

空天地网络中异构可靠感知的SFC嵌入与重嵌入研究

秦凯鑫¹⁾ 郭松涛¹⁾ 刘贵燕^{1,2)}

¹⁾(重庆大学计算机学院 重庆 400000)

²⁾(嵌入式系统与服务计算教育部重点实验室(同济大学) 上海 201804)

摘要 随着5G网络、6G网络的快速发展,在空天地一体化网络(Space-air-ground Integrated Network, SAGIN)中提供可靠服务需要合适的服务功能链(Service Function Chain, SFC)嵌入方案以保证服务质量。然而,SAGIN复杂的拓扑结构、有限的基础设施资源以及异构的硬件和软件等特征给现有可靠感知的SFC嵌入方案提供高质量服务带来一定挑战。采用SFC备份是个可行的解决方案。然而备份位置的选择对于立体的SAGIN拓扑和不确定的用户需求是一个较大的问题。针对上述问题,本文研究了SAGIN中异构可靠感知SFC嵌入和重嵌入问题(Heterogeneous Reliable SFC Embedding and Re-embedding, HRSER)。本文首先将问题建模为非线性整数规划问题,并证明它是NP难的。为了解决这个问题,本文提出了一种包括两个子算法的混合方案。其中两个子算法分别为可靠性感知的SFC嵌入(Reliability-aware SFC Embedding, RASE)和基于贪婪策略的重嵌入(Greedy SFC Re-embedding, GSR)。首先,RASE将利用深度强化学习的自主决策能力,计算在SAGIN中SFC嵌入和备份的最优解。在网络拓扑中会有相应的执行设备网络拓扑中会有相应的执行设备进行备份以提高服务的可靠性。随后,当网络负载超过预定义阈值时,GSR算法执行SFC重新嵌入以提高可靠性和SFC的接受率。此外,从理论上证明了重新嵌入所需的SFC备份数量的下限。最后,本文对RASE和GSR算法进行了算法复杂性分析。基于真实数据集的模拟结果表明,RASE方案在隐含层神经元数量不同的情况下都能收敛到一个合理的奖励。与现有技术相比,所提出的混合算法可以将SFC接受率提高接近82%,而额外付出的成本不超过5%。

关键词 服务功能链嵌入;可靠性;异构性;重嵌入;深度强化学习;空天地一体化网络

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2025.01976

Heterogeneous Reliable-Aware SFC Embedding and Re-Embedding in SAGIN

QIN Kai-Xin¹⁾ GUO Song-Tao¹⁾ LIU Gui-Yan^{1,2)}

¹⁾(College of Computer Science, Chongqing University, Chongqing 400000)

²⁾(Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai 201804)

Abstract With the rapid development of 5G and 6G networks, providing reliable services in the Space-air-ground Integrated Network (SAGIN) requires a suitable Service Function Chain (SFC) embedding scheme to ensure service quality. However, the complex topology, limited infrastructure resources, and heterogeneous hardware and software of SAGIN bring certain challenges to the existing reliable-aware SFC embedding schemes in providing high-quality services. Using SFC backup is a feasible solution. However, the backup location selection is a big

收稿日期:2024-12-20;在线发布日期:2025-05-07。本课题得到国家自然科学基金面上项目(62272069)、国家自然科学基金青年科学基金(62202073)、同济大学嵌入式系统与服务计算教育部重点实验室开放课题(ESSCKF2024-06)资助。秦凯鑫,硕士研究生,主要研究领域为网络功能虚拟化、边缘计算。E-mail: kxxxy_888@163.com。郭松涛(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为无线自组织网络与移动边缘计算。E-mail: songtao_guo@163.com。刘贵燕,博士,副研究员,主要研究领域为软件定义网络、数据中心网络、网络功能虚拟化与5G网络。

