谢学校所有老师的精心教导，以及所有参与论文评审和答辩的老师们的辛勤工作。

在此，我对所有给予我帮助和支持的老师、同学及朋友们表达最深的感激，同时预祝大家未来光明、万事如意！



经世济民

孜孜以求