



福建农林大学

硕士学位论文

基于 VNF 实例共享的服务功能链部署与动态调整算法研究

学 科 门 类: 工学

一级学科名称: 计算机科学与技术

二级学科名称:

研 究 方 向: 计算机网络与信息安全

论 文 编 号:

论文完成时间: 二〇二四年四月

Research on Service Function Chain Deployment and Dynamic Adjustment Algorithm Based on VNF Instance Sharing

**A Dissertation Submitted to
Fujian Agriculture and Forestry University
in Partial Fulfillment of the Requirements for
Master Degree of Engineering**

**College of Computer and Information Science
Fujian Agriculture and Forestry University
Fujian, P.R. China**

Submission Date: April, 2024

独创性声明

本人声明，所呈交的学位（毕业）论文，是本人在指导教师的指导下独立完成的研究成果，并且是自己撰写的。尽我所知，除了文中作了标注和致谢中已作了答谢的地方外，论文中不包含其他人发表或撰写过的研究成果。与我一同对本研究做出贡献的同志，都在论文中作了明确的说明并表示了谢意，如被查有侵犯他人知识产权的行为，由本人承担应有的责任。

学位（毕业）论文作者亲笔签名：

日期：

论文使用授权的说明

本人完全了解福建农林大学有关保留、使用学位（毕业）论文的规定，即学校有权送交论文的复印件，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，在 年后解密可适用本授权书。 ☐

不保密，本论文属于不保密。 ☐

学位（毕业）论文作者亲笔签名：

日期：

指导教师亲笔签名：

日期：

目 录

图 录.....	III
表 录.....	IV
摘 要.....	V
ABSTRACT.....	VII
1 绪论.....	1
1.1 研究背景和意义.....	1
1.2 研究现状.....	3
1.2.1 SFC 部署	3
1.2.2 SFC 资源动态调整	5
1.2.3 研究现状小结.....	6
1.3 研究内容.....	6
1.4 论文章节安排.....	8
2 相关理论基础.....	10
2.1 软件定义网络.....	10
2.2 网络功能虚拟化.....	11
2.3 服务功能链.....	13
2.4 深度强化学习.....	13
2.5 本章小结.....	15
3 基于 VNF 共享的服务功能链部署算法.....	16
3.1 引言.....	16
3.2 问题描述与建模.....	16
3.2.1 问题描述.....	16
3.2.2 系统模型.....	17
3.2.3 部署优化模型.....	18
3.3 问题求解.....	20
3.3.1 算法描述.....	20
3.3.2 算法复杂度分析.....	24
3.4 仿真设计及结果分析.....	24
3.4.1 实验方法及仿真参数设置.....	25
3.4.2 对比算法介绍.....	25
3.4.3 评价指标.....	26
3.4.4 实验结果与分析.....	26

3.5 本章小结.....	28
4 基于深度强化学习的服务功能链动态调整算法.....	29
4.1 引言.....	29
4.2 问题描述与建模.....	29
4.2.1 问题描述.....	29
4.2.2 系统模型.....	30
4.2.3 资源调整优化模型.....	32
4.3 问题求解.....	34
4.3.1 算法描述.....	34
4.3.2 算法复杂度分析.....	39
4.4 仿真设计结果及分析.....	39
4.4.1 实验方法及仿真参数设置.....	40
4.4.2 对比算法介绍.....	41
4.4.3 评价指标.....	41
4.4.4 实验结果与分析.....	42
4.5 本章小结.....	46
5 结论与展望.....	48
5.1 主要工作总结.....	48
5.2 创新点.....	48
5.3 不足与展望.....	49
参考文献.....	51
攻读学位期间的学术论文与研究成果.....	55

图 录

图 1-1 传统网络与 NFV 网络结构	2
图 1-2 VNF-FG.....	4
图 1-3 研究内容	7
图 1-4 论文的组织结构与章节安排	9
图 2-1 SDN 架构	10
图 2-2 NFV 框架	12
图 2-3 DDNQ 训练过程.....	15
图 3-1 无 VNF 实例共享&有 VNF 实例共享	17
图 3-2 VNF 实例共享引发链路增长	17
图 3-3 运营成本	26
图 3-4 链路转发成本	27
图 3-5 VNF 部署成本	27
图 3-6 VNF 利用率	28
图 3-7 服务接受率	28
图 4-1 动态调整示例	30
图 4-2 物理资源分布	42
图 4-3 在 50 条 SFC 下物理资源分布	43
图 4-4 总成本	43
图 4-5 实例化成本	44
图 4-6 迁移成本	44
图 4-7 接受率	45
图 4-8 总成本	46
图 4-9 长期收益	46

表 录

表 3-1 参数及意义	18
表 3-1 参数及意义（续）	19
表 3-2 关键参数和值	25
表 4-1 参数及意义	32
表 4-2 关键参数和值	40
表 4-3 SFC 相关参数	40
表 4-4 ISFCDA 相关参数	41

摘 要

网络功能虚拟化 (Network Function Virtualization, NFV) 的出现, 使虚拟网络功能 (Virtual Network Function, VNF) 可以部署在云服务提供商的数据中心, 通过将网络功能和专用硬件解耦, 以提高网络服务的灵活性和可扩展性, 但这也意味着网络的编排更加复杂, NFV 编排是当前一个挑战, 包括如何对服务功能链 (Service Function Chain, SFC) 进行低成本部署, 以及在部署后, 面对网络需求波动时, 对 SFC 进行资源动态调整。目前, 关于 SFC 部署主要集中在优化部署成本。现有研究主要通过减少 VNF 实例的数目以降低 VNF 部署成本, 或者通过优化流量转发路径来降低转发成本, 但在实际的网络中, 综合考虑二者是有必要的。而关于 SFC 动态调整, 现有研究往往只是单方面进行垂直或者水平缩放, 并且没有考虑调整后的资源分布, 从而导致后续较多的服务需求无法得到满足。

针对以上问题, 本文拟提出基于 VNF 实例共享的 SFC 部署算法以及 SFC 动态调整算法。本文考虑到 SFC 部署时的时延要求, 并指出现有的 VNF 部署机制在最小化端到端时延的同时忽略了部署成本问题。本文首先建立多目标和多约束的问题模型, 接着提出了一种启发式算法, 预估路径部署长度上限, 保证 SFC 时延需求, 并在路径部署长度限制范围内, 尽可能使 VNF 实例共享最大化, 以平衡链路转发成本和 VNF 部署成本, 最终得到 SFC 部署策略。与已有的 SPH (Shortest Path Heuristic) 和 GUS (Greedy on Used Server) 部署算法相比, 本文算法所得的总运营成本分别降低 6.6% 和 12.15%, 且当 SFC 数量增多时, 本文算法的服务接受率可达 89.33%。

针对 SFC 资源动态调整的问题, 本文提出一种智能 SFC 动态调整算法 (Intelligent Service Function Chains Dynamic Adaptation Algorithm, ISFCDA)。首先, 建立以最小化长期调整开销成本和资源分布标准差为目标, SFC 时延以及物理资源为约束的整数线性规划 (Integer Linear Programming, ILP) 模型, 并将优化过程转变为马尔可夫决策过程 (Markov Decision Process, MDP)。其次, 针对资源调

整过程中,面临状态和动作空间过于庞大,以及状态转移概率未知等问题,本文提出了一种基于深度强化学习的 SFC 动态调整算法,最终得到近似最优的调整策略。与已有的算法相比,本文 ISFCDA 平均资源分布标准差最高降低了 9.90%,而 ISFCDA 的平均接受率最高提高了 39.57%,平均长期收益最高提高了 42.92%。

关键词: 服务功能链, 虚拟网络功能, NFV 编排, 多目标优化

ABSTRACT

The advent of Network Function Virtualization (NFV) has enabled Virtual Network Function (VNF) to be deployed in the data centers of cloud service providers, decoupling network functions from specialized hardware to enhance the flexibility and scalability of network services. However, this also means that network orchestration becomes more complex. NFV orchestration is currently a challenge, including how to deploy Service Function Chain (SFC) at a low cost, as well as how to dynamically adjust the physical resources of SFCs in response to fluctuations in network demand. Presently, research on SFC deployment mainly focuses on optimizing deployment costs. Existing research mainly reduces VNF deployment costs by reducing the number of VNF instances or reducing forwarding costs by optimizing traffic forwarding paths. However, in actual networks, it is necessary to consider both aspects. Similarly, when it comes to the dynamic adjustment of SFC, much of the existing research tends to focus solely on either vertical or horizontal scaling strategies. This singular focus often overlooks the crucial aspect of resource distribution following the adjustment. Consequently, such an approach can lead to challenges in meeting the demands of subsequent service requests due to the suboptimal allocation of network resources.

In response to these issues, this thesis proposes an SFC deployment algorithm based on VNF instance sharing and an SFC dynamic adjustment algorithm. This study takes into account the delay requirements and deployment cost during SFC deployment, which is different from current VNF deployment mechanisms that ignore deployment cost issues while minimizing end-to-end latency. This thesis first establishes a multi-objective and multi-constraint problem model, then proposes a heuristic algorithm that estimates the upper bound of path deployment length, ensuring SFC delay requirements are met. Within the constraints of path deployment length, it aims to maximize VNF instance sharing to balance link forwarding costs and VNF deployment costs, ultimately determining an SFC deployment strategy. Compared with existing Shortest Path Heuristic (SPH) and Greedy on Used Server (GUS) deployment

algorithms, the algorithm presented in this thesis reduces total operational costs by 6.6% and 12.15% respectively, and achieves a service acceptance rate of 89.33% as the number of SFCs increases.

For the SFC resource dynamic adjustment issue, an Intelligent Service Function Chain Dynamic Adjustment Algorithm (ISFCDA) is proposed to address the above challenges. Firstly, an Integer Linear Programming (ILP) model is established with the objective of minimizing the long-term adjustment cost and reducing the imbalance in resource distribution. Then, we transform the optimization process into a Markov Decision Process (MDP). Secondly, to solve the problems that the state space and the action space are too large and the state transition probability is uncertain in MDP, a dynamic adjustment algorithm based on deep reinforcement learning is proposed. This algorithm can obtain an approximate optimal adjustment strategy for the ILP model. Compared with the existing algorithms, the average standard deviation of resource distribution of ISFCDA is reduced by up to 9.90%, the average acceptance rate of ISFCDA is improved by up to 39.57%, and the average long-term profit is improved by up to 42.92%.

KEY WORDS: service function chain, virtual network function, NFV orchestration, multi-objective optimization

1 绪论

1.1 研究背景和意义

在当前互联网技术高速发展的时代,用户的网络服务需求正变得日益多元化与复杂化。传统的网络架构依赖于一系列专用的硬件设备来实现网络功能,包括但不限于防火墙、网络地址转换器和负载均衡器等。这些专用设备的网络服务与其硬件紧密关联,造成了服务提供商在应对多变的服务需求时遇到诸多挑战。由于不同专用设备的标准化程度存在差异,这大大限制了网络配置的灵活性,加之这些设备的日常维护和管理代价高昂,导致网络运营的成本负担进一步加剧。此外,随着 5G/6G 网络技术的快速发展,垂直行业服务(如智能制造、远程医疗等)需求的显著增长,原有基于专用硬件设备的网络服务模式面临着前所未有的挑战。这些新兴的网络服务不仅对网络的灵活性和可扩展性提出了更高的要求,而且对网络的实时响应能力和服务质量(Quality of Service, QoS)有着严格的要求。然而,传统网络架构中的专用硬件设备由于硬件依赖性强、配置灵活性低,难以迅速适应快速变化的网络服务需求,导致无法高效地支持新兴网络应用,这不仅大大增加了网络运营成本,还限制了网络服务的创新。

针对传统网络架构中面临的种种挑战,欧洲电信标准协会(European Telecom Standards Institute, ETSI)提出了一种革命性的网络技术解决方案——网络功能虚拟化(Network Function Virtualization, NFV)技术^[1]。NFV 技术的核心理念在于利用虚拟化技术,将那些原本固定在专用硬件上的网络功能软件化,使得这些网络功能能够以虚拟网络功能(Virtual Network Function, VNF)的形式运行在通用的计算平台上,例如行业标准的服务器和云基础设施中。这一转变不仅极大提升了网络服务的部署灵活性,也为网络管理和服务创新带来了新的可能。通过 NFV,网络服务提供商可以在不依赖于物理硬件的前提下,快速部署和扩展网络服务,而无需为每一项网络服务投资购买、维护专用的硬件设备。这种方式显著降低了网络运营的资本开支(Capital Expenditure, CAPEX)和运营费用(Operating Expense, OPEX)^[2],因为它减少了对昂贵专用硬件的依赖,简化了网络设备的维护和管理,同时提高了资源利用率。如图 1-1 所示,通过使用 NFV 技术,传统的网络功能设备能够以虚拟网络功能的形式在行业标准服务器上实现,使虚拟化的网络服务可以广泛的应用到各种网络场景,无论是数据中心网络、5G 网络,还是云计算等场景, NFV 技术都能够支持多种网络功能的虚拟化和灵活配置,满足不同应用场景和服务需求的多样性,展现了其巨大的潜力和优势^[3]。

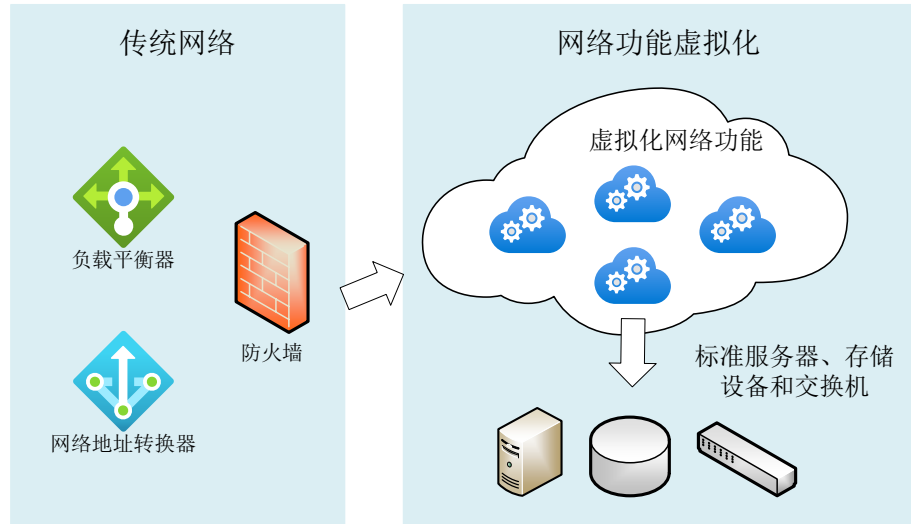


图 1-1 传统网络与 NFV 网络结构

尽管 NFV 技术极大地提升了网络服务的灵活性与成本效率，其所带来的网络编排复杂度的显著提升，无疑对网络管理提出了前所未有的挑战。此时，软件定义网络（Software Defined Networking, SDN）^[4]技术的灵活性和可编程性成为了解决这一问题的关键。SDN/NFV 技术的融合不仅仅是技术的叠加，而是为网络服务的部署与管理提供了一种全新的视角。在 SDN/NFV 的支撑下，网络服务不再是静态配置，而是可以根据实时需求，将不同的 VNF 通过服务功能链（Service Function Chain, SFC）按照特定的顺序动态连接，形成能够高效处理特定网络服务请求的 VNF 转发图（Virtual Network Function Forwarding Graph, VNF-FG）^[5]。然而，正是这种灵活性和动态性，加剧了编排的复杂性。NFV 管理器会根据 SFC 需求和物理设备容量限制，在标准通用服务器上部署 VNF 实例^[6]，考虑如何在保证服务质量的同时，合理地利用物理资源。并安排 VNF 的动态执行顺序，进行 SFC 调度^[7]，以减少总服务时间。同时由于 SFC 需求的不断变化^[8]，还需实现网络服务的资源动态调整。然而，在现有的 SDN/NFV 架构中，如何在不断变化的网络需求和物理资源限制之间找到最优解，灵活而高效地部署 SFC 和动态调整资源，降低运营成本，同时确保服务的连续性和可靠性，仍然是一个未被充分解决的问题。

鉴于此，本研究聚焦于探索高效的 SFC 部署和动态调整策略，旨在寻找一种能够平衡资源利用率和服务质量的最佳解决方案。通过深入分析现有 SDN/NFV 架构中的编排管理流程，并结合最新智能算法，本研究试图提出一种创新的编排管理框架。利用资源可共享特性，在保证用户网络的 QoS 的前提下（如满足时延、带宽需求），制定一种基于 VNF 实例共享的 SFC 部署算法上。通过分析网络服务的时延要求和资源利用的效率，制定出一套既能满足服务要求又能最大化资源共享的 VNF 部署策略，实现链路成本和 VNF 部署成本的最小

化。其次，在 VNF 初步部署后，面对网络负载和需求的动态变化，本研究将探索一种智能的 SFC 资源动态调整算法，最小化调整成本同时实现 VNF 的自适应调整，服务更多的服务请求。通过这种方式，能够保证即便在网络状态大幅波动的情况下，也能保持网络服务的稳定性和高效性，从而最大化资源的利用率。总之，通过本研究的开展，希望为网络服务提供商提供一个既能满足当前网络环境下服务需求，又能预见未来网络技术发展趋势的网络服务部署和管理解决方案。这不仅有助于降低网络运营商的运营成本，提高网络服务的质量和效率，还将推动整个网络技术的创新和发展，为用户提供更加丰富多样和高质量的网络服务。

1.2 研究现状

在过去的几年中，随着 NFV 和 SDN 技术的发展，SFC 的概念逐渐成为网络架构设计中的一个重要组成部分。学术界和工业界都对 SFC 的部署和动态调整进行了大量的研究，但仍然面临着许多挑战，例如最小化部署成本和链路转发成本、以及如何在保证网络服务质量的前提下优化资源分配，减少资源碎片等。本节将综述 NFV 编排管理过程中的 SFC 部署和资源动态调整的相关工作。

1.2.1 SFC 部署

SFC 部署即当一条网络服务请求 SFC 到达时，需要根据 SFC 的资源需求，底层物理设备的资源，选择合适的部署策略，在物理设备上进行 VNF 实例化，分配资源并进行路由。这一过程需要综合考虑多种因素，包括 VNF 的性能要求、网络的带宽资源、延迟限制以及设备的计算能力等。有效的 SFC 部署策略不仅可以减少网络的运营成本，还可以提升服务的响应速度和用户体验。为了应对 SFC 的部署映射问题，当前研究主要集中于在特定优化目标指导下进行 SFC 部署，确保满足用户的资源需求和 QoS。其中最常见的 QoS 需求为时延需求，而优化目标常为增加网络服务供应商的经济效益或减少资本支出。

对于用户的时延需求，文献[9]将两个 VNF 之间的跳数作为传输延迟。文献[10]将物理链路端到端传输延迟的总和视为请求的延迟。文献[11]提出一种基于深度强化学习的 SFC 映射算法，降低网络平均时延和部署失败率。文献[12]中，通过避免物理链路上的流量过载情况来降低请求延迟。然而，这些研究只考虑了 SFC 为单链的情况，很少有研究考虑其他类型的 VNF-FG，常见的 VNF-FG 如图 1-2 所示。文献[13]考虑了多播网络服务的 VNF 放置，但是其中 VNF-FG 是仅具有一个源端和几个目的端的特例，且没有考虑延迟的要求。

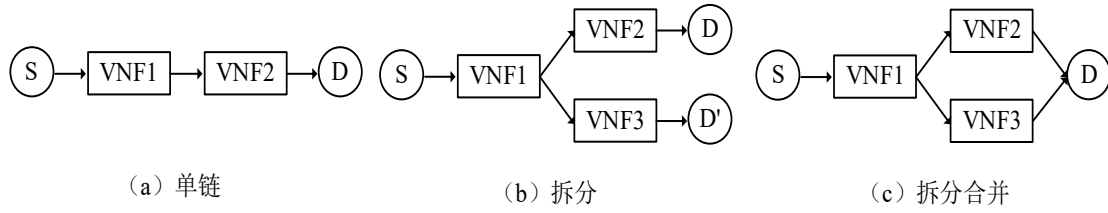


图 1-2 VNF-FG

对于增大经济效益或减少成本作为优化目标的，文献[14]建立基于马尔可夫决策的随机优化模型，优化 SFC 部署成本。文献[15]通过最小化活动虚拟机的数量，降低了部署成本。文献[16]提出了一种基于加权图匹配的匹配算法，以提高资源利用率，然后通过 VNF 迁移方法关闭空闲服务器，以降低服务器成本。文献[17]针对实时 VNF 部署和迁移，使用供需模型量化 VNF 干扰，提出一种实时自适应感知算法以最大化总收益。文献[18]提出一种在线扩展算法调整 VNF 实例部署，以平衡 VNF 运营成本，VNF 实例维护成本和 VNF 部署成本，但没有考虑请求时延，忽略了用户的 QoS。文献[19]提出基于时间窗的虚拟网络批处理映射策略，以减少虚拟网络映射成本和功耗。文献[20]通过减少部署网络服务的计算和带宽资源消耗，来降低运营成本。文献[21]提出了一种在边缘网络下最小化部署成本的部署方案。但这些研究，主要关注优化 VNF 实例的数目以减少 VNF 实例化成本，或者优化流量转发路径来减少带宽成本，优化目标仅为单个，而没有综合考虑路径转发成本以及 VNF 部署成本。而事实上链路转发成本和 VNF 实例部署成本此消彼长，因此作为目标函数共同考虑至关重要。