problem for the three-dimensional SAGIN topology and uncertain user needs. To address the above problems, this paper studies the problem of heterogeneous reliable SFC embedding and re-embedding (HRSER) in SAGIN. This paper first models the problem as a nonlinear integer programming problem and proves that it is NP-hard. To solve this problem, this paper proposes a hybrid scheme consisting of two sub-algorithms. The two sub-algorithms are Reliable-aware SFC Embedding (RASE) and Greedy SFC Re-embedding (GSR). First, RASE will use the autonomous decision-making ability of deep reinforcement learning to calculate the optimal solution for SFC embedding and backup in SAGIN. In the network topology, there will be corresponding execution devices for backup to improve the reliability of the service. Subsequently, when the network load exceeds the predefined threshold, the GSR algorithm performs SFC re-embedding to improve reliability and SFC acceptance rate. In addition, the lower bound of the number of SFC backups required for re-embedding is theoretically proved. Finally, this paper analyzes the algorithmic complexity of RASE and GSR algorithms. Simulation results based on real data sets show that the RASE scheme can converge to a reasonable reward with different numbers of hidden layer neurons. Compared with the existing technology, the proposed hybrid algorithm can increase the SFC acceptance rate by nearly 82%, while the additional cost does not exceed 5%.

Keywords service function chain embedding; reliability; heterogeneity; re-embedding; deep reinforcement learning; space-air-ground integrated network

1 引言

社会信息化发展持续推动5G网络、6G网络快速演进,相比上一代通信网络,6G网络有望支持更加复杂多样的应用场景,未来通信网络将呈现多维度的互联互通。但目前地面与空中网络缺乏协同机制,覆盖范围有限,因此6G网络需要突破地形环境限制,构建天空地一体化网络(Space-Air-Ground Integrated Network, SAGIN)^[1-2]。SAGIN是融合地面网络、天基网络和空间不同轨道、不同种类和不同性能的卫星网络的综合网络架构,它为大范围空间内的各类网络服务提供无处不在、智能、协同、高效的信息保障基础设施^[3-4]。

传统网络采用硬件与软件紧密耦合的架构,这种模式导致网络功能的部署成本高昂、管理复杂且故障率较高。网络功能虚拟化(Network Function Virtualization, NFV)这一概念提出后凭借其解决传统硬件网络固化和普适性不强问题上的显著优势,迅速引起学术界和工业界的广泛关注。NFV的核心目标是通过软件实现网络功能,使其能够运行在标准的服务器等硬件上,无需依赖专用硬件。借助NFV,网络服务提供商可以根据需求灵活地将网络功能动态部署到网络的不同位置^[1,5-6]。在NFV环境中,业务功能通常由一系列虚拟网络功能

(Virtual Network Function, VNF)组成链式结构,这种结构被称为服务功能链(Service Function Chain, SFC)。SFC必须嵌入到相关的物理设备中才能有效运行,如何有效进行SFC的嵌入以提供高效服务的问题称为SFC嵌入问题^[7-8]。

在SAGIN中提供高可靠的低延迟服务需要持续保障网络内部功能和差异化服务。例如,在车联网中,需要超可靠的低延迟通信来确保车辆安全。农作物监测需要高带宽来确保服务覆盖率和稳定性^[9]。许多研究机构和组织正在将SAGIN纳入其项目或工业生产工作中,例如全球信息网格^[10-11]。尽管基于硬件的网络基础设施能够实现目前多数功能和需求,但由于其僵化和固定的拓扑结构,阻碍了对动态网络环境的响应,故不适用于SAGIN。NFV技术的出现为网络功能的部署和迁移提供了一种新方案。使用虚拟化技术及SFC嵌入,网络服务提供商可以快速灵活地部署网络功能,提高SAGIN中的服务效率和服务质量。

在真实SAGIN网络拓扑中,物理设备和资源随时间快速变化,静态的SFC嵌入策略可能无法满足SAGIN的高移动性和网络快速变化的特点。因此,需要考虑重新嵌入或调度服务功能链来提高SAGIN中的服务质量。此外,服务可靠性也需要尽可能提高,以保障SAGIN的服务质量。服务可靠性与VNF可靠性、SFC可靠性以及物理设备的可靠

性密切相关,因此提升 VNF 可靠性和 SFC 可靠性十分必要。目前,大多数提升服务可靠性的有效方法是进行虚拟网络功能备份。目前相关研究工作提出了多种解决方案,包括在同一物理设备下进行多个 VNF 的备份部署和进行多个跨物理设备部署同一 VNF 的多个备份^[12-14]。