目前 SFC 部署问题在现有工作中，可以定义为具有不同目标的优化问题，为此提出了不同的解决方案。现有工作中，精确求解方案可以提供最优解，但其很大程度受网络规模的影响，如文献[22]利用 CPLEX 等计算工具得到小规模网络下的精确求解，但在大规模网络下运算会导致执行时间过长。而启发式算法可以在短时间内提供近似最优解，且通常不受网络规模的影响。现有工作为了实现优化目标，通常会将线性规划，如整数线性规划 (Integer Linear Programming, ILP) 或者混合整数线性规划 (Mixed Integer Linear Programming, MILP)，结合启发式算法进行求解。例如，文献[23]建立整数线性规划模型，在有线和无线混合网场景下，基于资源感知的节点选取算法和带宽最小的虚拟链路映射方案，提出了一种面向移动边缘计算网络的 SFC 高效可靠的部署方法。文献[24]为了实现服务功能链的低成本高效率部署，提出了一种基于改进麻雀搜索算法的服务功能链优化映射算法，但其容易陷入局部最优。文献[25]提出了一个联合 SFC 部署和路由的优化问题。鉴于该问题是 NP-Hard，其作者设计了一种候选路径选择的近似算法，提升了求解效率。

1.2.2 SFC 资源动态调整

在 SFC 初步部署后,网络请求的波动可能使原部署方案无法满足用户需求,因此,对于网络服务提供商来说,在动态变化的网络环境中,保证用户质量的前提下,最小化调整成本同时实现 VNF 的自适应调整是非常关键的。现存的研究中,SFC 动态调整主要分为三种:(a)水平缩放(b)垂直缩放(c)迁移。如文献[26]通过增加或减少 VNF 实例的数量来进行水平缩放。这种策略可以快速响应网络请求的波动,并提供较高的弹性和扩展性。但随着 VNF 实例数量的增加,配置和管理的复杂性也会增加,QoS 难以保证。文献[27]为了有效控制资本支出,对多核 CPU 进行垂直缩放。具体来说,随着数据流量的增加或减少,VNF 实例的计算能力也相应增加或减少。但如果底层物理设备的资源有限或不够灵活,垂直缩放可能受到限制,无法满足动态变化的需求。文献[28]提出了一种迁移算法,通过迁移已部署的 VNF 实例,可以使其与新的网络流量或新的网络功能实现集成,但只是针对 SFC 中部分 VNF,并且忽略了剩余资源的分布。这三种策略各具特点,但在单独应用时可能存在一些限制和局限性。在面临众多请求的情况下,有限的物理节点数量可能难以通过水平扩展来满足需求。此外,引入新的实例可能会导致更复杂的路由路径选择,进而难以达到所需的延时标准。同时,采取这种策略往往会引起资源使用上的碎片化,降低整体资源利用效率。

除此,现有工作中,如文献[29]提出基于最短服务中断时间的标准来选择 VNF 的水平缩放方式,从而实现最小的 VNF 停机时间和 SFC 延迟。文献[30]有效地利用多核架构,结合预测模型,通过垂直缩放,来达到最大的吞吐量和最佳的性能表现。文献[31]通过迁移 VNF 来进行聚合,从而减少数据中心的能源消耗。文献[32]通过迁移部分 VNF,在满足低时延服务下降低网络过载情况。但研究者都是单独的策略进行研究和优化,很少有联合考虑三种策略。然而,在实际网络中,综合考虑它们可能会带来更好的结果,以达到更好的资源分配和网络优化效果。为此文献[33]提出了一种启发式算法考虑了三种策略,降低了开销和能源消耗,但是作者并没有考虑剩余资源的分布情况,这可能导致资源利用不均衡的问题。

而现有工作中,各研究者进行动态调整的目标也是不同的。文献[34]目标是最小化能源消耗和重配置成本,但该工作是在具有固定流量和有限的状态-动作空间下进行求解。文献[35]设计了一种可用性感知的资源分配和重配置框架,重点是以最低的重新配置成本提供最高的可用性改进。文献[17]提出了一种延迟迁移策略,通过迁移最后一个 VNF 来优化物理资源分布,虽然在一定程度上减轻高负荷节点的负担,但并没有从全局的角度考虑资源分配的均衡性。

除此现有工作对调整方案的求解也不同。文献[36]提出了一种 ILP 方法和一种贪婪启发式方法来动态扩展资源,但在处理大规模网络时的计算复杂性较高,

更适合在小规模的网络下进行。文献[37]利用一种分层辅助图的方法重新配置 VNF 服务树，降低了迁移开销和重配置后的总资源占用。文献[38]以服务提供商的利润最大化为目标，以调整成本为界，提出了一种基于 CG（Column Generation）的方法来解决该问题。但传统的方法往往依赖于经验规则和启发式策略，而深度强化学习可以通过学习和优化过程，使智能体自主地探索和学习最佳的决策策略。这种基于深度强化学习的方法为网络调整提供了更灵活和智能的决策方式。

1.2.3 研究现状小结

针对上述 SFC 部署和资源动态调整的研究现状，可以总结如下：

（1）在 SFC 部署方面，虽然已有众多研究提出不同的优化目标和解决方案，但大多数研究依旧集中在单一目标的优化上，如降低部署成本、最小化链路转发成本或满足用户特定的服务质量需求。这种单一目标的优化方法虽然在特定情境下能取得较好的性能，但在实际应用中往往难以全面适应多变的网络需求和环境。此外，大部分研究集中于单链 SFC 场景，而对于复杂 VNF-FG 的场景探索较少，这限制了研究的通用性和实际应用的范围。

（2）针对 SFC 资源动态调整的研究现状，现有研究主要分为垂直缩放和水平缩放，以及通过 VNF 迁移来适应资源需求的变化。尽管这些策略在一定程度上能够应对网络需求的变化，但大多数方法仍然是基于单一策略进行优化。目前少有研究同时考虑这三种策略的联合应用，以及如何在保持资源分布均衡的同时，最小化调整成本并保证网络服务质量。此外，对于大规模网络环境，现有的调整方案面临计算复杂度高、实时响应能力不足的问题，亟需更高效、更智能的调整算法。

1.3 研究内容

通过对上述研究背景和研究现状的分析，以及期望能够为网络服务提供商提供一种高效、灵活且成本效益高的网络服务管理方案，从而更好地满足用户需求和网络服务的快速发展，本文在 SDN/NFV 技术支持下，根据虚拟化设施管理器以及 VNF 管理器中虚拟资源和 VNF 相关信息，首先针对 SFC 的部署问题，利用优化模型建立模型，开展问题研究。其次面对网络负载发生变化，对资源动态调整进行研究。如图 1-3 所示，根据真实的网络数据分别获得网络的拓扑结构和流量需求，由此可以获得仿真物理物理网络 and 用户需求。本文拟建立用户需求模型和物理资源模型，并将模型输入 NFV 编排管理系统，而编排系统根据本文设计的部署方案和动态调整策略，对 SFC 进行部署和动态调整。基于此，本文研

究内容有以下几点：

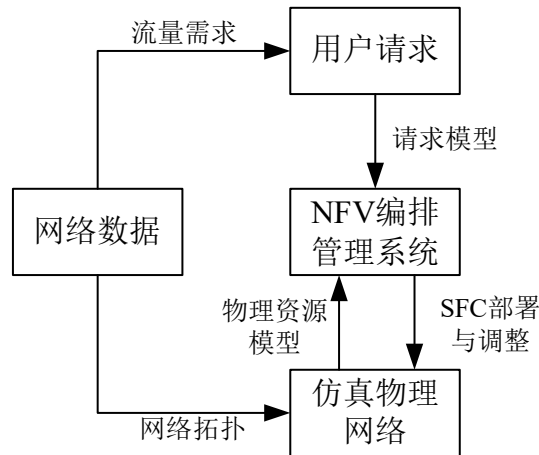


图 1-3 研究内容

(1) SFC 部署方案。随着 5G 网络的出现，更多的网络服务出现，例如自动驾驶，智慧医疗。这些服务对于时延要求极为敏感，所以 VNF 实例部署时，必须考虑其延迟要求。在当前的 VNF 部署策略中，尽管很多研究注重于优化 SFC 的整体延迟问题，却忽略了 VNF 部署所涉及的成本问题。这种做法在追求最小化端到端延迟的同时，可能无意中提高了网络运营的总成本。这是因为，在努力减少延迟时，VNF 实例往往被放置在距离源点和终点最近的路径上，导致某些服务器资源被低效利用。此外，如果物理服务器具备足够的处理能力，允许多个服务链共享同一类型的 VNF 实例不仅可以增强资源使用效率，还能减少部署成本。然而，这种共享机制可能引致路径长度的增加，增大了数据传输成本，并有可能不满足 SFC 的延迟要求。显然，在 NFV 编排中，VNF 实例的部署与链路分配之间的动态协调显得尤为关键。因此，本文第一个待解决的问题是如何在降低部署和转发成本的同时，确保满足服务链的资源需求和时延要求，实现成本与性能之间的最优平衡。本文拟首先将问题建模为多目标和多约束问题模型，该模型以最小化 VNF 实例部署成本和链路转发成本为目标，同时受限于每条 SFC 时延、物理链路带宽资源和服务 CPU 资源约束，其次为每条 VNF 转发路径（Virtual Network Function-Forwarding Path, VNF-FP）设置路径长度上限，以平衡链路和 VNF 实例的使用，接着提出基于 VNF 共享的服务功能链部署算法，联合 SFC 放置和路径选择问题，以节省链路转发成本并有效地重用 VNF 实例。最后，通过仿真实验验证部署策略的有效性和优越性。

(2) SFC 动态调整策略。随着网络流量的增大，流量波动也变得复杂，这使得 VNF 实例不能一直保持在最佳状态。因此，每个 VNF 实例所占用的资源应该随着流量的变化而进行动态调整，从而在面对网络中资源空闲或满载状态时，可以有效地均衡资源，避免资源的浪费或者网络的拥塞，保证 QoS 的前提下，减少运营成本。现有工作中，资源调整主要分为：(a) 水平缩放 (b) 垂直缩放 (c)

迁移。这三种策略各具特点，但在单独应用时可能存在一些限制和局限性。当有大量请求时，较少的物理节点是无法满足水平缩放的。而为了创建新的实例，可能需要更长的路由路径，从而导致无法满足时延，且这个方法容易产生较多的资源碎片，不利于服务更多的请求。现有工作很少同时考虑三种策略，并且很少考虑到调整后资源的分布情况。因此，本文拟将问题形式化为 ILP 问题，并将优化过程转化为马尔可夫决策过程，联合水平、垂直缩放和迁移，提出了一种基于深度强化学习的智能调整算法，以实现网络资源的平衡分配，同时最小化相关的调整成本并满足用户需求。最后，通过与现有的调整策略进行对比分析，评估本文所提算法的性能优势和实用性。

本研究的开展，旨在为网络服务提供商提供一种新的视角和工具，以应对快速变化的网络服务需求和复杂的网络环境。通过深入探索基于 SDN/NFV 技术的 SFC 高效部署和动态调整策略，本文不仅有助于提升网络服务的灵活性和效率，还能显著降低网络运营的成本。

1.4 论文章节安排

本文通过五个章节的结构详细阐述了面向 5G/6G 网络的 NFV 编排的研究工作，包括对相关理论的深入探讨、提出的创新算法及其验证。具体的论文结构如图 1-4 所示。

第一章：绪论。本章节为研究工作提供了背景，强调了 5G/6G 网络环境下 NFV 编排管理的重要性和迫切性。通过分析现有的研究现状，本章明确指出了网络服务功能链部署和资源动态调整在当前网络技术发展中所面临的主要挑战和研究缺口。此外，本章还概述了本文的主要研究内容，并以清晰的结构安排，呈现了一个全面的研究框架概览，为后续章节的深入讨论奠定了基础。

第二章：相关理论基础。在这一章中，本文深入探讨了 SDN 和 NFV 技术的基础知识，详细描述了这两种技术的架构和工作原理，为理解后续章节中提出的算法和策略提供了必要的背景信息。同时，本章也详细介绍了服务功能链的相关概念及其在网络管理中的作用，并对深度强化学习算法进行了概述。

第三章：基于 VNF 共享的服务功能链部署算法。针对“NFV 编排管理中存在 VNF 实例部署成本和转发路径成本难以权衡”的问题，提出基于 VNF 实例共享的 SFC 部署算法。通过对问题的详细描述和 ILP 模型的建立，本章提出了一种启发式算法来获取有效的 SFC 部署策略，并通过与现有部署算法的对比实验，验证了所提方法的有效性和优越性。

第四章：基于深度强化学习的服务功能链动态调整算法。针对“由于网络波动，NFV 编排管理中存在 SFC 部署后资源无法满足”的问题，提出了基于深度

强化学习的 SFC 动态调整算法。本章首先给出了资源动态调整问题的描述，之后根据问题建立了 ILP 模型，并证明了问题是 NP-hard 的，并将优化过程转变为马尔可夫决策过程，并针对资源调整过程中，存在状态和动作空间过大，以及状态转移概率未知等问题，该文提出了一种基于深度强化学习的 SFC 动态调整算法，最后通过仿真实验来证明本文所设计的调整策略的有效性。

第五章：结论与展望。最后一章对全文的工作进行了总结，并对未来可能的研究方向进行了展望。本章不仅回顾了本文的主要研究成果和贡献，还讨论了在 SFC 部署和动态调整方面存在的潜在研究机会，为未来在这一领域的研究工作提供了指导和灵感。

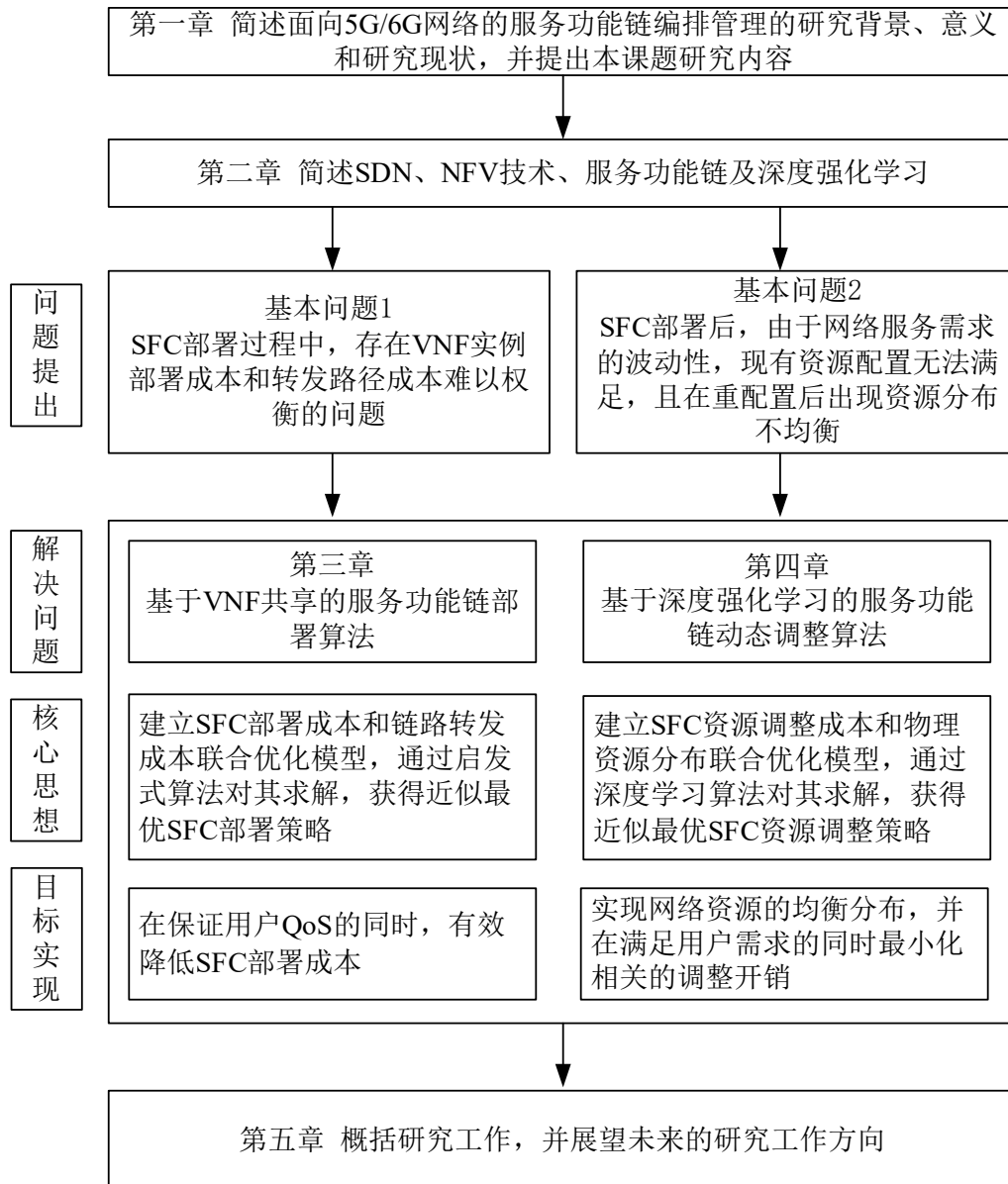


图 1-4 论文的组织结构与章节安排

2 相关理论基础

本节旨在详细阐述与本研究相关的核心技术和理论基础。遵循从网络控制和管理整体架构出发，逐步深入到网络服务的具体实现和优化的逻辑，对软件定义网络、网络功能虚拟化等网络技术进行介绍。此外，还对深度强化学习等算法进行了系统性介绍，为后续研究内容奠定理论基础。

2.1 软件定义网络

在传统网络中，每个网络设备（如交换机和路由器）都拥有自己的控制逻辑和转发功能，设备必须独立处理路由决策和数据包转发。而软件定义网络（Software Defined Network, SDN）是一种创新的网络架构，旨在通过软件化的方式提高网络的可编程性和灵活性。SDN 的核心思想在于将网络的控制平面（负责决策的部分）与数据平面（负责数据转发的部分）进行分离，并通过中心化的控制平面对整个网络进行统一管理。SDN 架构主要由以下三个基本组件构成^[39]，其架构如图 2-1 所示：

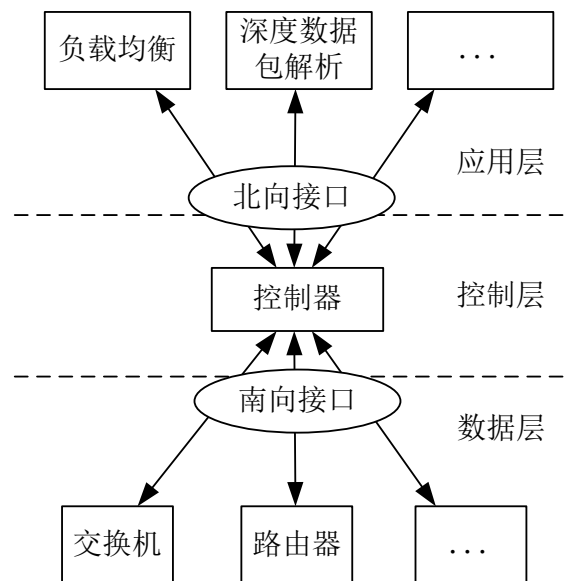


图 2-1 SDN 架构

（1）应用层：应用层位于 SDN 架构的最上层，它由运行于 SDN 控制器之上的各种网络应用和服务构成。这些应用覆盖了负载均衡、网络安全、网络监控和优化等多个领域，致力于实现网络的高级管理和配置。通过北向接口，应用层能够与控制层进行交互，获取网络状态信息，下发网络管理指令等。北向接口提供了丰富的 API，允许开发者和网络管理员根据需要，定制和部署特定的网络管

理和服务策略，以满足不同网络应用场景的需求。

(2) 控制层：控制层是 SDN 的心脏，由一个或多个 SDN 控制器组成。这些控制器为网络提供了一个逻辑上集中的控制点，通过抽象化底层的网络设备细节，简化了网络配置和管理过程。SDN 控制器使用南向接口，如广泛使用的 OpenFlow 协议，与数据层的物理和虚拟设备进行通信。控制器根据来自应用层的策略，计算出相应的路由决策、安全策略等，并通过南向接口向数据层设备下发流表规则，实现对数据流的精确控制和转发。

(3) 数据层：数据层位于 SDN 架构的底部，由物理和虚拟化的网络设备组成，如交换机、路由器等。这些设备的主要职责是根据控制层下发的流表规则，进行数据包的转发。数据层设备在 SDN 架构中被简化为可编程的转发设备，其转发逻辑和行为完全由控制层所控制。这种分离使得网络设备不再需要复杂的配置和管理，网络的变更和优化可以迅速地在控制层实现，大大提高了网络的灵活性和可编程性。