目前,SFC 嵌入方面的研究多集中在数据中心和边缘环境,SAGIN 方面的研究较少,尤其是 SFC 可靠性方面的研究^[12,14-16]。尽管在边缘环境和数据中心已经进行了大量关于 SFC 可靠性的研究,但由于高移动性、复杂的立体拓扑、设备和资源异构性的特点,这些工作的解决方案无法直接应用于 SAGIN^[16-18]。为了应对这些特点,深度强化学习(Deep Reinforcement Learning, DRL)因其自主决策和自适应选择等特点在解决 SFC 嵌入和重新嵌入问题时能提供较大帮助。深度强化学习中的智能体进行一种决策即一种嵌入方案每一次轮训练中接受环境的特征状态向量,进行一种决策即一种嵌入方案,在将方案作用在环境中后,获取环境的反馈即奖励,最终判断嵌入方案是否合理并进行提升^[14,19-21]。它可以通过考虑在特定环境(如 SAGIN)内采取的每个动作相关的奖励来评估 SFC 嵌入动作的质量,并生成随机值或通过神经网络优化嵌入动作以最终收敛。然而,在 SAGIN 拓扑下,现有的关于网络功能虚拟化环境中可靠的 SFC 嵌入和重新嵌入的研究仍然面临如下巨大挑战:

(1)SAGIN 中设备异构且分布在不同维度,导致拓扑结构复杂,不同层次设备的连接情况和网络状态也与地面拓扑不同,影响服务的可靠性和时延。

(2)SAGIN 基础设施部分资源有限,在同一个基础设施中嵌入过多 VNF 备份会导致网络服务不可用,从而导致服务质量急剧下降。

(3)网络中用户需求的动态性和不确定性使得 SAGIN 难以持续提供可靠的服务。

为了解决上述挑战,本文以计算资源、内存和带宽为代价,对可靠的 SFC 嵌入和重新嵌入问题进行了建模。此外,提出了一种结合可靠性感知的 SFC 嵌入(Reliable-aware SFC Embedding, RASE)和贪婪策略重嵌入(Greedy SFC Re-embedding, GSR)的混合算法。该算法可以以较低的成本有效地提高接受率。本文的主要贡献如下:

(1) 将异构可靠的 SFC 嵌入和重嵌入(Heterogeneous Reliable SFC Embedding and Re-embedding, HRSER)问题形式化为非线性整数规划,

通过考虑 SAGIN 的特征与 SFC 的可靠性对此问题进行建模,尽可能提升可靠性的同时,节约部署成本,提升资源利用率。

(2) 提出了一种混合算法来解决 HRSER 问题。该算法采用时间复杂度较低的贪婪策略来解决不确定的用户请求。随后,从理论上证明了当网络到达瓶颈时,SFC 重新嵌入所需的最少备份数。

(3) 对算法的复杂性和不可靠的环境情况进行分析。最后基于真实的数据集模拟了所提算法的性能。与已有工作相比,SFC 接受率提高了接近 82%,额外成本不超过 5%。

2 相关工作

本节将简要总结已有的相关工作,包括 SFC 嵌入和 SFC 重嵌入两个方面。

2.1 SFC 嵌入问题

目前已有工作在移动边缘计算(Mobile Edge Computing, MEC)和数据中心网络的背景下考虑 SFC 的嵌入。然而,这些方法并没有完全考虑到 SAGIN 独特的环境特征。例如,Solozabal 等^[22]在解决 VNF 前向图嵌入问题时重点关注单个 VNF 的放置,但忽略了 SFC 内 VNF 的顺序。Ya 等人^[23]引入了两个优化函数:一个针对 CPU 和带宽要求,另一个关注接受率。他们的贡献主要是在 DRL 框架内应用图卷积神经网络,但是对于网络拓扑,GCN 显然无法合理地关注到链接的特点。此外,在部署 SFC 时,Chen 等人^[24]考虑了用户的移动性,作者 Jin 等人^[25]考虑了 MEC 的低延迟要求场景。为了解决优化问题,此文章提出了两种延迟感知的 VNF 部署算法,以在资源约束内嵌入和迁移 SFC。当前大量研究的目标是在边缘环境中最大化资源利用率或最小化成本^[26]。除了上述优化目标外,可靠性也是 5G 网络性能的重要指标和关键衡量标准^[18],它是 SFC 成功嵌入一种表示,能够提升用户体验。