总之，SDN 架构通过应用层、控制层和数据层的紧密协作，不仅实现了对网络资源的高效管理和优化控制，还开启了网络技术创新和服务多样化的新篇章。已经被广泛应用于数据中心网络、企业网络和云服务中。随着 SDN 技术的不断成熟和发展，其在未来网络架构中的作用和影响将会越来越显著，为网络的智能化、灵活化和服务化提供了坚实的基础。

2.2 网络功能虚拟化

网络功能虚拟化（Network Function Virtualization, NFV）是一种网络架构概念，主要目的是将网络功能（如防火墙、负载均衡、网络地址转换等）从依赖专用硬件的传统架构中解耦出来，转移到通用计算平台上。这种转变使得网络服务能够以软件应用的形式运行在标准化的虚拟化环境中，如服务器、存储和云基础设施。通过虚拟化技术，NFV 提供了网络功能的快速部署和管理，从而大大提高了网络的灵活性和成本效率。NFV 架构包括以下三个关键组件^[40]，其框架如图 2-2 所示。

(1) 虚拟网络功能（Virtual Network Function, VNF）：VNF 的概念源自于将传统网络功能，如防火墙、DNS、负载均衡器等，从其物理硬件中解耦，并以软件形式实现。这些软件化的网络功能可以部署在标准化的服务器上，无论是在数据中心、网络边缘还是云平台。VNF 的灵活性使服务提供商能够快速响应网络需求变化，简化了网络服务的扩展和升级过程，同时也减少了物理设备的维护成本和复杂性。

(2) NFV 基础设施（NFV Infrastructure, NFVI）：NFVI 为 VNF 提供了必

要的运行环境，包括计算、存储和网络资源。NFVI 的核心是资源的虚拟化，它允许多个 VNF 共享同一物理资源，提高了资源的利用率和网络服务的弹性。NFVI 的部署位置极为灵活，可以分布在接近用户的网络边缘，或者集中在大型的数据中心，以满足不同的性能和延迟需求。

(3) 管理与网络编排 (Management and Orchestration, MANO)：主要包括三个模块：虚拟化基础设施管理、VNF 管理、编排管理。虚拟化基础设施管理负责管理和控制 NFVI，包括资源分配、资源调度、资源监控和故障管理等。虚拟化基础设施管理确保了 VNF 能够在适当的资源上高效运行，同时处理硬件故障和性能退化等问题。VNF 管理这一模块专注于 VNF 的生命周期管理，从 VNF 的初始部署、配置、监控到最终的销毁。VNF 管理器不仅确保 VNF 按照预定的服务质量运行，还负责故障恢复和性能优化。编排管理是实现网络服务自动化部署和管理的关键。它通过编排器自动化整个网络服务的生命周期，包括 SFC 的构建和网络策略的实施。编排器可以根据业务需求和资源可用性，动态地组合和配置 VNF，实现复杂网络服务的快速部署。

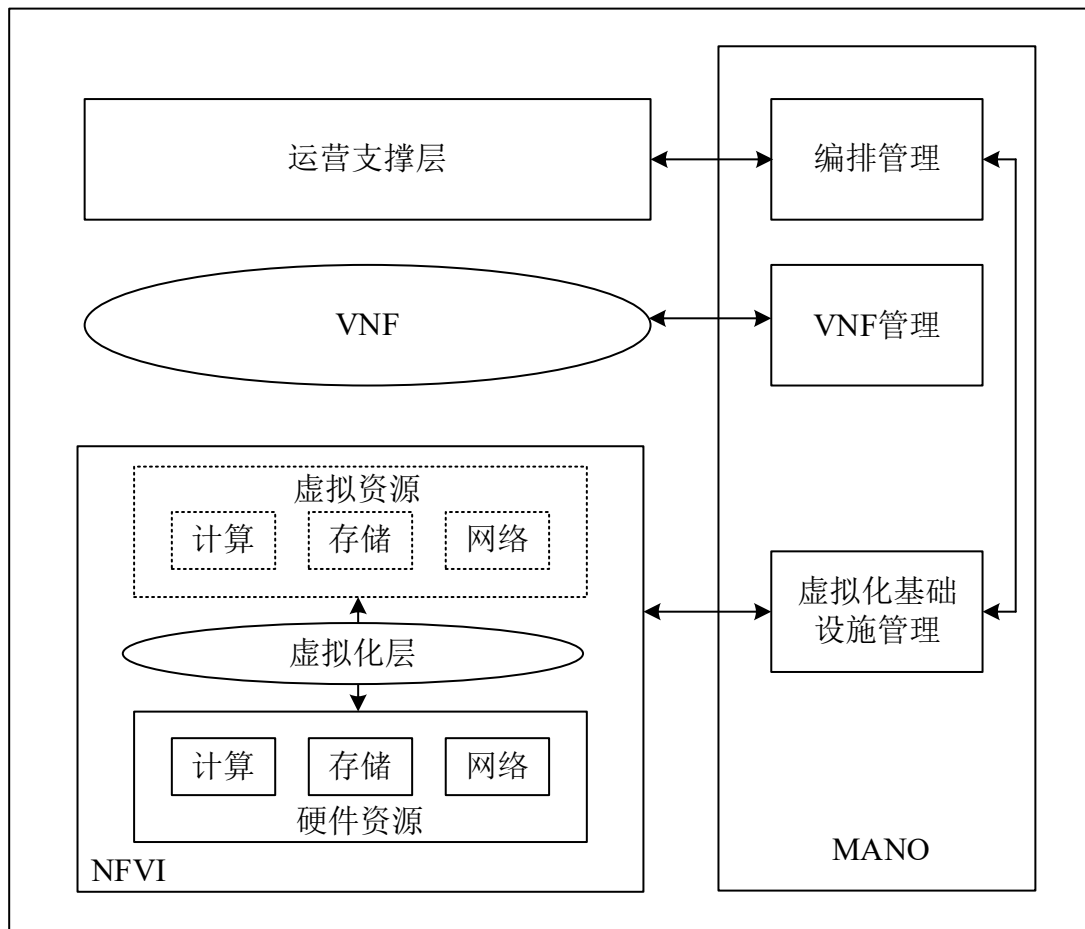


图 2-2 NFV 框架

NFV 和 SDN 虽然是两个独立的技术概念，但都旨在通过软件化和中心化的方式提高网络的灵活性和自动化管理水平，减少对专用硬件的依赖。NFV 侧重

于网络功能的虚拟化和部署，而 SDN 专注于通过中心化的控制层实现网络流量的控制和优化。SDN 的控制层可以为 NFV 提供动态的网络连接和服务保证，而 NFV 可以为 SDN 提供可编程和虚拟化的网络服务组件。二者的结合实现了更加灵活、可编程的网络，从而加快服务创新，降低运营成本。

2.3 服务功能链

服务功能链（Service Function Chain, SFC）指的是将不同的 VNF 按照特定的顺序链接起来，形成一个有序的服务链，以支持复杂的网络服务需求。SFC 的创建和部署通常依赖于 NFV 的 MANO 以及 SDN 控制器的协同工作。SDN 控制器负责整个网络的流量控制和路由决策，确保网络流量按照 SFC 规定的顺序经过所有配置的 VNF；而 NFV 管理系统则负责 VNF 的生命周期管理，包括部署、配置和监控等。通过 SDN 和 NFV 的结合，可以灵活创建和管理 SFC，实现流量的精确控制和优化。

2.4 深度强化学习

深度强化学习（Deep Reinforcement Learning, DRL）是将深度学习^[41]和强化学习^[42]这两种先进的机器学习技术相结合的方法，利用深度神经网络来逼近最优策略或值函数。它的核心目的是让智能体在与环境的交互中自主学习，以做出在特定任务中的最佳决策。在强化学习框架下，智能体通过观察环境状态并采取行动，根据从环境获得的奖励或惩罚来评估其行动的效果。智能体的终极目标是通过学习策略，最大化其获得的长期奖励。然而，传统强化学习在处理包含高维状态空间或需要复杂决策的任务时，往往会遇到难题。而深度学习，尤其是卷积神经网络和循环神经网络，擅长于从大规模且高维度的数据中提取有用的特征表示进行处理和学习。深度强化学习通过整合深度学习的强大数据处理能力，赋予了智能体处理更加复杂环境和学习更为复杂策略的能力。简言之，深度强化学习技术通过融合深度学习的特征提取与学习能力和强化学习的决策学习机制，开辟了解决复杂决策问题的新途径。这种方法已在多个领域，如游戏、自动驾驶、机器人控制及自然语言处理等，展现出了显著的应用潜力和优越性能。

（1）DQN（Deep Q-Network）：DQN 由 DeepMind 于 2013 年提出^[43]，是一种深度强化学习算法，它在解决 MDP 问题中具有重要意义。传统的强化学习方法在处理具有大状态空间和连续动作空间的问题时面临挑战，因为它们需要存储和更新每个状态-动作对的 Q 值。这对于大型问题来说是不可行的。而 DQN 使用深度神经网络来逼近函数 Q ，DQN 中有两个网络，分别是目标网络（网络参

数 θ) 和主网络 (网络参数 θ')。DQN 通过主网络计算每个动作的 $Q(S_t, a_t; \theta)$ ，而动作 a_t 可以使用 ϵ -贪婪策略 (保证一定的探索) 得到即：

$$a_t = \begin{cases} \arg \max_a Q(S_t, a; \theta), & 1 - \epsilon \\ \text{random}, & \epsilon \end{cases} \quad (2-1)$$

执行动作 a_t ，将得到奖励 r ，而状态转为 S_{t+1} 。通过这样一个交互，就可得到一个经验 (S_t, a_t, S_{t+1}, r) 。在训练网络过程中，DQN 采用经验回放，随机选择一批经验进行训练，以稳定训练过程。训练可以看成是一个监督学习，其目的就是最小化主网络估计的 Q 值与目标网络计算出的 Q 值的损失值，从而拟合函数 Q 。 S_{t+1} 输入到目标网络中，会得到每个动作的 $Q(S_{t+1}, a_{t+1}; \theta')$ ，选择 $\max_{a_{t+1}} Q(S_{t+1}, a_{t+1}; \theta')$ ，从而将式 (2-2) 作为 DQN 逼近的真实值，而 $Q(S_t, a_t; \theta)$ 作为网络的预测值。

$$\hat{y} = r + \gamma \cdot \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}; \theta') \quad (2-2)$$

其中， γ 为折扣因子。

得到这个实际值，就可以计算损失值 (式 (2-3))，并通常使用梯度下降来最小化损失函数，经过迭代更新神经网络的参数，DQN 逐渐逼近最优的 Q 函数，从而学习到最优策略。

$$Loss = (\hat{y} - \max_{a_t} Q(S_t, a_t; \theta))^2 \quad (2-3)$$

(2) DDQN (Double DQN)：尽管 DQN 在许多问题上表现出色，但它仍存在问题。其中一个问题是过高估计。在传统的 DQN 中，使用单个神经网络同时估计当前状态下所有动作的 Q 值。这可能导致对某些动作的过高估计，从而导致学习到的策略偏向于价值较高的动作。因此 DDQN 由 DeepMind 在 2015 年提出^[44]，旨在解决 DQN 中存在的过高估计问题。DDQN 基于 DQN 的基本原理，引入了一种新的思想，即使用两个相互独立的神经网络来分别估计当前状态下的最优动作和其对应的 Q 值。其中目标网络用于选择最优动作，主网络用于计算该动作对应的 Q 值。DDQN 的训练过程如图 2-3 所示。从经验池取出 (S_t, a_t, S_{t+1}, r) ，可以得到 $Q(S_t, a_t; \theta)$ ，将 S_{t+1} 输入到主网络中，会选择最大的 Q 值对应的动作 a_{t+1} ，同时将 S_{t+1} 输入到目标网络，得到 a_{t+1} 对应的 $Q(S_{t+1}, a_{t+1}; \theta')$ ，从而将式 (2-4) 作为 DDQN 的实际值，而 $Q(S_t, a_t; \theta)$ 作为网络的预测值，进行误差反向传播。其中主网络的参数 θ 更新如式 (2-5)，在经过数次更新后，将主网络的参数更新到目标网络中。

$$\hat{y} = r + \gamma \cdot Q(S_{t+1}, a_{t+1}; \theta') \quad (2-4)$$

$$\theta \leftarrow \theta - [\hat{y} - Q(S_t, a_t; \theta)] \nabla_{\theta} Q(S_t, a_t; \theta) \quad (2-5)$$

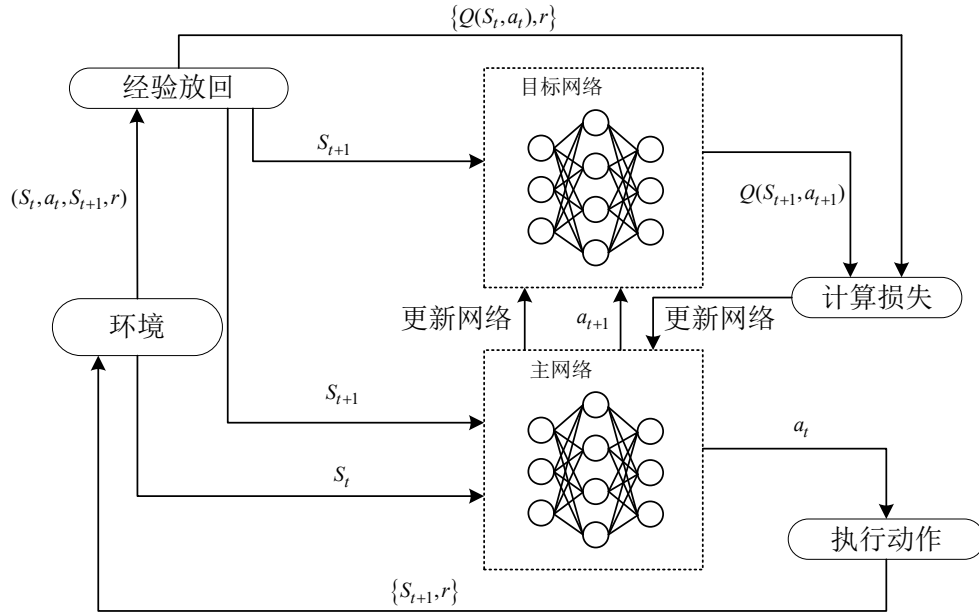


图 2-3 DDQN 训练过程

2.5 本章小结

本章深入探讨了 SDN、NFV 以及 SFC 这三个在现代网络技术中不可或缺的构成部分。并且介绍了深度强化学习相关方法，了解到 DDQN 在解决复杂环境下的有效性。结合 SDN 的网络流量控制、NFV 的功能虚拟化以及 SFC 的服务定制能力，现代网络架构能够更加灵活地响应变化多端的业务需求，同时降低运营成本并加速服务的创新。

3 基于 VNF 共享的服务功能链部署算法

3.1 引言

在第一章研究现状的讨论中，目前对于 SFC 的部署问题主要聚焦于单一优化目标的策略，尤其是在追求最小化 SFC 的端到端时延的过程中，往往忽略了 VNF 实例的部署成本对运营成本的重要影响。这种策略虽然能够在一定程度上满足快速响应的需求，但却导致了资源的低效分配与利用，尤其是当 VNF 实例被优先部署在离源点和终点最近的位置时，不仅增加了运营商的成本负担，同时也浪费了服务器资源的潜在价值。进一步地，如果考虑到服务器的计算能力充足，允许相同类型的 VNF 实例在不同服务链之间进行共享^[45]，不但可以显著提升资源的利用效率，降低 VNF 实例的整体部署成本，还能通过资源共享机制带来经济效益的提升。然而，这种共享策略也可能带来新的挑战，如转发路径可能因此变长，增加了数据传输的开销，进而影响服务的时延性能，有时甚至可能无法达到 SFC 的严格时延要求。随着网络环境的不断变化和服务需求的日益多样化，寻找一种能够在保证 SFC 服务质量（如时延要求）的同时，有效平衡 VNF 部署成本与链路转发成本的策略，显得尤为关键。这不仅要求对现有的部署模型和优化算法进行深入的研究和改进，还需要开发新的策略以适应网络状态变化带来的挑战，确保能够在动态变化的网络环境中实现资源的高效利用和服务的高质量保障，从而达到网络服务管理的最优化。

3.2 问题描述与建模

在本节中，首先将探讨研究所面临的关键问题，并对其进行详细的问题描述。随后，本文将详细阐述所采用的系统模型框架，这包括但不限于对模型中使用的核心符号、定义等。

3.2.1 问题描述

如图 3-1 (a) 所示，有两条 SFC (SFC_1 和 SFC_2)，在不考虑 VNF 实例共享下， SFC_1 有两条转发路径，分别为 $A \rightarrow B \rightarrow E \rightarrow F \rightarrow C \rightarrow D$ 和 $A \rightarrow B \rightarrow C \rightarrow D$ ， SFC_2 经过节点 $I \rightarrow J \rightarrow G \rightarrow H$ 。可以发现两条 SFC 中都存在 VNF_2 ，所以可以复用 VNF_2 实例，将节点 J 上的 VNF_2 部署到节点 F 上，如图 3-1 (b) 所示，则 SFC_2 经过节点 $I \rightarrow F \rightarrow G \rightarrow H$ ，此时部署成本降低。但这样会存在，某些 SFC 在部署 VNF 实

例时，为了到达共享的 VNF 实例，会映射更长的物理路径，增加了带宽资源的使用，如图 3-2 (a) 所示，在不考虑 VNF 实例共享下， SFC_1 经过节点 $A \rightarrow B \rightarrow E \rightarrow F \rightarrow C \rightarrow D$ 和 $A \rightarrow B \rightarrow C \rightarrow D$ ， SFC_2 经过节点 $I \rightarrow J \rightarrow K \rightarrow L$ 。但在图 3-2 (b) 中，若将 VNF_2 共享后， SFC_2 经过节点 $I \rightarrow F \rightarrow G \rightarrow H \rightarrow K \rightarrow L$ ，增加了链路带宽资源的消耗，同时还违反了 SFC 的时延要求，如何平衡链路和 VNF 实例之间的关系，是目前需要解决的问题。

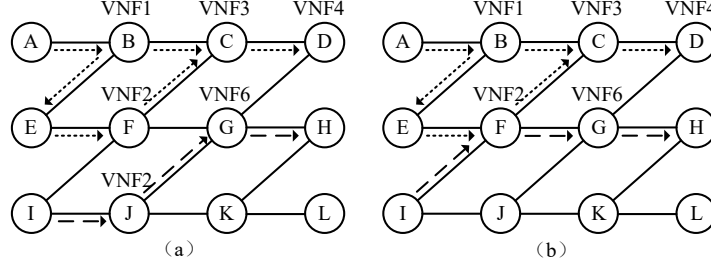


图 3-1 无 VNF 实例共享&有 VNF 实例共享

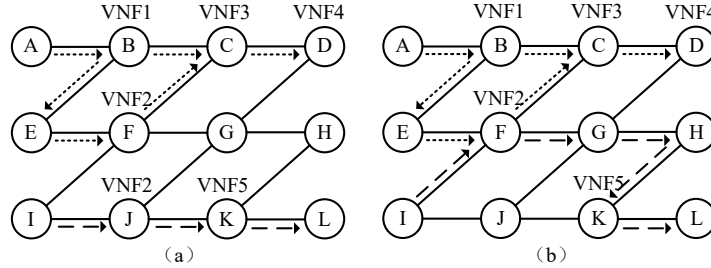


图 3-2 VNF 实例共享引发链路增长

3.2.2 系统模型

(1) 底层物理网络模型

底层物理网络为无向连通图，用 $G=(N,L)$ 图表示， N 表示物理节点的集合，可以进行 VNF 实例化， L 表示物理链路集合，节点 $u \in N$ 的 CPU 计算资源为 C_u ，限制了共享 VNF 实例的 SFC 数量。 $B_{u,v}$ 表示连接节点 u, v 的链路 $l_{uv} \in L$ 的可用带宽资源。 τ_{uv} 表示链路 $l_{uv} \in L$ 的传输延迟。

(2) 服务功能链

在模型中，所有种类的 VNF 集合由 F 表示，SFC 集合由 I 表示。其中 SFC 请求 $i (i \in I)$ 由一个五元组表示 $(s_i, t_i, G_i, D_i, F_i)$ ， s_i 表示源端， t_i 表示目的端， G_i 表示 SFC 的 VNF-FG， D_i 表示 SFC 允许的最大延迟， F_i 表示构成 SFC 的 VNF 集合。相同类型的 VNF 在不同 SFC 中所需资源 $r_j^i (j \in F)$ 不同。

对于 VNF-FG，与文献[46]相同，本文考虑三种基本拓扑，如图 1-2 所示：a) 单链；b) 具有不同目的端的拆分；c) 具有单个目的端的拆分合并。利用经典的图搜索算法，如深度优先搜索可以很轻松地构建 VNF-FG 的 VNF-FP 集合。

SFC 请求 i 的 VNF-FP 的集合用 $\mathcal{F}_i$ ($i \in I$) 表示, 而 i 的第 j 条转发路径为 $f_{i,j}$ ($f_{i,j} \in \mathcal{F}_i$), 如图 3-1 (a) 中 SFC_I 所需的转发路径为 $\mathcal{F}_I = (f_{I,1}, f_{I,2})$, 其中 $f_{I,1} = (l_{AB}, l_{BC}, l_{CD})$, $f_{I,2} = (l_{AB}, l_{BE}, l_{EF}, l_{FC}, l_{CD})$ 。其中第 z 条虚拟链路, 为 $f_{i,j,z}$ ($f_{i,j,z} \in f_{i,j}$), 它所对应的真实物理转发路径存储在 $P_{f_{i,j,z}}$ 中, $b_{f_{i,j,z}}$ 表示虚拟链路 $f_{i,j,z}$ 所需的带宽资源。第 j 条转发路径对应的真实物理转发路径集合为 $P_{f_{i,j}}$ ($P_{f_{i,j,z}} \in P_{f_{i,j}}$)。 $F_{f_{i,j}}$ 为第 j 条转发路径上所需的 VNF 集合。

(3) 运营成本

当 i 中第 j 个 VNF 实例化在物理节点 u 时, 记 $z_{iju} = 1$ 。当物理节点 u 已有 VNF 实例, 则为 $x_u = 1$ 。 V_{cost} 表示启用一个物理节点所用的成本。 V_l 表示在物理链路上使用单位带宽的成本。

则在物理节点 u 实例化 VNF 的成本为 V_l , 其中若该节点已经实例化 VNF 则无需再花费部署成本:

$$V_I = V_{cost} \cdot \sum_{i \in I} \sum_{j \in \mathcal{F}_i} \sum_{u \in N} z_{iju} \cdot (1 - x_u) \quad (3-1)$$

则 SFC 的链路转发成本 V_2 :

$$V_2 = V_l \cdot \sum_{i \in I} \sum_{f_{i,j} \in \mathcal{F}_i} \sum_{f_{i,j,z} \in f_{i,j}} |P_{f_{i,j,z}}| \cdot b_{f_{i,j,z}} \quad (3-2)$$

因此, 总运营成本为:

$$V_{total} = \alpha_1 V_1 + \alpha_2 V_2 \quad (3-3)$$

其中, α_1 , α_2 是对成本组成进行优先排序的权重因子。

(4) 时延

对于每个 SFC 来说, 每条 VNF-FP 映射到物理链路上, 在物理链路上的传输时延为:

$$d_{f_{i,j}} = \sum_{f_{i,j,z} \in f_{i,j}} \sum_{(u,v) \in P_{f_{i,j,z}}} \tau_{uv} \quad (3-4)$$

则每个 SFC 所需的传输时延取转发路径中所需最长的时延作为该条 SFC 的传输时延:

$$d_i = \max_{f_{i,j} \in \mathcal{F}_i} \{d_{f_{i,j}}\} \quad (3-5)$$

本节涉及的主要参数及意义如表 3-1 所示。

表 3-1 参数及意义

符号	说明
N, L	分别代表可用的物理节点、物理链路的集合
C_u	节点 $u \in N$ 的计算资源
B_{uv}, τ_{uv}	分别代表链路 $l_{uv} = (u, v) \in L$ 的可用带宽资源、传输延迟

表 3-1 参数及意义 (续)

符号	说明
I	SFC 请求集合, $ I $ 为请求数量
F_i	i 的 VNF 集合
G_i	i 的 VNF-FG
s_i, t_i, D_i	分别代表 i 的源端, i 的目的端, 以及 i 的允许时延
$\mathcal{F}_i$	i 中转发路径的集合, $ \mathcal{F}_i $ 为转发路径数量
$F_{f_{i,j}}$	i 的第 j 条转发路径上 VNF 的集合, $ F_{f_{i,j}} $ 为转发路径上的 VNF 数量
r_j^i	i 中 VNF_j 所需的计算资源
$f_{i,j}$	i 的第 j 条转发路径
$f_{i,j,z}$	i 的第 j 条转发路径的第 z 条虚拟链路
$P_{f_{i,j,z}}$	$f_{i,j,z}$ 所对应的真实物理转发路径
$P_{f_{i,j}}$	第 j 条转发路径对应的真实物理转发路径集合, $ P_{f_{i,j}} $ 为集合中转发路径数量
$b_{f_{i,j,z}}$	虚拟链路 $f_{i,j,z}$ 所需的带宽资源
V_l	物理链路上使用单位带宽的成本
V_{cost}	部署一个 VNF 所用的成本
z_{iju}	二进制变量, i 中 VNF_j 映射在节点 u 上, 则为 1
x_u	二进制变量, 物理节点 u 已被实例化 VNF, 则为 1

3.2.3 部署优化模型

本文是对服务功能链进行成本优化部署, 保证每个 SFC 的服务质量, 因此本文同时对 VNF 实例化成本和链路转发成本联合优化, 优化目标整理为:

$$\begin{aligned}
 & \min V_{total} \quad (3-6) \\
 st \quad & \begin{cases}
 C1: \sum_{i \in I} \sum_{j \in F_i} z_{iju} \cdot r_j^i \leq C_u, \forall u \in N \\
 C2: \sum_{i \in I} \sum_{f_{i,j} \in \mathcal{F}_i} \sum_{f_{i,j,z} \in f_{i,j}} b_{f_{i,j,z}} \leq B_{u,v}, \forall (u,v) \in P_{f_{i,j,z}} \\
 C3: \sum_{u \in N} z_{iju} \leq 1, \forall i \in I, \forall j \in F_i \\
 C4: \sum_{j \in F_i} \sum_{u \in N} z_{iju} = |F_i|, \forall i \in I \\
 C5: d_i \leq D_i, \forall i \in I \\
 C6: z_{iju} \in \{0,1\}, \forall i \in I, \forall j \in F_i, \forall u \in N \\
 C7: x_u \in \{0,1\}, \forall u \in N
 \end{cases}
 \end{aligned}$$

本文的优化目标受到 C1~C7 约束, 其中 C1 表示由于 CPU 资源, 每个服务器只能支持有限个 VNF 实例。C2 表示分配给 SFC 相应的转发路径带宽不能超过物理链路的带宽。C3 表示 SFC 中的每个 VNF 只能部署在单个物理节点上,

不能由多个物理节点服务。C4 表示对于每条 SFC，一个物理节点上只能部署一种类型的 VNF 实例。C5 表示 SFC 的实际传输时延不可超过最大允许时延。C6 为二进制变量约束，当 i 中的第 j 个 VNF 实例化在物理节点 u 上时， z_{ju} 为 1，否则为 0。C7 为二进制变量约束，当物理节点 u 已被实例化，则 x_u 为 1，否则为 0。

3.3 问题求解

3.3.1 算法描述

假设时间被平均划分为多个时隙，这些时隙编号为 1, 2, ..., T ，其中 T 为监测周期的长度。每当一个 SFC 请求到达网络系统时，决策系统会立即判断该 SFC 在下一个时间段内是否被接纳进网络，或者因资源不足等原因被拒绝。一旦 SFC 被接受，将在一定时间内占用或共享部分资源。当服务完成后会将资源释放供其他 SFC 需要。因此在时隙 $t-1$ 结束时网络中是存在部分 VNF 实例的， t 时刻到达的 SFC，在部署时可以重用已部署的相同类型的 VNF 实例，也可以在新的服务器上部署。

为了优化资源的利用率并降低整体的网络运营成本，本文提出了一种综合考量路径选择和资源共享的策略。沿着更短的路径部署 VNF 意味着更少的链路转发成本，但也代表着减少了共享 VNF 实例，因此需要在更多的物理节点部署 VNF 实例。相反，共享 VNF 实例意味着产生更长的转发路径，并且可能违反 SFC 时延约束，因此这两者间是存在矛盾的。本文将这个问题分为两步解决：a) 为每个服务 SFC 转发路径设置路径长度上限，用 $Len_{i,j}$ 表示；b) 在路径长度约束下最大化重用 VNF 实例数量。

对于 SFC 中每条转发路径，为了确保可以连接， $Len_{i,j}$ 至少要大于两台服务器之间的最短路径。因为 SFC 中无法共享的 VNF 实例必须部署在新的节点上，因此沿最短路径可能没有足够的空服务器来放置，所以需要扩展最短路径到达更多空服务器，并考虑所需的额外链路资源，与此同时，尽可能的满足 SFC 的时延需求。

所以 $Len_{i,j}$ 可以用下式表示：

$$Len_{i,j} = |SP_{f_{i,j}}| + \max\left(w_{P_{f_{i,j}}} \cdot \left(|F_{f_{i,j}}| - n_{P_{f_{i,j}}}\right), 0\right) \quad (3-7)$$

其中 $|SP_{f_{i,j}}|$ 表示 i 中第 j 条转发路径从源端到目的端上的最短路径长度， $w_{P_{f_{i,j}}}$ 是当前最短路径上服务器到达相邻空服务器和可共享 VNF 所在服务器的平均距

离, $|F_{f_{i,j}}|$ 代表 i 中第 j 条转发路径上所需的 VNF 数量, $n_{p_{f_{i,j}}}$ 表示当前最短路径上的空服务器数量。

对于长度上限, 可以根据式 (3-7) 得出, 而如何最大化重用因子是本文主要解决的: 对于请求 i , 其第 j 条转发路径为 $f_{i,j}$, 需要找到一条连接源端 S_i 和目的端 T_i 的路径 $P_{f_{i,j}}$, 为 $f_{i,j}$ 上的每个 VNF_k ($VNF_k \in F_{f_{i,j}}$) 进行部署, $F_{f_{i,j}}$ 为 $f_{i,j}$ 上 VNF 集合, 即对 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 找到一个部署最优解 $f(P_{f_{i,j}}, F_{f_{i,j}})$, 最大化重用因子 (即在该条物理路径上可共享的 VNF 数量)。对于 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 可以分为两个子问题, 即 $Q(P_{f_{i,j,z'}}, F'_{f_{i,j}})$ 和 $Q(P_{f_{i,j,z''}}, F''_{f_{i,j}})$, 其中 $P_{f_{i,j,z'}}$ 为未加入最后一条实际链路的集合, $P_{f_{i,j,z''}}$ 为最后一条实际链路, 而 $F'_{f_{i,j}} = (VNF_1, \dots, VNF_h)$, $F''_{f_{i,j}} = (VNF_{h+1}, \dots, VNF_{|F_{f_{i,j}}|})$ 。因此本文优化方法使用动态规划思想, 对于 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 问题, 拆分成两个子问题 $Q(P_{f_{i,j,z'}}, F'_{f_{i,j}})$ 和 $Q(P_{f_{i,j,z''}}, F''_{f_{i,j}})$ 进行递归求解, 其问题的解组合后即为 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 的解。可以表达为 $f(P_{f_{i,j,z}}, F_{f_{i,j}}) = f(P_{f_{i,j,z'}}, F'_{f_{i,j}}) + f(P_{f_{i,j,z''}}, F''_{f_{i,j}})$, 若子问题没有解, 则 $f(P_{f_{i,j,z}}, F_{f_{i,j}}) = \emptyset$ 。由此可以得到动态方程:

$$f(P_{f_{i,j,z}}, F_{f_{i,j}}) = \arg \max_{f_h: 0 \leq h \leq |F_{f_{i,j}}|} x(f_h) \quad (3-8)$$

$$\text{其中 } f_h = \begin{cases} f(P_{f_{i,j,z'}}, F'_{f_{i,j}}), & h = |F_{f_{i,j}}| \\ f(P_{f_{i,j,z''}}, F''_{f_{i,j}}), & h = 0 \\ f(P_{f_{i,j,z'}}, F'_{f_{i,j}}) + f(P_{f_{i,j,z''}}, F''_{f_{i,j}}), & 0 < h < |F_{f_{i,j}}| \end{cases}$$

其中 $x(f_h)$ 为 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 求解后得到的重用因子。而对于单个转发路径上重用因子的求解, 利用动态规划递归求解, 很容易找到子问题的解, 从而可以在最大化重用因子下得到部署策略。

如算法 1 所示, 使用动态规划 (第 3 至 17 行) 将当前虚拟转发路径与所找到的一条物理路径进行比较, 得到当前物理路径可支持顺序部署的 VNF 数量和重用因子。如当前虚拟转发路径上 VNF 顺序集合 $F_{f_{i,1}} = \{VNF_1, VNF_2, VNF_3, VNF_4\}$, 物理路径上服务器部署情况 $PF = \{VNF_1, VNF_2, \text{empty}, \text{empty}, VNF_4\}$, 使用算法 1 可得到当前物理路径上可支持部署的 VNF 数量 Len 为 4, 重用因子 x 为 3。

算法 1 计算物理路径的可支持长度和重用因子算法

输入: 虚拟转发路径上 VNF 顺序集合 $F_{f_{i,j}}$, 物理路径上服务器部署集合 PF

输出：物理路径可支持部署的 VNF 数量 Len ，重用因子 x

1. 初始化二维数组 t , w 动态记录可支持部署的 VNF 数量和重用因子，初始化一维数组 dep 记录部署方案
2. // 将虚拟转发路径上 VNF 与物理路径上部署情况进行比较
3. for $k=1$ to $|F_{f_{i,j}}|$, do
4. for $z=1$ to $|PF|$, do
5. if 虚拟转发路径上 VNF 与当前服务器上 VNF 相同时, then
6. $t[k][z] = t[k-1][z-1] + 1$, $w[k][z] = w[k-1][z-1] + 1$
7. elif 当前服务器未实例化 VNF, then
8. $t[k][z] = t[k-1][z-1] + 1$, $w[k][z] = \max(w[k-1][z], w[k][z-1])$
9. else
10. $t[k][z] = \max(t[k-1][z], t[k][z-1])$, $w[k][z] = \max(w[k-1][z], w[k][z-1])$
11. end if
12. end for
13. if $t[k][|PF|] < k$, then //无法按照顺序放置 VNF
14. $k = k - 1$, break //停止放置
15. end if
16. end for
17. $Len = t[k][|PF|]$, $x = w[k][|PF|]$ /* 物理路径可支持部署 VNF 数量和重用因子 */
18. return Len, x

算法 2 根据算法 1 中记录可支持顺序部署的 VNF 数量和重用因子的二维数组，回溯得到该条虚拟路径在最优物理路径上的部署方案。如对于 $F_{f_{i,1}} = \{VNF_1, VNF_2, VNF_3, VNF_4\}$, $PF = \{VNF_1, VNF_2, empty, empty, VNF_4\}$ 使用算法 2 可以得到部署方案 $dep = \{1, 2, 4, 5\}$ ，即相对应的物理路径的位置下标。

算法 2 单个转发路径的 VNF 部署算法

输入：虚拟转发路径上 VNF 顺序集合 $F_{f_{i,j}}$ ，物理路径上服务器部署情况集合 PF ，算法 1 中动态记录可支持部署的 VNF 数量和重用因子的二维数组 t, w

输出：部署方案 dep

1. 初始化一维数组 dep 记录部署方案
2. // 回溯得到在当前物理路径上的部署方案
3. $Index = |PF|$ // index 记录 PF 的下标
4. for $m = |F_{f_{i,j}}|$ to 1, do // 从后往前遍历虚拟转发路径上 VNF
5. $flag = 0$
6. for $n = Index$ to 1, do
7. if 虚拟转发路径上 VNF 与当前服务器上 VNF 相同时且 $F_{f_{i,j}}$ 中第 m 个

```

        VNF 放置在物理路径第  $n$  个位置时可支持的 VNF 数量  $Len$  为  $m$ , then
    8.          $index = n$ , break
    9.         if 当前服务器未实例化且  $F_{f_i,j}$  中第  $m$  个 VNF 放置在物理路径第  $n$  个
               位置时可支持的 VNF 数量  $Len$  为  $m$ , then
    10.             $index = n, flag = 1$ , break
    11.        end if
    12.    end for
    13.    if  $flag == 1$  then
    14.        for  $n = index$  to 0, do
    15.            if 虚拟转发路径上 VNF 与服务器上 VNF 相同时且  $F_{f_i,j}$  中第  $m$  个 VNF
               放置在物理路径第  $n$  个位置时  $Len$  为  $m$ , then
    16.                 $index = n$ , break
    17.            end if
    18.        end for
    19.         $dep[m] = index, index = index - 1$  // 得到部署方案
    20.    end if
    21. end for
    22. return  $dep$ 
    
```

综上所述, 本文提出基于 VNF 共享的服务功能链部署算法, 如算法 3 所示。依次遍历每条 SFC 的转发路径, 计算路径长度上限 (第 5 行), 根据 SFC 的网络资源需求, 在物理网络中查找源端到目的端的路径, 筛选出符合路径长度上限和时延需求的可行路径 (第 6 行), 遍历可行路径, 调用算法 1, 计算可支持部署的 VNF 数量和重用因子, 比较各路径的重用因子与物理路径长度的比值, 取比值最高的物理路径, 调用算法 2 进行部署 (第 10 至 18 行)。若 SFC 无法被提供服务, 则 SFC 被拒绝 (第 20 至 22 行)。更新网络资源后 (第 23 行), 继续下一条 SFC 的部署。

算法 3 基于 VNF 共享的服务功能链部署算法

输入: 在 t 时刻到达的 SFC

输出: 部署策略

1. 初始化一维数组 $bestpath$ 记录部署方案, 初始化变量 $reject$ 为 0, 记录拒绝的服务数量
 2. for $i=1$ to $|I|$, do
 3. for $j=1$ to $|\mathcal{F}_i|$, do
 4. $g = 0$ // 记录重用因子与物理路径长度的比值
 5. 根据网络中资源, 使用式 (3-7) 计算路径长度上限 $Len_{i,j}$
 6. 根据资源需求和时延需要, 找出从源端到目的端符合 $Len_{i,j}$ 和时延需求
-

```

        的可选路径,存放在  $P_{f_{i,j}}$  中
    7.     if  $P_{f_{i,j}}$  为空, then
    8.         break
    9.     end if
    10.    for each  $P_{f_{i,j}}$ , do
    11.        调用算法 1,得到可支持部署的 VNF 数量  $Len$ ,重用因子  $x$ 
    12.        if  $Len$  等于虚拟路径上所需的 VNF 数量,then
    13.            if  $x/|P_{f_{i,j}}| > g$ , then
    14.                 $g = x/|P_{f_{i,j}}|$ 
    15.            end if
    16.        end if
    17.    end for
    18.    取比值最高的物理路径,调用算法 2 返回放置方案存入  $bestpath$ 
    19.    end for
    20.    if 无部署方案, then
    21.        拒绝请求
    22.    end if
    23.    更新网络资源
    24.    end for
    25.    return 部署策略
    
```

3.3.2 算法复杂度分析

在 SFC 部署阶段,共有 $|I|$ 条 SFC 到达,其中每条 SFC 又有 $|\mathcal{F}_i|$ 条转发路径。对于每条转发路径,本文与 $|P_{f_{i,j}}|$ 条可选路径进行比较,选择最佳路径,即算法 1 所示,而算法 1 需要进行 $|F_{f_{i,j}}| \times |PF|$ 循环,所以算法 1 的时间复杂度为 $O(|F_{f_{i,j}}| \times |PF|)$,其中 $|F_{f_{i,j}}|$ 为转发路径上 VNF 数量, $|PF|$ 为物理节点数量。当选出最佳部署路径时,进行算法 2 部署,它的时间复杂度也为 $O(|F_{f_{i,j}}| \times |PF|)$,因此本文所提出的算法时间复杂度为 $O(|I| \times |\mathcal{F}| \times |P_{f_{i,j}}| \times |F_{f_{i,j}}| \times |PF|)$ 。

3.4 仿真设计及结果分析

本节对提出的基于 VNF 共享的服务功能链部署算法进行仿真和性能分析,

为了确保评估的准确性和公正性,本研究还引入了两种现有的启发式算法作为比较基准,并从运营成本、VNF 利用率和服务接受率进行分析,旨在全面评估算法的性能表现。

3.4.1 实验方法及仿真参数设置

本文采用含有 58 个节点,87 条链路的网络拓扑,这个网络拓扑来源于 Knight 等人^[47]的一项致力于收集各国不同网络拓扑的研究。每个的 CPU 资源为 100%,而物理链路的带宽为 10 Gbps^[48]。物理链路的延迟在 2ms 和 4ms 之间^[48]。

在单个时隙的情况下,本文假设部分 VNF 实例已经随机放置在网络中。每条 SFC 是通过选择 N 中的节点作为源 s_i 和目标 t_i 随机生成的,并根据文献[49]分配其所需带宽资源和延迟要求,其中根据文献[48]设置每个网络功能所需 CPU 计算资源,如表 3-2 所示。每个 SFC 的服务链的生成与文献[47]中的一致。

表 3-2 关键参数和值

参数	值
VNF 数量	1-5
$ \mathcal{F}_i $	1-2
D_i	40ms~100ms
$b_{f_i,j,z}$	10 Mbps~120Mbps
r_j^i	10%~25%