2.2 SFC 调度和重嵌入

目前一部分研究工作重点在 SFC 的调度和重嵌入。在动态环境中,每当有流到达并满足设置的触发条件时,VNF 就会被重新嵌入。一部分工作提出使用 DRL 来实现重新嵌入,通过部署多个备份提高可靠性^[27],然而 DRL 涉及的重新训练和神经网络推理可能会消耗一定的时间。为了实现更优的负载均衡,Tu 等人^[28]考虑在多租户云中频繁更新 SFC 路由。然而,所考虑的环境与 SAGIN 环境存在区

别,无法适应SAGIN,频繁的更新也会导致降低系统性能。此外,Jia^[18]和Yang^[27]等人也考虑了动态环境下的SFC调度问题,他们通过合并同类型的备份来提高可靠性。然而,由于SFC的随机到达,早期嵌入方案可能不适合下一个SFC,因此重新嵌入是必须的。然而,与Yang等人的工作^[27]类似,DRL方法可能会造成相对较高的延迟。

综上所述,近期解决SFC可靠性问题的工作主要集中在数据中心和边缘服务器上,而考虑到SAGIN特性的SFC嵌入研究则较少。如图1所示是一个空天地一体化网络中SFC嵌入的例子,SFC中的VNF可以嵌入到SAGIN的不同层中,然而,跨层嵌入VNF会导致服务稳定性降低和传输延迟增加,尤其是从地面到天空和卫星等。因此,与上述研究不同,本文提出了一种旨在增强SAGIN中SFC可靠性的混合算法,包括两个算法:RASE和GSR。首先,RASE利用DRL来处理SAGIN的特点,进行备份选择和部署,GSR以经过理论证明的最小备份数量进行重新嵌入达到保障可靠性的基础上最小化成本。该混合算法在保证SFC接受率的同时,最小化部署成本和所需资源。

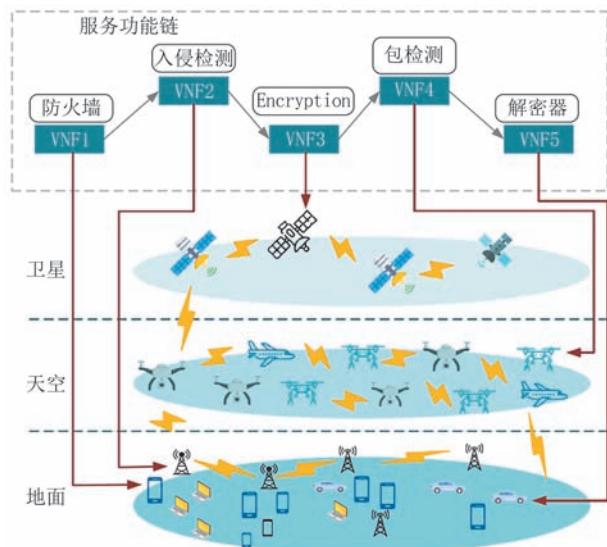


图1 SAGIN下进行SFC嵌入

3 系统模型和问题形式化定义

本节具体描述了系统模型,并形式化构建了异构可靠感知SFC嵌入和重嵌入问题。相关符号列于表1中。

表1 部分模型符号说明

变量	说明
$\varpi_{f_i}, \varrho_{f_i}$	f_i 需要的计算资源和内存资源
$\zeta_{f_i, f_{i+1}}$	虚拟链接 f_i, f_{i+1} 需要的带宽
η_u, v_u	单位计算资源和内存的经济代价
$\varsigma_{u, v}$	单位带宽需要的经济代价
$\phi^{f_i, s}$	表示 f_i 的可靠性
ν_u	设备 u 的可靠性
\mathfrak{R}^s	s 可容忍的最低可靠性