3.4.2 对比算法介绍

由于 SFC 部署问题的复杂性以及评估指标的多样性,目前尚缺乏统一的标准数据集和评估框架,这使得不同研究之间的比较存在一定的难度^[50]。但为了确保比较的公正性和有效性,本文基于同一仿真环境和相同的评估标准,选取了文献[17]中两种被认可的启发式算法作为比较对象:

1. **Shortest Path Heuristic (SPH)**: 该方法的核心思想是沿着源端和目的端之间的最短路径进行 SFC 的部署,即对于 SFC 中的每个 VNF-FP,均选择网络中的最短路径进行部署,能够在一定程度上减少链路转发成本。
2. **Greedy on Used Server (GUS)**: 侧重于 VNF 实例的重用,优先在网络中已有的、可共享的 VNF 实例上放置新的 SFC,以避免占用新的物理资源,一定程度上降低总体的网络运营成本。

3.4.3 评价指标

运营成本：作为本文的优化目标，运营成本即为 VNF 部署成本和链路转发成本之和。可以由式（3-3）所得。

利用率：即物理网络中 VNF 实例利用率。

接受率：即在 t 时隙成功部署 SFC 数量，衡量网络服务能力的一个重要指标，直接反映了网络对新服务请求的处理能力。可以由式（3-9）所得。

$$AC = \frac{AC_{suc}}{|I|} \quad (3-9)$$

其中， $|I|$ 为到达的 SFC 数量， AC_{suc} 为 t 时隙成功部署 SFC 数量。

3.4.4 实验结果与分析

运营成本：作为本文的优化目标，运营成本即为 VNF 部署成本和链路转发成本之和。在仿真中，本文将 α_1 ， α_2 均设为 1，这意味着链路和 VNF 实例部署一样重要。为了清晰的观察各个算法的结果，本文对数据进行了归一化处理，结果在[0,1]之间。如图 3-3 所示，相对于另外两种算法，本文提出的算法的总运营成本最低，其中 SPH 和 GUS 所需要的平均总成本分别高于本文算法 6.6%和 12.15%。其中链路转发成本如图 3-4 所示，GUS 的平均链路转发成本高于本文算法 50.38%，该算法为了尽可能的共享 VNF 实例，不可避免的增加了路径长度。对于 VNF 实例部署成本，如图 3-5 所示，SPH 的部署成本最高，高于本文算法 33.94%，因为每个 SFC 都贪婪的部署在最短路径上，而忽略共享 VNF 实例。通过比较，本文基于 VNF 共享的服务功能链部署算法很好的平衡了链路转发成本和部署成本，能够有效降低运营成本。

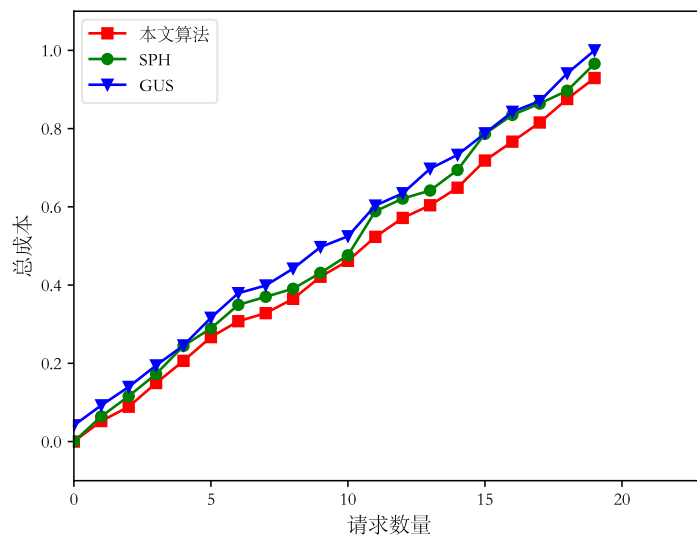


图 3-3 运营成本

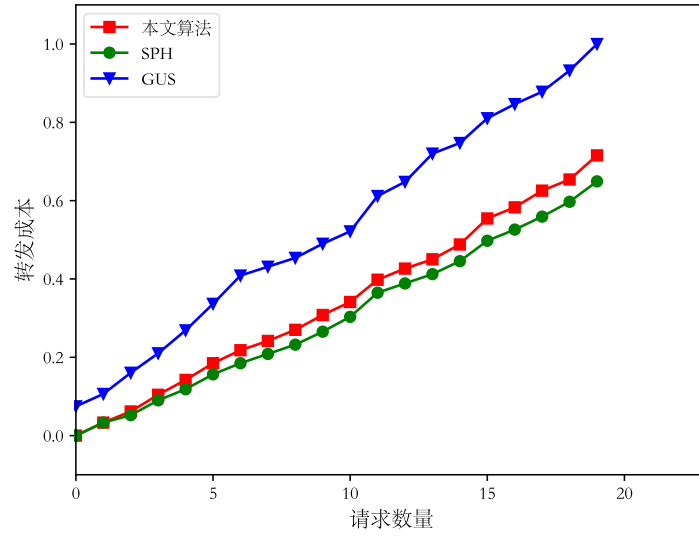


图 3-4 链路转发成本

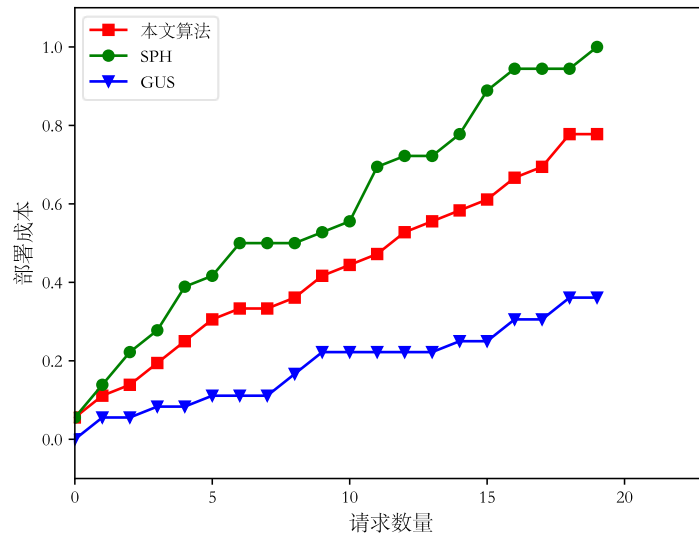


图 3-5 VNF 部署成本

利用率：如图 3-6 所示，由于 GUS 优先考虑 VNF 实例共享，因此其 VNF 实例利用率最高，高于本文算法 42.4%，而 SPH 优先考虑最短路径部署，考虑较少的 VNF 实例共享，因此它的利用率最低，低于本文算法 23.82%。虽然本文的利用率低于 GUS，但是牺牲了部分 VNF 实例部署成本，从而避免了链路转发成本的过度增长。

接受率：如图 3-7 所示，当 SFC 数量增大时，本文基于 VNF 共享的服务功能链部署算法服务接受率均高于 SPH 和 GUS，如当 SFC 数量达到 75 时，SPH 和 GUS 的服务接受率分别为 84% 和 86.67%，而本文算法的服务接受率为 89.33%。因为 SPH 优先考虑最短路径部署，导致网络中存在 VNF 实例独占服务器资源，而 GUS 虽然共享了 VNF 实例，但是因为路径不可避免的增长，使链路带宽资源无法满足后续 SFC，因此本文所提出的算法在网络中可以支持更多的 SFC。

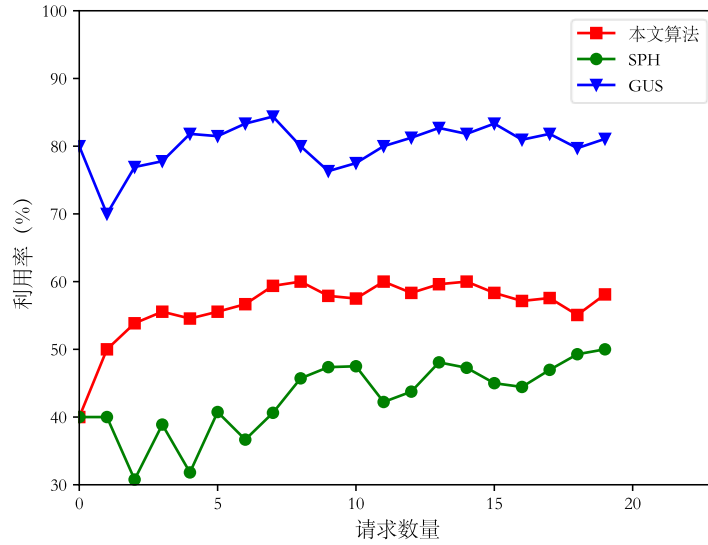


图 3-6 VNF 利用率

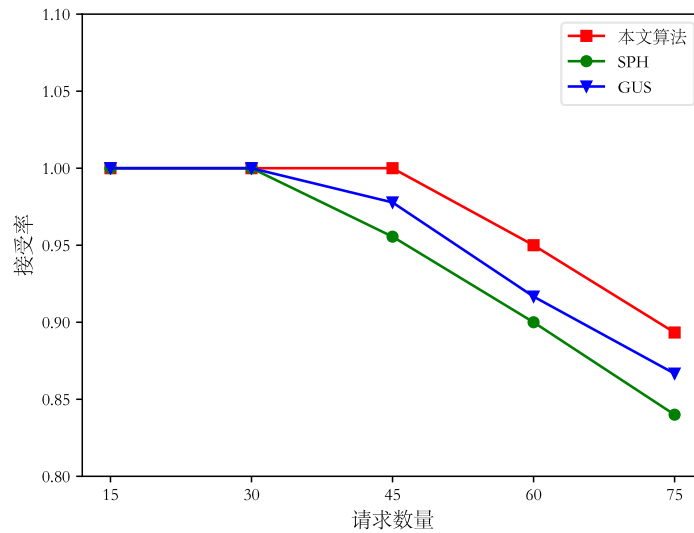


图 3-7 服务接受率

3.5 本章小结

针对 SDN/NFV 架构下 SFC 部署成本优化问题，本文首先将问题建模为多目标和多约束问题模型，该模型以最小化 VNF 实例部署成本和链路转发成本为目标，同时受限于每条 SFC 时延、物理链路带宽资源和服务器 CPU 资源约束，其次为每条 VNF-FP 设置路径长度上限，以平衡链路和 VNF 实例的使用，最后提出基于 VNF 共享的服务功能链部署算法，联合 SFC 放置和路径选择问题，以节省链路转发成本并有效地重用 VNF 实例。仿真结果表明，与 SPH 和 GUS 算法相比，本文的算法可以更好地管理链路和 VNF 实例之间的关系，显著的降低了运营成本。

4 基于深度强化学习的服务功能链动态调整算法

4.1 引言

在 SFC 部署完成后, 请求的资源需求仍在不断变化, 这种变化可能影响物理网络的资源消耗, 并且违反服务等级协议 (Service Level Agreement, SLA)。因此, NFV 编排器需要动态地管理 VNF 资源以满足变化的需求。当 SFC 的资源需求增加, 就会造成部分物理服务器的 CPU 资源负载增加, 从而影响服务功能链时延, 降低 QoS, 而当 SFC 的资源需求降低, 则会导致服务器的资源利用率下降, 进而增加网络能耗。因为如果服务器没有有效进入低功耗状态, 网络能耗仍然会保持较高的状态。此外, 即使在低功耗下运行, 服务器仍然消耗其最大功率的 70%^[51]。如果服务器上的 VNF 在较长时间内对资源的需求较低, 迁移该 VNF 并随后使服务器进入休眠状态, 可以提高整体能效。因此, 本文可以考虑三种类型的缩放策略对服务器上部分实例动态调整: (a) 扩展/缩小当前 VNF 实例资源 (b) 实例化新的 VNF 实例 (c) 迁移当前 VNF 实例, 从而实现网络资源分布的均衡, 保证用户 QoS。然而, 这些策略会产生相应的开销。当资源需求增大, 若服务器还有足够的剩余资源且资源分布较为均衡, 此时扩展当前 VNF 实例资源是合适的, 调整开销较小。若当前服务器资源无法满足或资源分布不均衡时, 可以考虑进行 VNF 实例迁移或者实例化新的实例, 但是创建新的 VNF 实例, 虽然可以很好的解决资源变化, 但可能会产生更高的调整开销。同样 VNF 实例迁移, 不仅会有迁移开销, 还会增加带宽消耗, 还可能影响请求时延, 且当前 VNF 实例可能是由多个服务链共享, 所以迁移需要考虑对多个服务功能链的影响。因此在本文, 我们尝试提出一个有效的动态调整策略, 满足用户需求的同时, 最小化相关的调整开销, 实现网络资源的均衡。

4.2 问题描述与建模

4.2.1 问题描述

如图 4-1 (a) 所示, 有三条请求 $SFC_1\{VNF_1, VNF_2, VNF_3\}$ 、 $SFC_2\{VNF_1, VNF_3, VNF_4\}$ 、 $SFC_3\{VNF_2, VNF_5\}$ 已部署在标准通用服务器上, 每个服务器有 10 个单位的 CPU 资源, SFC_1 中的三个网络功能分别部署在 Ser_1 、 Ser_2 、 Ser_3 上, SFC_2 中网络功能分别部署在 Ser_1 、 Ser_5 、 Ser_4 上, 其中 VNF_1 与 SFC_1 中的 VNF_1 共享 Ser_1 上的实例。 SFC_3 中网络功能分别共享 Ser_2 、 Ser_5 上实例, 图中括号中数据为

各实例所占用的资源。随着时间的变化, SFC_2 中 VNF_1 的需求增长, 而服务器有足够的网络资源, 可以满足需求增长, 如图 4-1 (b) 所示, 因此 Ser_1 中实例进行垂直缩放, 满足需求变化。但若 VNF_1 的需求长期如此, 该节点可能过载而导致性能下降, 在这种情况下如图 4-1 (c) 所示, 在 Ser_6 上创建新的 VNF_1 实例, 并且需要对 SFC_2 进行重新路由, 此时不仅需要考虑服务器上资源能否满足, 也需要考虑重新路由的 SFC_2 请求时延能否满足。虽然这种调整会产生开销, 但是网络资源分布更加均衡, 这在后续需求变化时产生正向反馈, 从而满足更多的请求。当 SFC_1 中 VNF_3 的需求降低时, 如图 4-1 (d) 所示, Ser_5 有足够的资源满足 VNF_3 的需求, 且迁移后请求时延仍然满足, 因此为了降低能耗, 可以将 Ser_3 中的 VNF_3 实例迁移到 Ser_5 上, 对实例进行合并, 充分利用 Ser_5 上资源。从该示例中, 可以观察到, 运营商可以根据 SFC 需求变化选择三种不同的解决方案 (垂直和水平缩放、迁移)。本文的目标是通过确定合适的动态调整策略, 保证资源需求和请求时延的同时, 降低运营商的运营成本, 使网络资源分布更均衡。

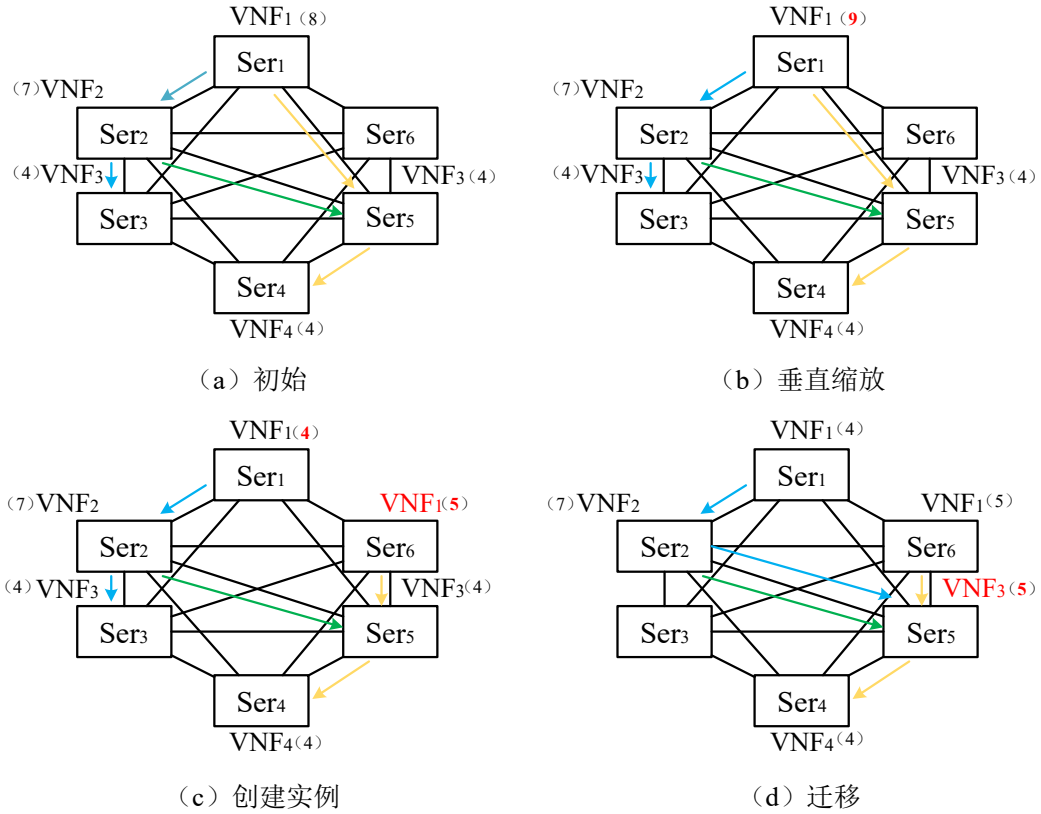


图 4-1 动态调整示例

4.2.2 系统模型

(1) 底层物理网络模型

底层物理网络为无向连通图，用图 $G=(N,L)$ 表示， N 表示物理节点的集合，可以进行 VNF 实例化， L 表示物理链路集合，节点 $u \in N$ 的 CPU 计算资源为 C_u ， $B_{u,v}$ 表示链接节点 u, v 的链路 $l_{uv} \in L$ 的可用带宽资源。 τ_{uv} 表示链路 $l_{uv} \in L$ 的传输延迟。每个节点都有一定的 CPU 资源，限制了共享 VNF 实例的请求数量。

(2) 服务功能链

在模型中，所有种类的 VNF 集合由 F 表示，服务请求集合由 I 表示，相同 VNF j 的在不同请求中所需资源 r_j^i ($j \in F$) 不同。每个请求 i ($i \in I$)，由多个有序的 VNF 构成，其构成的 VNF 集合由 F_i 表示。请求 i ($i \in I$) 由一个四元组表示 (s_i, t_i, F_i, D_i) ， s_i 表示源端， t_i 表示目的端， D_i 表示请求允许的最大延迟。请求 i 的第 j 条虚拟链路为 $f_{i,j}$ ($f_{i,j} \in i$)， b_i 表示 i 所需的带宽资源。考虑到 SFC 资源动态调的复杂性，本节侧重于单链 SFC 条件下，有效降低问题的复杂度，使得问题的建模和求解变得更为可行，为处理更复杂的多链 SFC 问题奠定基础。这些基础理论和方法在后续研究中可以被扩展或修改以适应更复杂的场景。

(3) 资源调整成本

在本文中，SFC 的调整成本主要分为两个部分：迁移成本和实例化成本。下面是对这两个成本的定义。

迁移成本：当服务器上 VNF 实例迁移到另一个服务器上时，记所花费的成本为 $C_{mig}(t)$ 。参考文献[17]， $c_{mig}(t) = C_b \cdot b_i \cdot |P_{m_i}(t)|$ 为 t 时刻迁移后未处理的数据包所占用的带宽资源成本。其中， C_b 表示在物理链路上使用单位带宽的成本， b_i 表示处理 i 所需的带宽资源。 $|P_{m_i}(t)|$ 表示迁移后 i 的未处理数据包的路径数量，例如，一条 SFC 其转发路径是 $(Ser_1 \rightarrow Ser_2 \rightarrow Ser_3)$ ，在迁移后其转发路径为 $(Ser_1 \rightarrow Ser_4 \rightarrow Ser_5 \rightarrow Ser_3)$ ，则 $|P_{m_i}(t)| = 3$ ，本文将其定义为式 (4-2)，其中 $\pi_{u,v}^{f_{i,j}}(t)$ 是二进制变量，表示在 t 时刻是否将 i 中第 j 条虚拟链路映射在物理链路 $l_{uv} \in L$ 上。因此，总迁移成本可定义为式 (4-1)。

$$C_{mig}(t) = \sum_{i \in I} c_{mig}(t) = \sum_{i \in I} C_b \cdot b_i \cdot |P_{m_i}(t)| \quad (4-1)$$

$$|P_{m_i}(t)| = \sum_{f_{i,j} \in i} \sum_{(u,v) \in L} (\pi_{u,v}^{f_{i,j}}(t) - \pi_{u,v}^{f_{i,j}}(t-1)) \cdot \pi_{u,v}^{f_{i,j}}(t) \quad (4-2)$$

实例化成本：同文献[33]一样，本文将实例化成本 $C_{delopy}(t)$ 定义为当前 t 时刻创建的新 VNF 实例成本。

$$C_{delopy}(t) = \sum_{i \in I} \sum_{j \in F_i} \sum_{u \in N} C_{ins} \cdot z_{iju}(t) \quad (4-3)$$

其中， C_{ins} 为实例化单个 VNF 成本。 $z_{iju}(t)$ 为二进制变量， t 时刻 i 中的 VNF $_j$ 实例化在节点 u 上时，记为 1。

因此，总成本可定义为：

$$C_{total}(t) = \alpha_1 \cdot C_{mig}(t) + \alpha_2 \cdot C_{delopy}(t) \quad (4-4)$$

其中， α_1 ， α_2 为成本之间的常数系数。

本文涉及的主要参数及意义如表 4-1 所示。

表 4-1 参数及意义

符号	说明
N, L	分别代表可用的物理节点、物理链路的集合
C_u	节点 $u \in N$ 的计算资源
B_{uv}, τ_{uv}	分别代表链路 $l_{uv} = (u, v) \in L$ 的可用带宽资源、传输延迟
I	SFC 请求集合， $ I $ 为请求数量
F_i	i 的 VNF 集合
s_i, t_i, D_i	分别代表 i 的源端， i 的目的端， i 允许的最大时延
r_j^i	i 中 VNF_j 所需的计算资源
$f_{i,j}$	i 的第 j 条转发路径
b_i	i 所需的带宽资源
P_{m_i}	i 迁移后未处理的转发路径
C_{ins}	实例化单个 VNF 成本
z_{iju}	二进制变量， t 时刻 i 中 VNF_j 映射在节点 u 上，则为 1
$\pi_{u,v}^{f_{i,j}}(t)$	二进制变量， t 时刻 i 中第 j 条虚拟路径映射在链路 $l_{uv} \in L$ 上，则为 1

4.2.3 资源调整优化模型

本节对 SFC 动态调整问题正式提出数学定义。首先对问题进行相关约束，其次明确问题的优化目标。

(1) 相关约束

一个有效的调整策略必须确保满足以下条件：

由于每个服务器上 CPU 资源是有限的，每个服务器只能支持有限个 VNF。因此，可以得到约束：

$$\sum_{i \in I} \sum_{j \in F_i} z_{iju} \cdot r_j^i \leq C_u, \forall u \in N \quad (4-5)$$

同时，分配给相应的 SFC 的转发路径带宽不能超过实际物理链路的带宽资源。因此，可以得到约束：

$$\sum_{i \in I} \sum_{f_{i,j} \in F_i} b_i \cdot \pi_{(u,v)}^{f_{i,j}} \leq B_{u,v}, \forall (u,v) \in L \quad (4-6)$$

与文献[52]相同，SFC 中的每个 VNF 只能部署在单个物理节点上，不能由多个物理节点服务。因此，可以得到约束：

$$\sum_{u \in N} z_{iju} \leq 1, \forall i \in I, \forall j \in F_i \quad (4-7)$$

除此之外，传输路径的总延迟应该小于请求所需的延迟。因此，可以得到约束：

$$d_i \leq D_i, \forall i \in I \quad (4-8)$$

其中，请求 i 的传输时延 d_i 定义如下：

$$d_i = \sum_{f_{i,j} \in i} \sum_{(u,v) \in L} \tau_{uv} \cdot \pi_{(u,v)}^{f_{i,j}}(t) \quad (4-9)$$

(2) 优化目标

为了有效应对 SFC 的需求变化，保证服务的连续性和质量，实现其动态资源调整具有至关重要的意义。其中如何在动态调整过程中，不仅满足用户的实时需求，还要降低整体的动态调整成本，同时优化资源的整体分布是本文所需要解决的。前文已经设定了确保用户 QoS 满足的约束条件，并详细定义了调整成本。本节将进一步精细化资源分布的定义，构建一个旨在综合考虑成本效益、用户需求满足度和资源分布最优化的多目标多约束优化问题模型。

资源分布的合理性对于网络性能和成本效益至关重要，本文借鉴文献[53]中的做法，采用标准差作为评价物理资源分布均衡性的关键指标。

$$RD = \sqrt{\frac{1}{|N|} \sum_{u=1}^{|N|} (RN_{avi}^u - \overline{RN_{avi}})^2} + \sqrt{\frac{1}{|L|} \sum_{l=1}^{|L|} (RL_{avi}^l - \overline{RL_{avi}})^2} \quad (4-10)$$

其中， $\overline{RN_{avi}}$ 是节点可用资源的平均值， RN_{avi}^u 为单个节点的可用资源， RN_{avi}^u 可以由式 (4-11) 得。 $\overline{RL_{avi}}$ 为链路可用资源的平均值， RL_{avi}^l 为单个链路的可用资源， RL_{avi}^l 可以由式 (4-12) 得。

$$RN_{avi}^u = C_u - \sum_{i \in I} \sum_{j \in F_i} z_{iju}(t) \cdot r_j^i \quad (4-11)$$

$$RL_{avi}^l = B_{u,v} - \sum_{i \in I} \sum_{f_{i,j} \in F_i} b_i \cdot \pi_{(u,v)}^{f_{i,j}}(t) \quad (4-12)$$

而在 T 时段内，动态调整的长期总成本可以定义为：

$$C = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T C_{total}(t) \quad (4-13)$$

其中， $C_{total}(t)$ 可以由式 (4-4) 计算出。

因此优化目标可以整理为：

$$\begin{aligned} & \min C + RD \quad (4-14) \\ st & \left\{ \begin{array}{l} \text{式(4-5) ~ 式(4-8)} \\ z_{iju} \in \{0,1\}, \forall i \in I, \forall j \in F_i, \forall u \in N \\ \pi_{(u,v)}^{f_{i,j}}(t) \in \{0,1\}, \forall f_{i,j} \in i, \forall (u,v) \in L, \forall t \in T \end{array} \right. \end{aligned}$$

其中, 优化目标是最小化动态调整的长期总成本和资源分布标准差。 z_{iju} 为二进制变量, 当请求 i 中第 j 个 VNF 映射在物理节点 u 上时, z_{iju} 为 1, 否则为 0。

$\pi_{(u,v)}^{f_{i,j}}(t)$ 为二进制变量, t 时刻 i 中第 j 条虚拟路径映射在链路 $l_{uv} \in L$ 上, 则 $\pi_{(u,v)}^{f_{i,j}}(t)$ 为 1, 否则为 0。

定理: 优化问题为 NP-hard。

证明: 虚拟网络嵌入 (Virtual Network Embedding, VNE) 问题是指将虚拟图映射到物理图上, 并确保虚拟节点/链路的总容量不超过相应的物理节点和链路的容量, 这一问题被公认为是 NP-hard^[54]。因此, 本文的资源调整问题可以简化为 VNE 问题, 从而证明其为 NP-hard。在本文优化问题中, 存在一个物理网络 $G=(N,L)$, 其中 N 是物理节点的集合, L 是物理链路的集合。本文是想确定一个调整策略, 该策略由两个决策变量 z_{iju} 和 $\pi_{(u,v)}^{f_{i,j}}(t)$ 来约束, 确保符合由式 (4-5) 和式 (4-6) 分别表示的节点和链路资源约束, 并且其中 SFC 可以被视为一个虚拟图。而嵌入成本, 即嵌入虚拟网络所消耗的底层资源量, 可以为 VNE 问题的优化目标^[54]。因此, 可以将本文的目标函数式 (4-14) 简化为仅考虑调整成本的公式。由此, 本文的优化问题可以被视为 VNE 问题。鉴于 VNE 是 NP-hard, 本文的优化问题也同样是 NP-hard。

4.3 问题求解

4.3.1 算法描述

(1) 马尔可夫决策过程

尽管本研究中已建立的 ILP 模型为解决优化问题提供了一个结构化的解决方案, 但在实际应用中, 特别是在面对动态变化和高复杂度的环境时, ILP 模型的求解效率和灵活性往往受限。为了克服这些限制, 同时实现更加动态和自适应的决策制定过程, 本研究采用 DRL 方法。为了将现有的优化问题适配到 DRL 方法中, 需要将优化问题转化为一个马尔可夫决策过程 (Markov Decision Process, MDP) 模型。

MDP 是一种用于描述和求解序贯决策问题的数学框架。在强化学习中, MDP 被广泛应用于建模环境和制定智能体的决策策略, 可以通过与环境交互来学习最佳策略^[55]。MDP 可以由一个五元组表示 (S, A, TP, R, γ) , 其中 S 为一组状态空间, A 为一组动作空间, TP 为状态转移概率, R 为奖励函数, γ 为折扣因子。

状态空间: 表示系统在某一时刻的特定状态。在资源动态调整问题中状态包含两种类型的信息。可以定义为式 (4-15)。其中 $u(t)$ 表示 t 时刻每个物理服务器

上资源剩余的向量，其表达式为 $u(t) = [u_{n_1}(t), \dots, u_{n_{|N|}}(t)]$ 。 $\kappa(t)$ 表示 t 时刻每个类型 VNF 在服务器上部署成功的向量，即 $\kappa(t) = [\kappa_{v_1}(t), \dots, \kappa_{v_{|F|}}(t)]$ ，其中 $\kappa_{v_x}(t)$ 表示 t 时刻 VNF 类型为 v_x 在所有服务器上的映射向量，即 $\kappa_{v_x}(t) = [\kappa_{n_1}^{v_x}(t), \dots, \kappa_{n_{|N|}}^{v_x}(t)]$ ，而 $\kappa_{n_i}^{v_x}(t) \in \{0,1\}$ 表示 t 时刻 VNF 类型为 v_x 是否部署在服务器 n_i 上， $|N|$ 表示服务器数量， $|F|$ 表示 VNF 种类的数量，所以状态空间大小为 $|N| + |N| * |F|$ 。这些信息的整合，更全面地描述当前系统状态，对于决策资源分配和 SFC 调整非常重要。

$$S(t) = \{u(t), \kappa(t)\} \quad (4-15)$$

动作空间：智能体是根据当前状态 $S(t)$ 对资源需求变化的 SFC 选择最佳的调整策略，因此资源调整问题中动作是一个 $|I|$ 维向量，即

$$A(t) = \{a_1(t), \dots, a_{|I|}(t)\} \quad (4-16)$$

其中 $a_i(t)$ 表示在 t 时刻 i 选择的动态调整策略。需要注意的是，在动作选择后，成功的调整要满足约束式 (4-5) 至式 (4-8)。

状态转移概率：在给定状态下执行某个动作后，系统从一个状态转移到另一个状态的概率分布。在本文中，状态转移概率是未知的，因为下一个状态的转移不仅取决于所选择的动作，还取决于外部因素，这些因素增加了网络环境的复杂性和预测难度，导致状态转移概率难以直接获得。

奖赏：奖励信号的设计对于指导智能体学习有效的决策策略至关重要。奖励信号不仅需要准确反映每个动作的即时效果，还应考虑长期的战略目标。在 MDP 中，每个状态转移都会伴随一个奖励信号，表示智能体对该转移的评估或反馈。奖励可以是即时的（立即获得），即：

$$r(t) = \begin{cases} -C_{total}(t) - RD(t), & \text{调整成功} \\ -\sigma, & \text{调整失败} \end{cases} \quad (4-17)$$

在式 (4-17) 中，一个成功的动态调整策略是在不违反约束式 (4-5) 至式 (4-8) 的情况下进行的。在本文动态调整问题中，其目标是实现长期成本最小化并且网络资源标准差最小化，而智能体的目标是最大化长期奖励，因此可以将奖励定义为总成本和资源标准差的负值。此外可以设定一个惩罚项 $-\sigma$ ，其中 σ 远大于总成本和资源标准差，表示对 SFC 调整失败的惩罚，从而使智能体在尽量减小总成本和均衡资源分布的同时，遵守约束条件。与文献[56]中为多个目标分别设定奖赏不同，本文通过设计一个统一的奖励信号，有效避免了在优化过程中需要单独处理每个目标的复杂性，从而减少了智能体的学习负担。这种综合性的奖励设计不仅有助于智能体更快地学习到在不同目标间权衡的策略，还提高了算法的收敛速度和效率。

在强化学习框架中，智能体的学习目标是通过交互与环境，以实现长期累积

奖励的最大化。这要求智能体不只是寻求立即的奖励，而是在整个决策过程中考虑每一步骤的长远影响。在本研究的 SFC 动态资源调整问题中，这一目标尤为关键，因为智能体的每一次资源调整决策都可能对未来的网络性能和成本产生深远影响。则在 t 时隙的累计奖励值为：

$$R(t) = r(t) + \gamma \cdot r(t+1) + \gamma^2 \cdot r(t+2) + \dots = \sum_{n=0}^{\infty} \gamma^n \cdot r(t+n) \quad (4-18)$$

其中， γ 为折扣因子，影响了智能体对未来奖励的考虑程度， n 为迭代次数。

(2) 基于深度强化学习的 SFC 动态调整算法

MDP 通常通过动态规划或强化学习方法来解决^[57]。在面对 SFC 动态资源调整问题时，传统的动态规划方法虽然理论上能够找到最优解，但实际应用中受限于其对完整模型信息的需求以及处理大规模状态空间时的计算复杂性。动态规划方法的这些限制使得它难以适应于那些模型信息不完全或状态空间庞大的动态环境，如 SFC 的动态调整问题。这个问题的复杂性在于，服务功能链的需求是不断变化的，而且这些变化具有随机性，这导致状态转移概率未知，使得问题的状态空间和动作空间极为庞大且复杂。而 RL 方法适用于状态空间较大、无法事先获得完整模型的问题。它可以通过智能体与环境的交互来学习最优策略，并且能够应对复杂的问题和不确定的环境。因此本文采用深度强化学习方法来解决，其结合了传统强化学习在无模型情况下通过探索学习最优策略的优点，以及深度学习在处理高维数据方面的强大能力。在第二章中已介绍了相关深度强化学习相关方法，传统 DQN 中存在过估计问题，这可能导致学习过程的不稳定和次优策略的学习。在此背景下，考虑到 DDQN 在处理高维状态空间和学习稳定性方面的优势，因此提出基于 DDQN 的智能服务功能链动态调整算法 (Intelligent Service Function Chains Dynamic Adaptation Algorithm, ISFCDA)，以解决 SFC 调整中的挑战。

ISFCDA 利用 DDQN 的能力，根据 SFC 的需求变化动态调整 VNF 资源的分配。考虑到资源需求的连续变化和保持 SLA 的要求，算法旨在优化资源分布，同时确保高质量的服务交付。前文已介绍了调整算法的数学模型，并讨论了其关键组成部分，包括状态表示、动作选择、状态转移概率和奖励计算。此外，介绍了 DDQN 算法的训练过程，其中包括与环境的交互和经验回放，以提高数据效率并增强学习过程。接下来本文将深入讨论 ISFCDA 的细节。

如算法 1 所示，第 1 至 18 行为 ISFCDA 训练的过程，算法开始时，初始化一个经验池 E ，用于存储 (S_t, a_t, S_{t+1}, r) (第 1 行)。同时初始化主网络和目标网络的参数。(第 2 至 3 行)。对 ISFCDA 进行 $episode_{num}$ 次训练，对于每个 $episode$ ，初始化状态并对资源变化的 SFC 进行以下步骤：

1. 主网络中使用 ϵ 贪婪策略选择动作 A (第 7 行)；

2. 根据选择的动作，调用特定的子算法（算法 2、算法 3 或算法 4）对 SFC 进行相应的调整（第 8 至 13 行）；
3. 算法根据所做的调整获得奖励（其计算由式（4-17）可得）和下一状态 S_{t+1} ，并存入经验池 E （第 14 行）；
4. 在经验池中积累足够的经验后，算法随机采样一批样本，并使用梯度下降法更新主网络的参数，由式（2-5）可得（第 15 行）；
5. 在更新 Y 次后，将主网络的参数复制给目标网络，以稳定和改进学习过程（第 16 行）。

训练结束后，保存训练的网络模型，将资源变化的 SFC 输入算法中，可由主网络获得 SFC 的调整策略 π （第 19 行）。

算法 1 智能服务功能链动态调整算法

输入：惩罚值 σ 、初始探索率 ε 、折扣因子 γ 、学习率、更新次数 T ，资源变化的 SFC

输出：调整策略 π

1. 初始化经验池 E
 2. 初始化主网络的参数 θ
 3. 初始化目标网络的参数 $\theta' = \theta$
 4. for $episode=1$ to $episode_num$ do
 5. 初始化状态 $state = \{u_1, \dots, u_N; \kappa_1, \dots, \kappa_M\}$
 6. for $i=1$ to sfc_num do
 7. 在主网络中采取 ε -贪婪策略选取动作 A
 8. if $A==0$ then
 9. 调用算法 2
 10. else if $A==1$ then
 11. 调用算法 3
 12. else if $A==2$ then
 13. 调用算法 4
 14. 获得奖励 r 和下一状态 S_{t+1} ，将 (S_t, a_t, S_{t+1}, r) 放入经验池 E
 15. 从经验池 E 中随机取小批量样本，使用梯度下降更新主网络参数 θ
 16. 在更新 Y 次后将主网络参数 θ 复制给目标网络
 17. end for
 18. end for
 19. 通过主网络得到 SFC 调整策略 $\pi = \{\pi_1, \pi_2, \dots\}$
-

其中动作的执行，如选取动作为 SFC 上 VNF 在当前节点上进行资源调整时，如算法 2 所示，若 VNF 的需求 r_j^i 小于当前节点上剩余资源 $C_{u,avi}$ 则在当前节点上调整资源（第 3 行），返回调整后的状态、奖励及策略（第 8 行）。否则，调整失败（第 5 行）。

算法 2 缩放

输入: SFCs

输出: 调整结果 π_i

1. for j in F_i
 2. if $r_j^i < C_u.avi$ then
 3. 在当前节点上扩大/减少实例资源
 4. else
 5. return state, $-\sigma$, failure, π_i
 6. end if
 7. end for
 8. return state, r, success, π_i
-

当选取动作为 SFC 上 VNF 在新的节点上创建实例时，如算法 3 所示，查找候选节点，满足资源需求，再按照节点剩余资源降序排序（第 1 行）。接着对候选节点进行筛选，当在该节点实例后，链路资源不满足或者违法了时延则将节点移出，否则，记录实例化后 SFC 的转发路径长度（第 3 至 8 行）。如果候选节点为空，则调整失败（第 9 至 10 行）。相反，则说明调整成功，返回转发路径最短所对应的策略，及状态和奖励（第 12 行）。

算法 3 实例化 VNF

输入: SFCs

输出: 调整结果 π_i

1. 查找候选节点 $Candia_node$ ，满足 VNF 资源且无相同的实例，并按照剩余资源降序排序
 2. for node in $Candia_node$
 3. if 在节点上实例化后，链路资源满足且时延满足 then
 4. 记录转发路径
 5. else
 6. 将候选节点移出
 7. end if
 8. end for
 9. if $Candia_node == \text{NULL}$ then
 10. return state, $-\sigma$, failure, π_i
 11. else
 12. return state, r, success, 转发路径最短的 π_i
 13. end if
-

当选取动作为对 VNF 进行迁移，除了算法 4 中候选节点是找含有相同实例的节点外，其他过程与算法 3 一样，不再赘述。

算法 4 迁移 VNF

输入: SFCs

输出: 调整结果 π_i

1. 查找候选节点 $Candia_node$, 满足 VNF 资源且有相同的实例, 并按照剩余资源降序排序
 2. for node in $Candia_node$
 3. if 迁移到当前节点, 链路资源满足且时延满足 then
 4. 记录转发路径
 5. else
 6. 将候选节点移出
 7. end if
 8. end for
 9. if $Candia_node == \text{NULL}$ then
 10. return state, $-\sigma$, failure, π_i
 11. else
 12. return state, r, success, 转发路径最短的 π_i
 13. end if
-

4.3.2 算法复杂度分析

算法 2 的时间复杂度为 $O(|F_i|)$, 其中 $|F_i|$ 表示每个服务请求中的 VNF 数量。算法 3 和算法 4 的时间复杂度均为 $O(|N|)$, 其中 $|N|$ 代表物理节点的数量。 $O(|d_{in}| \cdot |d_{out}|)$ 表示全连接网络的时间复杂度, 其中 $|d_{in}|$ 是输入维度, $|d_{out}|$ 是输出维度。因此, 算法 1 的时间复杂度为 $O(episode_{num} \cdot |I| \cdot |F_i| \cdot |N|^2 \cdot d_{in} \cdot d_{out})$, 其中 $|I|$ 表示请求的数量, $episode_{num}$ 代表训练次数。

4.4 仿真设计结果及分析

在本节中, 详细阐述了使用 Pytorch 框架进行的仿真实验设置, 以全面评估本文提出的 ISFCDA 的性能。首先介绍了本文用来评估算法的仿真设置, 包括网络拓扑结构、不同规模的服务需求变化以及资源配置条件。其次, 通过与其他两种算法的性能比较, 直观地展示了 ISFCDA 在资源均衡方面的有效性, 以及 ISFCDA 在长期可持续收益方面的明显优势。