3.1 系统模型

3.1.1 SAGIN网络拓扑

图1描绘了SAGIN的拓扑结构,其中设备包括三层:卫星层,空中层和地面层。每一层都有各类异构设备。如地面层包括基站,车辆和手机。空中可能存在无人机,卫星层涉及各种轨道的卫星。该拓扑可以用无向图 $\mathcal{G}^P = \{\mathcal{N}^P, \mathcal{L}^P, \mathbf{A}^P\}$ 表示,其中 \mathcal{N}^P 表示物理设备集合, \mathcal{L}^P 表示这些设备之间的链接。由于SAGIN的构建目标及其特性,物理链接存在于不同层之间以及同一层内,因此令 $\mathcal{L}^P = \{\mathcal{L}_S^P \cup \mathcal{L}_A^P \cup \mathcal{L}_G^P \cup \mathcal{L}_{S,A}^P \cup \mathcal{L}_{S,G}^P \cup \mathcal{L}_{A,G}^P\}$ 。此处 \mathcal{L}_S^P 表示不同轨道卫星设备之间的链接, \mathcal{L}_A^P 代表空中设备(例如无人机等)之间的链接,而 $\mathcal{L}_{S,G}^P$ 是地面设备和轨道卫星设备之间的链接。

参数 $\mathbf{A}^P = \{C_u^P, M_u^P, D_u^P, B_{u,v}^P, D_{u,v}^P, \nu_u\}$ 包含了SAGIN拓扑的各种资源参数和属性参数。其中, C_u^P 为SAGIN中网络节点的计算资源量, M_u^P 为节点的内存大小, D_u^P 表示节点对任务的处理延迟, $B_{u,v}^P$ 为链接 $l_{u,v}$ 的带宽大小, $D_{u,v}^P$ 为 $l_{u,v}$ 的传输延迟。与SAGIN中的传输延迟相比,SFC的传播延迟和排队延迟可以忽略不计,因此本文在后续计算时忽略。最后, ν_u 表示网络设备 u 的可靠性,即设备能正常运行,不出现宕机等状况的概率。因此, $1 - \nu_u$ 即可表示设备提供服务失败或终止的概率。

3.1.2 SFC请求定义

SFC请求是用户发送的一系列请求流,该请求流依次通过每个需要的VNF。包含了 k 个SFC的SFC请求集合表示为 $\mathcal{S} = \{s_1, s_2, s_3, \dots, s_k\}$, 其中每个SFC $s \in \mathcal{S}$ 可以表示为 $s = \{f_1, f_2, f_3, \dots, f_n\}$, 其中 f_i 表示SFC中的第 i 个VNF。令 $\mathbf{A}^V = \{\varpi_{f_i}, \varrho_{f_i}, \zeta_{f_i, f_{i+1}}, \phi^{f_i, s}, D_s\}$ 。其中, ϖ_{f_i} 和 ϱ_{f_i} 分别表示VNF f_i 所需的计算资源和内存。 $\zeta_{f_i, f_{i+1}}$ 表示VNF f_i 和 f_{i+1} 之间需要的链路带宽。 $\phi^{f_i, s}$ 是VNF f_i 的可靠性,则 $1 - \phi^{f_i, s}$ 表示

当前 VNF 发生故障并停止服务的概率。最后, D_s 表示当前 SFC 可以容忍的最大延迟。

3.1.3 SFC 嵌入的备份

现有的解决 SFC 可靠性问题的相关工作大致可以分为两类,如图 2 所示,嵌入方案可以分为将一种 VNF 的备份放在不同的物理设备上和放在同一设备上。目前,大多数工作认为将备份放在同一设备上会存在设备故障时整个服务不可用的问题,即当服务器或者基站出现故障关机,此方案会导致无论有多少备份,系统都将无法运行,未达到优化系统可靠性的目的。因此,所提出的模型考虑物理设备故障和软件漏洞及异常,从而采用分布式备份方法应对故障。