4.4.1 实验方法及仿真参数设置

实验中的物理网络是来自 SDNlib^[58]中的 Nobel-Germany 拓扑，该物理网络共有 17 个节点，26 条链路。为了更接近真实网络环境，每个的 CPU 资源初始化在 (30, 50) 之间，模拟了不同服务器的计算能力，而物理链路的带宽为 10 Gbps。由于可以得到拓扑结构的节点的物理坐标。因此，相邻物理节点之间的时延与它们之间的物理距离成正比^[33]，链路单位时延为 1。当违法约束时，惩罚参数 σ 设为 10000。资源的调整成本计算为线性函数，成本相关系数均为 1。相关参数设置如表 4-2。

表 4-2 关键参数和值

参数	值
节点数	17
链路数	26
VNF 数量	6
节点容量	(30, 50)
链路带宽	10 Gbps
链路单位延迟	1ms
成本相关系数	$\alpha_1=1, \alpha_2=1$

在单个时隙下，假设已有 SFC 使用第三章中的算法放置在网络中。每个请求 i 的生成，及不同服务中每个网络功能所需 CPU 计算资源与文献[33]中的一致。其所需带宽资源和延迟要求如表 4-3。

表 4-3 SFC 相关参数

服务类型	SFC	带宽	时延
Web service	NAT-FW-TM-WOC-IDS	100kbps	500ms
VoIP	NAT-FW-TM-FW-NAT	64kbps	100ms
Video Streaming	NAT-FW-TM-VOC-IDS	100kbps	500ms

本节将采用 ISFCDA 算法在此基础上进行动态资源调整。本文 ISFCDA 算法中使用 Pytorch 建立 3 层全连接网络，其主要参数如表 4-4 所示。输入层同状态维度，隐藏层包含 128 个神经元，所有神经元均由线性整流函数 (Rectified Linear Unit, ReLU) 作为激活函数。在 DDQN 中，贪心探索的作用是平衡探索和利用，但如果仅使用贪心策略，可能会导致算法陷入局部最优解中，无法发现更优的解决方案。因此，为了保证算法的探索性，本文引入了 ϵ -贪婪策略，初始探索率为 0.9，会随着 $\epsilon(t+1) = \max(\epsilon_{\min}, \epsilon(t) - decay_{rate})$ 下降。为了提高学习过程的样本利用效率，减少连续样本之间的相关性，将记忆池大小设为 600。并采用 Adam 优化器，将学习率设为 0.015，批量大小设为 32。除此之外，为了提高稳定性、减少目标值偏差，将更新周期设为 200。

表 4-4 ISFCDA 相关参数

参数	值
惩罚参数 σ	10000
初始探索率 $\varepsilon(t)$	0.9
最小探索率 ε	0.05
衰减率	0.005
输入层	119
批量大小	32
折扣因子 γ	0.95
记忆池大小	600
优化器	学习率为 0.015 的 Adam 优化器
目标网络更新周期	200

4.4.2 对比算法介绍

为了全面评估本文提出的 ISFCDA 算法的性能,并验证其在解决 SFC 资源动态调整问题中的优越性,本文与以下两种方法进行比较:

DQN: 作为一种经典的深度强化学习方法, DQN 算法已被广泛应用于各种优化问题中。在本实验中,通过将 DQN 算法应用于 SFC 资源动态调整的问题模型,能够评估深度强化学习方法在本领域的基线性能。

REAP^[33]: 一种启发式算法,旨在解决虚拟化数据中心中 VNF 重配置问题,即在考虑垂直缩放、水平缩放和迁移作为重配置解决方案的情况下,最小化 NFVI 中的能源消耗和重配置开销。

4.4.3 评价指标

在本文中,为了全面评估 ISFCDA 算法在资源动态调整领域的性能,参照文献[53]的方法,定义了四个关键的评价指标:

资源分布: 衡量物理资源分布的均衡性。均衡的资源分布可以有效避免资源瓶颈和过载,保证网络服务的稳定性和可靠性。可以由式(4-10)所得。

总成本: 对 T 时隙需求变化的 SFC 进行调整所花费的总开销。可以由式(4-13)所得。

接受率: 在 T 个时隙满足 SFC 变化的接受率。较高的接受率意味着网络能够更好地适应需求变化,从而提供更加可靠和连续的服务

$$AC(T) = \frac{\sum_{t=0}^T AC_{suc}(t)}{\sum_{t=0}^T |I|} \quad (4-18)$$

其中, $|I|$ 为资源产生变化的 SFC 数量, $AC_{suc}(t)$ 为 t 时刻满足 SFC 需求的数

量。

长期收益： T 时隙网络运营商的利润，反映了算法在运营环境中的经济可行性和效益。

$$Pro(T) = \int_{t=0}^T \sum_{i \in I} (Ben_{cpu} \sum_{j \in F_i} r_j^i + Ben_{bw} \sum_{f_{i,j} \in \mathcal{F}_i} b_{f_{i,j}}) \quad (4-19)$$

其中， Ben_{cpu} 为单位 CPU 资源收益， Ben_{bw} 为单位带宽资源收益。在本文中， Ben_{cpu} 和 Ben_{bw} 均设为 1。

4.4.4 实验结果与分析

(1) 实验 1：不同 SFC 数量下的比较

首先本文采用三种算法对单时隙不同数量的 SFC 进行资源动态调整。如图 4-2 所示，本文提出的 ISFCDA 在不同数量下，均能够在调整资源后，使网络达到最好的资源分布。而 DQN 的效果略逊于 ISFCDA，而 REAP 调整后，资源分布情况最差。ISFCDA 平均资源分布标准差比 DQN 低 2.41%，比 REAP 低 9.90%。如图 4-3 所示，在 50 条 SFC 下，ISFCDA 的资源分布标准差比 DQN 低 6.11%，比 ISFCDA 低 15.47%。

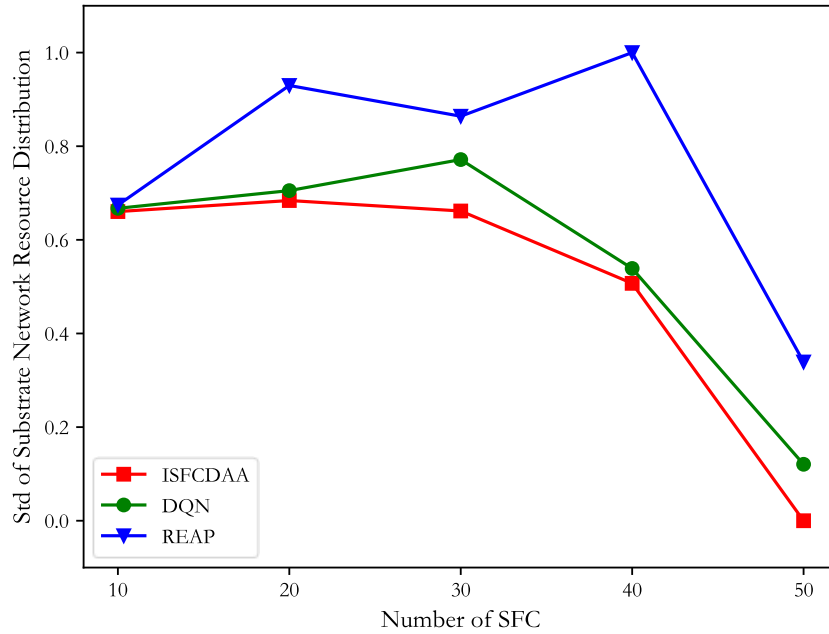


图 4-2 物理资源分布

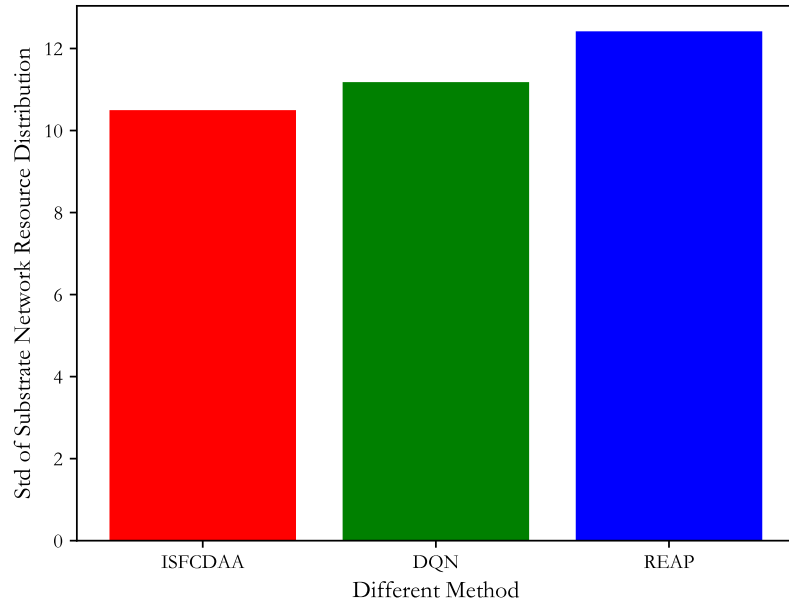


图 4-3 在 50 条 SFC 下物理资源分布

但是如图 4-4 所示, 本文算法调整成本略高于 REAP 算法, 在 SFC 数量增加到 20 条后, ISFCDA 算法总成本低于 DQN, 高于 REAP。ISFCDA 总成本平均比 REAP 高 14.29%, 比 DQN 低 5.14%。这意味着在资源调整过程中, ISFCDA 算法需要更多的成本来实现最佳的资源分布。

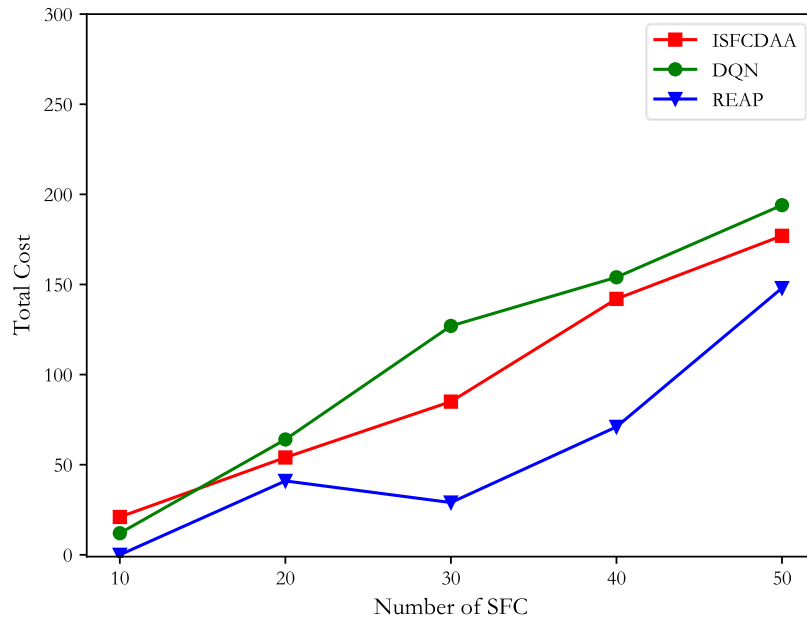


图 4-4 总成本

在本文中, 我们总成本包括实例化成本和迁移成本。如图 4-5、图 4-6 所示, ISFCDA 算法在实例化新的实例和进行迁移两种调整方案, 选取较为均衡, 从

而使资源分布更有效。而 REAP 更倾向于进行 VNF 的实例化，它涉及的重配置开销相对较少，这与文献[33]描述相同。而本文算法在追求资源分布均衡的过程中，可能选择调整成本较高的策略。因此在总成本上 ISFCDA 高于 REAP。

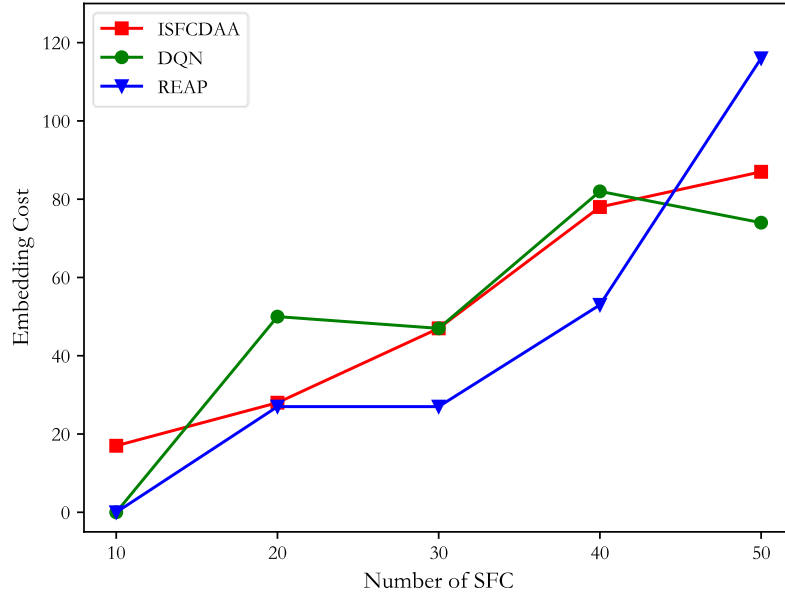


图 4-5 实例化成本

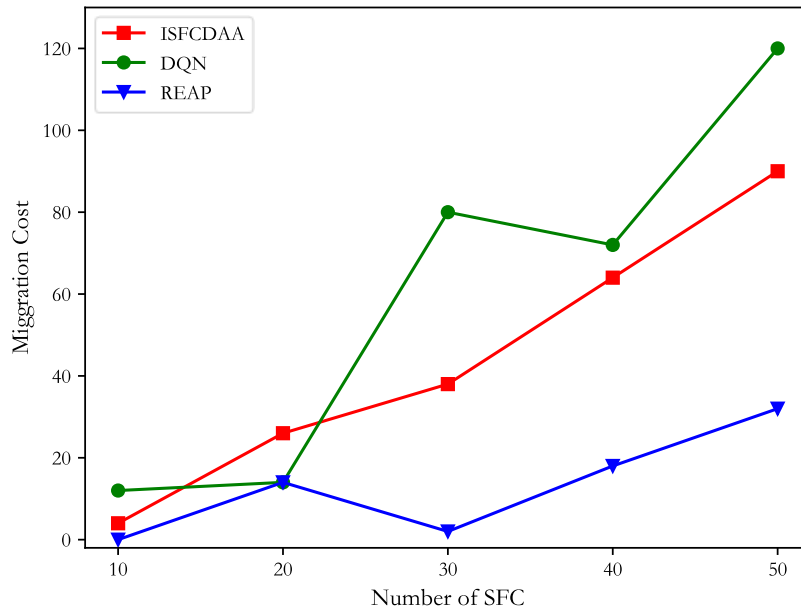


图 4-6 迁移成本

(2) 实验 2：长期收益比较

由于本文算法在单时隙下可以达到资源分布最优，但是成本略高于基准算法 REAP，为了进一步验证 ISFCDA 的有效性，本文对比了三种算法在 35 个时隙

下 SFC 的接受率，其中每个时隙有 20 条 SFC 产生资源变化。如图 4-7 所示，ISFCDA 算法整体来看，能够更好地适应 SFC 资源变化，并通过资源调整实现较高的接受率。ISFCDA 平均接受率比 DQN 高 15.11%，比 REAP 高 39.57%。DQN 相较于 REAP 表现更好，虽然不及 DDQN，但仍能在一定程度上适应 SFC 资源变化，并实现较高的接受率。REAP 算法在资源变化较大的 SFC 上表现不佳，无法满足其资源需求。因为 REAP 的资源分布最差，在资源变化较大的情况下无法灵活适应，后续 SFC 需求变化时，物理网络资源不均，导致 SFC 即使想要迁移或者实例新的 VNF，其 QoS 也无法满足，从而接受率较差。

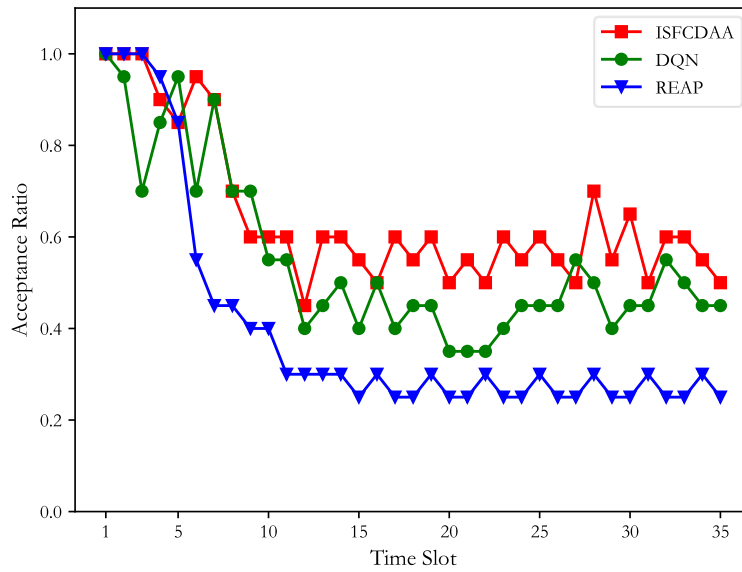


图 4-7 接受率

如图 4-8 所示，在前两个时隙下 REAP 的总成本最低，后逐渐增大，甚至高于其他两个算法，在第 5 个时隙后，总成本骤然降低，之后一直维持在较低水平。至于 REAP 为什么在图 4-8 中总成本降至为 0，是因为如图 4-4、图 4-5、图 4-6 所示，REAP 在重配置过程中产生的成本最低，但资源分布的标准差最高。这表明 REAP 更倾向于成本控制，忽视了资源分布，导致每个节点上的资源碎片化。随着时间的推移，SFC 资源发生变化，后续的调整变得具有挑战性，甚至可能违反 SLA。因此，水平扩展和迁移变得不可行，导致调整成本降至零。而本文算法在前 5 个时隙前总成本处于中间，比 DQN 总成本低 16.92%，比 REAP 高 11.09%。在第五个时隙后，总成本降低，之后成本虽高于 REAP，但也维持在较低水平，平均比 DQN 总成本低 7.99%。但 REAP 和 DQN 的接受率远不如 ISFCDA。

因此本文比较了在 35 个时隙下，各算法进行调整后所带来的收益，如图 4-9 所示。可以看到，ISFCDA 所带来的长期收益最高，而 DQN 其次，REAP 的长期收益最低。ISFCDA 平均长期收益比 DQN 高 21.47%，比 REAP 高 42.92%。

这证明了 ISFCDA 在资源动态变化环境下的灵活性和优越性。同时，它还突出了 DQN 相对于 REAP 的性能优势，以及 REAP 在资源变化较大的情况下的限制。

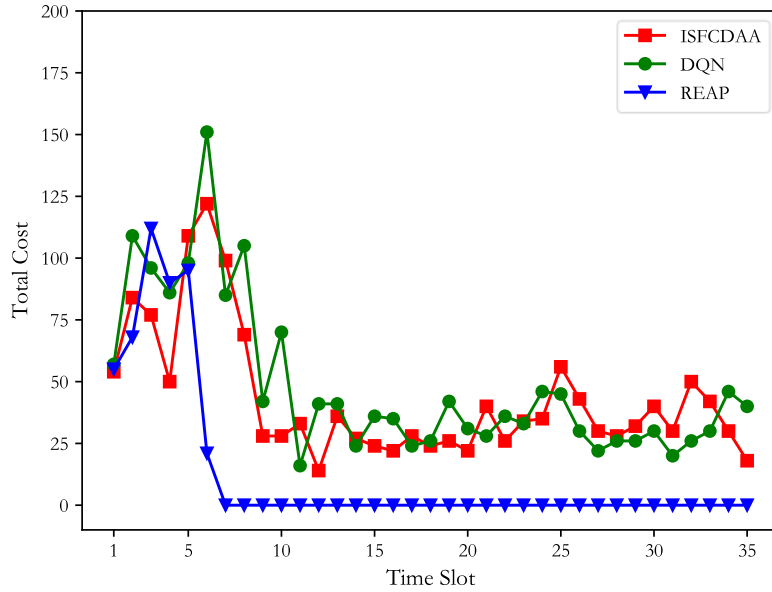


图 4-8 总成本

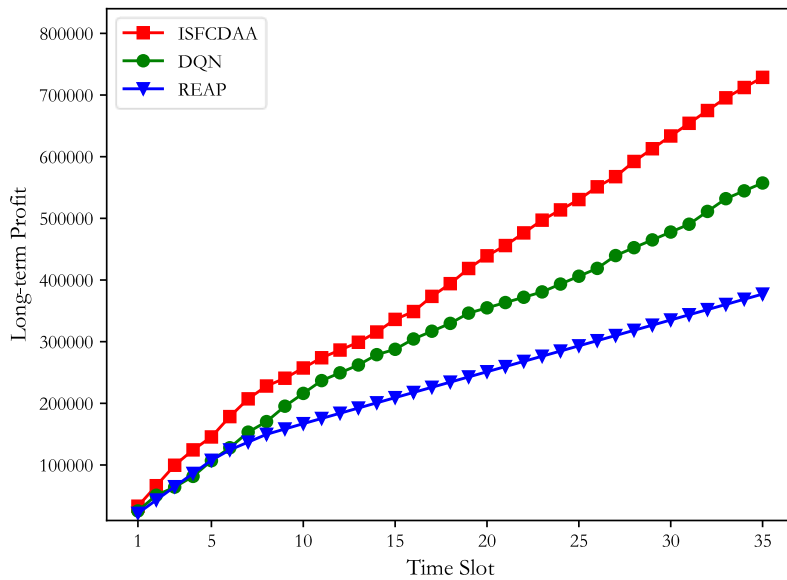


图 4-9 长期收益

4.5 本章小结

本文提出了一种基于深度强化学习的智能调整算法，针对网络功能虚拟化架

构下 SFC 资源需求的实时变化，以解决动态网络环境中资源管理的复杂问题。本文首先将问题形式化为 ILP，并将优化过程转换为 MDP，从而为 DRL 提供了适用的问题框架。接着提出了一个高效的算法，通过综合考虑垂直扩展、创建新的 VNF 实例和 VNF 迁移来解决问题，该算法可以根据需求变化，智能的选择合适的策略，实现资源利用的最优化，确保网络资源的高效均衡分布，并在满足用户需求的同时最小化相关的调整开销。最后，评估了 ISFCDA 与其他调整方法的性能。仿真结果表明，ISFCDA 在单时隙下能够达到更均衡的资源分布。此外，从长期来看，ISFCDA 的接受率和长期收益也明显优于 DQN 和 REAP 算法，为 NFV 编排器的资源管理提供了有效的解决方案。