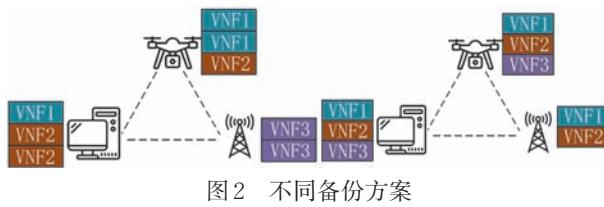


图 2 不同备份方案

3.1.4 SFC 重嵌入问题

为了保证 SFC 的可靠性,模型通常在前期将大量的备份部署到网络拓扑中。然而,这种方式会造成网络负载过重。当网络负载过重,设备资源使用率达到阈值 Ψ 时,网络的整体性能和 SFC 接受率会受到很大影响,从而影响服务体验和服务质量,其中 Ψ 为网络系统所规定的最大负载。在满足网络负载触发条件时,会考虑进行 SFC 重嵌入。触发条件其定义为式(1):

$$\max \left\{ \frac{\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varpi_{f_i}}{C_u^P}, \frac{\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varrho_{f_i}}{M_u^P} \right\} \geq \Psi \quad (1)$$

单个设备负载是对整个网络的负载的一种反映,如果一个网络中内存或者计算资源的最大使用量超过阈值,意味着这个网络即将进入重度负载状态。因为这个节点的负载过大,后续的一系列请求可能会被拒绝。从而,不等式(1)表达出了网络性能下降的一个趋势和信号,因此,本文认为,当满足不等式(1)时,后续的 SFC 就会阻塞,需要重新嵌入 SFC 触发贪婪策略 SFC 重嵌入 GSR 算法。

3.2 问题形式化

部署 SFC 需要消耗各种资源,包括计算资源、内存资源和带宽^[14,29-32]。当互联网服务提供商考虑在其物理设备上部署 SFC 时,其一般关注的关键因素之一是嵌入所需要的经济代价。因此,本模型主

要关注经济成本,包括计算成本、内存成本和带宽成本。总成本由所需资源量乘以相应的单价得出,如下式所示:

$$\begin{aligned} \text{Cost}(s) = & \sum_{f_i \in s} \eta_u \varpi_{f_i} x_u^{f_i, s} + \sum_{f_i \in s} \varrho_u \varrho_{f_i} x_u^{f_i, s} + \\ & \sum_{l_{f_i, f_{i+1}} \in s} \varsigma_{u, v} \zeta_{f_i, f_{i+1}} h(l_{f_i, f_{i+1}}) x_{f_i, f_{i+1}}^{u, v}, \\ \forall u, v \in \mathcal{N}^P, (u, v) \in \mathcal{L}^P, s \in \mathcal{S} \end{aligned} \quad (2)$$

其中 $x_u^{f_i, s}$ 和 $x_{f_i, f_{i+1}}^{u, v}$ 是 0-1 变量,分别表示 VNF f_i 是否放置在物理节点 u 上以及虚拟链路 $l_{f_i, f_{i+1}}$ 是否映射到物理链路 $l_{u, v}$ 上。而变量 $h(l_{f_i, f_{i+1}})$ 表示链路 $l_{f_i, f_{i+1}}$ 的跳数。其中,相邻设备之间的跳数 $h(l_{f_i, f_{i+1}}) = 1$ 。由于请求不只是经过一个设备和物理链路,因此,请求要经过的所有链路都需要提供当前请求所需的带宽,带宽成本为 $\varsigma_{(u, v)} \zeta_{f_i, f_{i+1}} h(l_{f_i, f_{i+1}})$ 。

(1) 备份约束:每个 VNF 在物理设备上部署至少一个备份以保证 SFC 可用,并且模型采用分布式备份,使得每台物理设备上只能存在一个同类型的 VNF 备份,就有如下约束:

$$\sum_{u \in \mathcal{N}^P} x_u^{f_i, s} \geq 1, \forall s \in \mathcal{S}, f_i \in s \quad (3)$$

(2) 可靠性约束:假设每个 SFC 都有一个最小容忍可靠性阈值,则模型应确保 SFC 的整体可靠性超过该最小阈值。已知,SFC 的可靠性由组成它的每个 VNF 的可靠性以及设备可靠性决定。则约束可以表示为:

$$R^s = \prod_{f_i \in s} R_{f_i}^{s, s} \geq \mathfrak{R}^s, \forall s \in \mathcal{S} \quad (4)$$