5 结论与展望

5.1 主要工作总结

为了适应 5G 网络中各种垂直行业的多元化需求，SFC 技术成为了提供个性化网络服务的关键技术。而随着网络环境的不断演进，SFC 的生命周期管理和资源调整成为了确保资源分配均衡和网络服务质量的关键因素。为此，本文围绕 SFC 部署和动态资源调整机制进行了系统性的研究和实验验证，提出了一系列创新解决方案，以提高网络资源利用效率，降低运营成本，同时保障用户服务的高质量。具体工作内容如下：

1. 本文首先探讨了 SFC 的部署及其资源动态调整的研究领域，通过对现有文献的综合分析，梳理了该领域的研究现状和挑战，明确了本文的研究内容。此外，本文还深入探讨了与 SFC 部署和资源调整相关的理论基础，包括 NFV、SDN、以及深度强化学习等前沿技术，为本研究的理论框架和方法论提供了坚实的基础。
2. 针对 SFC 部署过程中，存在 VNF 实例部署成本和转发路径成本难以权衡的问题，本文首先针对多链 SFC，建立 VNF 和虚拟链路映射模型，并预估路径部署长度上限，保证 SFC 时延需求。其次，在路径部署长度限制范围内，尽可能使 VNF 实例共享最大化，以平衡链路转发成本和 VNF 部署成本，得到 SFC 部署策略。最终与已有的 SPH 和 GUS 部署算法相比，验证了本文算法可以在保证用户服务质量的同时，有效降低 SFC 部署成本。
3. 在 SFC 部署后，面对网络流量的增大和复杂的波动，本文研究了 VNF 实例资源的动态调整问题，以应对网络资源空闲或满载的状态，从而有效地均衡资源，避免资源浪费或网络拥堵。本研究首先建立了问题模型，并证明了问题是 NP-hard，接着综合考虑了水平缩放、垂直缩放和迁移三种策略，提出了一种基于 DDQN 的智能调整算法，以实现网络资源的平衡分配，最小化调整成本，满足用户需求。最后通过仿真实验，与已有的 DQN 和 REAP 调整算法相比，验证了所提出动态调整策略的有效性。

5.2 创新点

在当前的网络技术研究领域，虽然 NFV 编排已经取得了一定的进展，但大多数研究依旧局限于对 NFV 编排的单一阶段进行优化，未能全面覆盖 SFC 的整

个生命周期，尤其是在考虑 VNF 实例共享对部署和调整阶段综合成本优化的影响方面。本研究工作通过跨阶段考虑 SFC 的部署与动态调整，提出了一系列创新性的优化策略，旨在实现 SFC 部署和资源动态调整的成本效率最大化，具体的创新点包括但不限于以下几方面：

1. 本文的第一个主要创新点在于提出了一个全新的基于 VNF 实例共享的 SFC 部署模型，该模型不仅针对 VNF-FG，还巧妙地处理了 VNF 实例部署成本与链路转发成本之间的潜在矛盾。利用 ILP 模型对问题进行建模，在保证用户 QoS 要求的同时，寻求最小化整体运营成本的部署方案。与此同时，本研究还引入了动态规划思想的启发式算法，以更高效地解决部署问题。与目前工作往往针对的都是单链 SFC，且只单方面考虑 VNF 实例的部署成本或者链路转发成本不同。最终与已有的算法相比，本文算法所得的总运营成本分别降低 6.6% 和 12.15%，且当 SFC 数量增多时，本文算法的服务接受率可达 89.33%。
2. 面对网络资源变化时，本文建立了多目标优化模型，不仅考虑了调整成本，还考虑了一个在以往相关工作中大多忽视的维度，即物理资源分布的平衡。现有的工作，往往采用单个调整策略，而本文同时考虑 VNF 实例垂直缩放、水平缩放和迁移策略，充分挖掘了这些策略在综合应用中的潜力。并在本文场景下，提出了一种基于 DDQN 的智能调整算法，用以解决具有大状态空间、动作空间以及未知转移概率的问题。通过智能学习与决策，ISFCDA 算法能够在时变的网络环境下实现资源的有效调整，显著提高了网络管理的灵活性和成本效率。与已有的算法相比，本文 ISFCDA 平均资源分布标准差最高降低了 9.90%，而 ISFCDA 的平均接受率最高提高了 39.57%，平均长期收益最高提高了 42.92%。

5.3 不足与展望

虽然本研究在 SFC 的部署与动态资源调整领域已经取得了一些重要进展，展现了基于 VNF 共享的部署算法和基于深度强化学习的动态调整策略的有效性，但在研究过程中也暴露出一些局限性和未来研究的可能方向：

1. 本文提出的基于 VNF 共享的 SFC 部署算法虽然在 SFC 部署成本方面展现出了优越性，但在实际应用中仍存在一定的局限性。当前提出的启发式部署算法在小规模的网络环境下效果显著，但面对大规模网络系统时，算法的计算效率及时效性可能受到挑战。特别是启发式算法在处理极大规模网络时的效率问题，随着网络规模的扩大，计算和时间成本随之增加，这对于实时或近实时的网络服务管理场景构成了挑战。因此，未来的研究

中需要探索更为高效的算法或优化策略，或探索算法的并行化和分布式处理策略，以进一步降低算计算开销。

2. 虽然本文采用的基于深度强化学习的 SFC 资源调整策略在促进资源利用效率和满足服务质量方面取得了一定的成效，但由于为了专注于验证核心思想的有效性和算法的实用性，本文主要关注的是 SFC 单链情况。然而，实际网络环境中 SFC 资源调整往往更为复杂，可能涉及多链情况等更多维度的考量，在处理更广泛场景和更复杂需求方面还存在一定的局限。因此，未来的研究需要进一步考虑这些复杂因素，通过改进现有模型或开发新的策略，来提高算法在更广泛和复杂网络环境下的适用性和鲁棒性。
3. 当前研究在 NFV 编排中主要集中于 SFC 的部署和资源动态调整方面，而对于 SFC 的调度策略并未进行深入研究。SFC 调度作为网络服务管理的另一个关键环节，涉及到如何根据网络状态和服务需求动态地安排 SFC 中 VNF 实例的执行顺序和路径选择，以优化整体网络性能和服务质量。有效的 SFC 调度策略不仅可以减少服务延迟，提升用户体验，还能进一步提高网络资源的利用效率，降低运营成本。然而，SFC 调度问题在考虑多链环境、不同服务等级要求及实时网络状态变化的条件下变得更加复杂和挑战性。因此，未来研究需深入探讨 SFC 调度的优化问题，研发能够适应多变网络环境和满足多样化服务需求的智能调度算法，为网络服务提供商在 NFV 编排管理中提供更全面的解决方案。

参考文献

- [1] ETSI NFV ISG. Network Functions Virtualization-White Paper3[EB/OL]. <https://portal.etsi.org/Portals/0/Tbpages/NFV/Docs/>, 2014-10-14
- [2] Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: State-of-the-art and research challenges[J]. IEEE Communications surveys & tutorials, 2015, 18(1): 236-262.
- [3] Attaoui W, Sabir E, Elbiaze H, et al. Vnf and cnf placement in 5g: Recent advances and future trends[J]. IEEE Transactions on Network and Service Management, 2023, 20(4): 4698-4733.
- [4] Hu F, Hao Q, Bao K. A survey on software-defined network and openflow: From concept to implementation[J]. IEEE Communications Surveys & Tutorials, 2014, 16(4): 2181-2206.
- [5] Cao J, Zhang Y, An W, et al. VNF-FG design and VNF placement for 5G mobile networks[J]. Science China Information Sciences, 2017, 60: 1-15.
- [6] Bu C, Wang J, Wang X. Towards delay-optimized and resource-efficient network function dynamic deployment for VNF service chaining[J]. Applied Soft Computing, 2022, 120: 108711.
- [7] Fang J, Zhao G, Xu H, et al. Reveal: Robustness-aware VNF placement and request scheduling in edge clouds[J]. Computer Networks, 2023, 233: 109882.
- [8] Ali K, Jammal M. Proactive VNF scaling and placement in 5G O-RAN using ML[J]. IEEE Transactions on Network and Service Management, 2023, 21(1): 174-186.
- [9] Yang B, Chai W K, Xu Z, et al. Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications[J]. IEEE Transactions on Network and Service Management, 2018, 15(1): 475-488.
- [10] Huang M, Liang W, Ma Y, et al. Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement[J]. IEEE Transactions on Cloud Computing, 2019, 9(4): 1535-1548.
- [11] 金明, 李琳琳, 张文瑾, 等. 基于深度强化学习的服务功能链映射算法[J]. 计算机应用研究, 2020, 37(11): 3456-3460+3466.
- [12] Mebarkia K, Zsóka Z. Service traffic engineering: Avoiding link overloads in service chains[J]. Journal of Communications and Networks, 2019, 21(1): 69-80.
- [13] Xu Z, Liang W, Huang M, et al. Efficient NFV-enabled multicasting in SDNs[J]. IEEE Transactions on Communications, 2018, 67(3): 2052-2070.
- [14] 唐伦, 贺兰钦, 连沁怡, 等. 基于改进深度强化学习的虚拟网络功能部署优化算法[J]. 电子与信息学报, 2021, 43(06): 1724-1732.
- [15] Aklamanu F, Randriamasy S, Renault E. Utility and A*-based algorithm for network slice placement and chaining[C]. 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.
- [16] Chen X, Shi J, He Y, et al. Deployment strategy of resource optimized service function chain in NFV environment[C]. International Conference on Cryptography, Network Security, and Communication Technology (CNSCT 2022). SPIE, 2022, 12245: 130-137.
- [17] Zhang Q, Liu F, Zeng C. Online adaptive interference-aware VNF deployment and migration for 5G network slice[J]. IEEE/ACM Transactions on Networking, 2021, 29(5): 2115-2128.
- [18] Luo Z, Wu C. An online algorithm for VNF service chain scaling in datacenters[J]. IEEE/ACM

- Transactions on Networking, 2020, 28(3): 1061-1073.
- [19] 柴蓉, 谢德胜, 陈前斌. 基于成本及功耗联合优化的 SDN 虚拟网络映射算法[J]. 电子学报, 2021, 49(08): 1615-1624.
- [20] Liu J, Lu W, Zhou F, et al. On dynamic service function chain deployment and readjustment[J]. IEEE Transactions on Network and Service Management, 2017, 14(3): 543-553.
- [21] Liu M, Alias S B. Cost-Efficient Virtual Network Function Placement in an Industrial Edge System: A Proposed Method[J]. IEEE Systems, Man, and Cybernetics Magazine, 2023, 9(1): 10-17.
- [22] Bari F, Chowdhury S R, Ahmed R, et al. Orchestrating virtualized network functions[J]. IEEE Transactions on Network and Service Management, 2016, 13(4): 725-739.
- [23] 陈嘉亮, 王丰, 张潇. 移动边缘计算网络下的服务功能链部署优化设计[J]. 计算机应用研究, 2022, 39(10): 3108-3113.
- [24] 朱国晖, 景文焕, 李世昌. 基于改进麻雀搜索算法的服务功能链优化映射算法[J]. 计算机应用研究, 2022, 39(07): 2120-2123+2131.
- [25] Shang X, Liu Z, Yang Y. Network congestion-aware online service function chain placement and load balancing[C]. Proceedings of the 48th International Conference on Parallel Processing. Association for Computing Machinery, 2019: 1-10.
- [26] Mao M, Humphrey M. Auto-scaling to minimize cost and meet application deadlines in cloud workflows[C]. Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2011: 1-12.
- [27] Savi M, Tornatore M, Verticale G. Impact of processing-resource sharing on the placement of chained virtual network functions[J]. IEEE Transactions on Cloud Computing, 2019, 9(4): 1479-1492.
- [28] Zhao D, Sun G, Liao D, et al. Mobile-aware service function chain migration in cloud-fog computing[J]. Future Generation Computer Systems, 2019, 96: 591-604.
- [29] Hawilo H, Jammal M, Shami A. Orchestrating network function virtualization platform: Migration or re-instantiation?[C]. 2017 IEEE 6th International Conference on Cloud Networking (CloudNet). IEEE, 2017: 1-6.
- [30] Đani V, Huljenić D, Ožegović J. Virtual network resource optimization model for network function virtualization[J]. Wireless communications and mobile computing, 2021, 2021: 1-21.
- [31] Eramo V, Ammar M, Lavacca F G. Migration energy aware reconfigurations of virtual network function instances in NFV architectures[J]. IEEE Access, 2017, 5: 4927-4938.
- [32] Liu D, Zhou Z, Zhang D, et al. Efficient service reconfiguration with partial virtual network function migration[J]. Computer Networks, 2024, 241: 110205.
- [33] Padhy S, Chou J. Reconfiguration aware orchestration for network function virtualization with time-varied workload in virtualized datacenters[J]. IEEE Access, 2021, 9: 48413-48428.
- [34] Eramo V, Miucci E, Ammar M, et al. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures[J]. IEEE/ACM Transactions on Networking, 2017, 25(4): 2008-2025.
- [35] Ayoubi S, Zhang Y, Assi C. A reliable embedding framework for elastic virtualized services in the cloud[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 489-503.
- [36] Houidi O, Soualah O, Louati W, et al. An efficient algorithm for virtual network function scaling[C]. GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE, 2017: 1-7.

- [37] Harutyunyan D, Behraves R, Slamnik-Kriještorac N. Cost-efficient placement and scaling of 5G core network and MEC-enabled application VNFs[C]. 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2021: 241-249.
- [38] Chen Z, Li H, Ota K, et al. HyScaler: a dynamic, hybrid VNF scaling system for building elastic service function chains across multiple servers[J]. IEEE Transactions on Network and Service Management, 2023, 20(4): 4803-4814.
- [39] Xia W, Wen Y, Foh C H, et al. A survey on software-defined networking[J]. IEEE Communications Surveys & Tutorials, 2014, 17(1): 27-51.
- [40] Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations[J]. IEEE communications magazine, 2015, 53(2): 90-97.
- [41] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
- [42] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of artificial intelligence research, 1996, 4: 237-285.
- [43] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.
- [44] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]. Proceedings of the AAAI conference on artificial intelligence. AAAI Press, 2016, 30(1).
- [45] Beck M T, Botero J F. Coordinated allocation of service function chains[C]. 2015 IEEE global communications Conference (GLOBECOM). IEEE, 2015: 1-6.
- [46] Zhang Q, Liu F, Zeng C. Adaptive interference-aware VNF placement for service-customized 5G network slices[C]. IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 2449-2457.
- [47] 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.
- [48] Luizelli M C, Bays L R, Buriol L S, et al. Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions[C]. 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015: 98-106.
- [49] Huang M, Liang W, Xu Z, et al. Throughput maximization in software-defined networks with consolidated middleboxes[C]. 2016 IEEE 41st conference on local computer networks (LCN). IEEE, 2016: 298-306.
- [50] 祖家琛, 胡谷雨, 严佳洁, 等. 网络功能虚拟化下服务功能链的资源管理研究综述[J]. 计算机研究与发展, 2021, 58(01): 137-152.
- [51] Sun G, Zhou R, Sun J, et al. Energy-efficient provisioning for service function chains to support delay-sensitive applications in network function virtualization[J]. IEEE Internet of Things Journal, 2020, 7(7): 6116-6131.
- [52] Addad R A, Dutra D L C, Taleb T, et al. AI-based network-aware service function chain migration in 5G and beyond networks[J]. IEEE Transactions on Network and Service Management, 2021, 19(1): 472-484.
- [53] Feng H, Shu Z, Taleb T, et al. An aggressive migration strategy for service function chaining in the core cloud[J]. IEEE Transactions on Network and Service Management, 2023, 20(2): 2025-2039.
- [54] Cao H, Wu S, Hu Y, et al. A survey of embedding algorithm for virtual network embedding[J]. China Communications, 2019, 16(12): 1-33.
- [55] Luo F M, Xu T, Lai H, et al. A survey on model-based reinforcement learning[J]. Science China Information Sciences, 2024, 67(2): 121101.
- [56] Bi Y, Meixner C C, Bunyakitanon M, et al. Multi-objective deep reinforcement learning

- assisted service function chains placement[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4134-4150.
- [57] Ye J, Zhang Y J A. DRAG: Deep reinforcement learning based base station activation in heterogeneous networks[J]. IEEE Transactions on Mobile Computing, 2019, 19(9): 2076-2087.
- [58] Orlowski S, Wessäly R, Pióro M, et al. SNDlib 1.0—Survivable network design library[J]. Networks: An International Journal, 2010, 55(3): 276-286.

攻读学位期间的学术论文与研究成果

■ 已发表学术论文

- [1] 王媛滔, 舒兆港, 钟一文, 邱彩钰, 田佳霖. 基于 VNF 实例共享的服务功能链部署算法[J]. 计算机应用研究, 2023, 40(06): 1806-1811. (中文核心期刊, 第一作者, 对应本文第三章内容)
- [2] Y. Wang, Z. Shu, S. Chen, J. Lin, Z. Zhang. A cost and demand sensitive adjustment algorithm for service function chain in data center network[J]. Computer Networks, 2024, 242: 110254. (SCI2 区, IF5.6, 第一作者, 对应本文第四章内容)
- [3] H. Feng, Z. Shu, T. Taleb, Y. Wang, Z. Liu. An Aggressive Migration Strategy for Service Function Chaining in the Core Cloud[J], IEEE Transactions on Network and Service Management, 2023, 20(2): 2025-2039. (SCI2 区, IF5.3, 第四作者)
- [4] H. Feng, Z. Shu, Z. Zhang, Y. Wang, Z. Liu. Bandwidth Resources Prediction for Dynamic Networking Slicing Based on Time Series Analysis[C]. 2022 International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM). IEEE, 2022: 1-5. (EI 会议, 第四作者)
- [5] H. Feng, Z. Shu, Z. Zhang, Y. Wang, Z. Liu. Predict Critical Node Pairs Based on Reinforcement Learning in SDN[C]. 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). IEEE, 2022: 979-982. (EI 会议, 第四作者)

■ 科研项目

- [1] 福建省自然科学基金项目, “基于 SDN/NFV 下 5G 切片资源动态分配和优化”, 2020J01574, 2020.08-2023.08。
- [2] 中国高校产学研创新基金项目, “基于机器学习的跨域 5G 网络切片机制”, 2021FNA05003, 2022.01-2024.12。

致谢

三年的时间转瞬即逝，回顾这段旅程，我的心中充满了感激。

首先，我要特别感谢舒兆港副教授。在老师的严格要求和不懈鞭策下，我收获了丰富的知识和宝贵的经验，每一天都过得十分充实和有意义，也为我提供了优良学习环境和研究资源，使我能够顺利完成学业和研究。

我还要感谢我的家人，包括父母和爷爷奶奶，他们是我坚实的后盾。在我成长的每一个阶段，不管我做出了怎样的选择，都能得到他们无条件的支持和鼓励，他们的爱是我前进的动力。

同时，我也要感谢我的男朋友，他的精神鼓励是我在面对困难时坚持下来的重要力量。在人生的旅程中，有他的陪伴和支持，让我感到温暖和勇敢。

此外，我不能忘记感谢我的学长、同门、师弟师妹们。你们的存在让我的学习生活更加丰富多彩，也为我的研究生生活留下了难忘的记忆。

最后，我也要感谢我自己，感谢自己的坚持和努力，没有放弃，勇敢地面对每一个挑战。

对于未来，我希望自己能够继续不惧困难，勇敢向前，活出每一天的精彩。希望我们都能够拥抱每一个梦想，珍惜每一段旅程，用心感受生活的每一刻。