其中 R^s 和 $R_{f_i}^{s, s}$ 分别表示 SFCs 和 VNF f_i 的可靠性。

另外,每个 VNF 的可靠性取决于它所有备份中是否至少有一个可用,而备份可用意味着软件和硬件都可用,即前文所提到的软件可靠性和硬件可靠性的概率要命中。因此,VNF 和备份的可靠性定义可以用以下等式来表达:

$$R_{f_i}^{s, s} = 1 - \prod_{u \in \mathcal{N}^P} (1 - R_u^{f_i, s}), \forall s \in \mathcal{S}, f_i \in s \quad (5)$$

$$R_u^{f_i, s} = \nu_u \phi^{f_i, s} x_u^{f_i, s}, \forall s \in \mathcal{S}, f_i \in s, u \in \mathcal{N}^P \quad (6)$$

(3) 资源约束:在 SAGIN 中,设备的带宽、内存和计算资源是有限的,因此资源容量约束可以表示为:

$$\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varpi_{f_i} x_u^{f_i, s} \leq C_u^P \quad (7)$$

$$\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varrho_{f_i} x_u^{f_i, s} \leq M_u^P \quad (8)$$

$$\sum_{s \in \mathcal{S}} \sum_{l_{f_i, f_{i+1}} \in s} \zeta_{f_i, f_{i+1}} x_{f_i, f_{i+1}}^{u, v} \leq B_{l_{u, v}}^P \quad (9)$$

(4) 延迟约束: SFC 的服务质量中需要重点考虑的一个约束就是服务响应延迟。因此, SFC 的处理延迟和传输延迟之和小于最大可接受延迟 D_s , 其表达式为:

$$\sum_{f_i \in s} D_u^P x_{f_i, s}^{f_i, s} + \sum_{l_{f_i, f_{i+1}} \in s} D_{u, v}^P x_{f_i, f_{i+1}}^{u, v} \leq D_s, \quad \forall s \in \mathcal{S}, u \in \mathcal{N}^P, l_{u, v} \in \mathcal{L}^P \quad (10)$$

SAGIN 的复杂拓扑结构导致不同层的物理连接延迟不同, 因此考虑到 SAGIN 的特征给出具有以下约束。其中, 太空卫星间的连接的最大延迟最高, 其次是空中, 地面上的设备延迟被认为是最低。最小延迟同理。

$$\max D_{\mathcal{L}_G^P}^P \leq \max D_{\mathcal{L}_A^P}^P \leq \max D_{\mathcal{L}_S^P}^P \quad (11)$$

$$\min D_{\mathcal{L}_G^P}^P \leq \min D_{\mathcal{L}_A^P}^P \leq \min D_{\mathcal{L}_S^P}^P \quad (12)$$

基于以上考虑, HRSER 问题可以形式化如下:

$$\begin{aligned} & \min \sum_{s \in \mathcal{S}} \text{Cost}(s) \\ & \text{s.t. (3) - (12)} \\ & x_{f_i, s}^{f_i, s} \in \{0, 1\}, x_{f_i, f_{i+1}}^{u, v} \in \{0, 1\} \end{aligned} \quad (13)$$

在定理 1 中描述了问题 HRSER 是 NP 难的并给出证明。

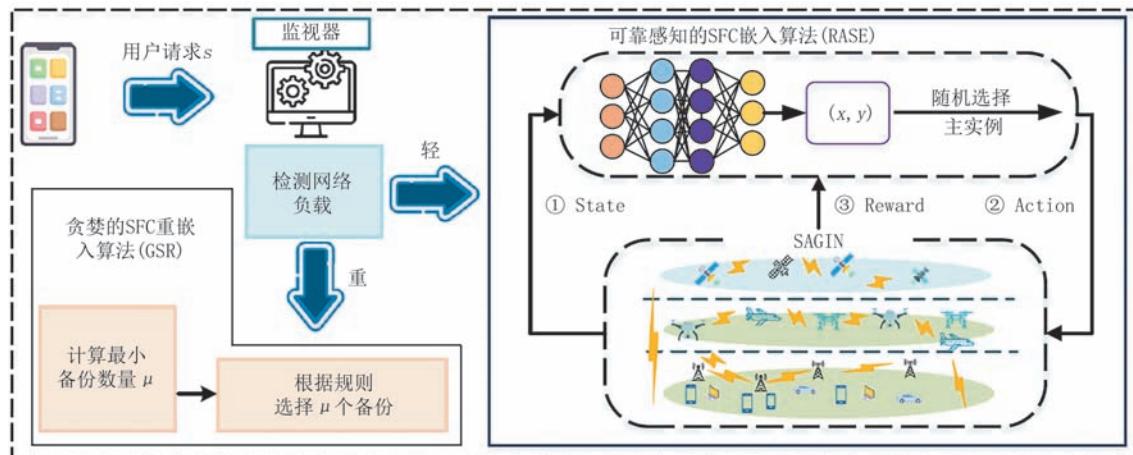


图3 混合算法框架

RASE 和 GSR 的核心思想是使用 DRL 和贪婪策略来解决问题并获得 SAGIN 中的 SFC 嵌入和重嵌入解决方案。两个算法相对独立, 仅由监视器在初期进行网络状态检测以决策使用哪种方案, 两个算法使用相同的网络状态参数, 数据相同, 仅是在部署时选择的两种方案。在 SAGIN 中进行 SFC 嵌入不仅要考虑不同物理设备之间的异构参数变化, 还要考虑

4 算法详细设计

由于该问题是 NP-hard, 因此采用近似算法来获得近似最优解是可行的一种方案。本节提出了一种混合算法来解决 HRSER 问题。介绍整个混合算法的框架和思路, 该算法集成了 RASE 和 GSR 算法。

4.1 算法框架

首先进行该框架的概述, 如图 3 所示。网络拓扑中存在一个监视器, 用于检测网络负载是否超过阈值, 并在 SFC 请求进入时确定是否需要重新嵌入。如果网络负载超过阈值, 则将重新嵌入网络中的达到设备瓶颈的 SFC。否则, 将直接通过 RASE 算法嵌入。

在 RASE 中, 系统可以任意选择多个备份作为服务的主实例。然而, 考虑到故障后将一个实例切换到另一个备份需要成本, 且由于设备与地面服务器之间的延迟低、服务器资源规模大, 地面服务器等设备节点在被选择为主实例时具有优先权。随后, 在 GSR 中, 本文证明了满足约束的备份数量下限, 从而使模型可以选择最少数量的备份进行贪婪策略地重新嵌入。

这些设备的空间分布。因此, 使用基于 DRL 和贪婪策略的方法, 通过设置有效的奖励函数并设计合理的动作空间和状态空间来有效地解决这个问题。

4.2 SFC 嵌入算法

设计一个良好的 DRL 智能体能够捕捉问题所关注的特征和复杂的网络拓扑结构, 同时能减少频繁违反约束导致的试错成本。双重深度 Q 网络

(Double Deep Q-learning, DDQN) 是 DRL 处理离散动作空间的一种方案, 通过经验重放缓冲区从先前的经验和随机生成的动作和奖励中学习。

A3C 采用异步并行的方式, 通过多个智能体在不同环境的副本中学习更新相互协作以优化策略。DDPG 等算法则基于确定性策略梯度, 适用于连续动作空间, 通过 Actor-Critic 结构, 利用经验回放和目标网络稳定学习过程。DQN 是处理离散动作空间的更合适的算法, 特别是它的两个神经网络可以相互协作, 减少高估问题, 提高学习稳定性。其他 DRL 算法如 A3C 和 DDPG 则更适合处理连续动作。对于 SFC 嵌入问题, 每个动作都是离散的, 且在服务功能链嵌入问题中, A3C 采用的异步并行在多个智能体之间的参数冲突可能会导致收敛波动, 而 DDPG 则对离散动作的处理能力略差, 因此这两个算法显然不适用。

在 HRSER 问题中, DDQN 可以将 VNF 所有备份的嵌入位置视为一个动作, 同时通过设置和成本有关的奖励, 以在训练时最小化成本。在 DDQN 中, 首先使用三元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 来表示状态空间, 动作空间和奖励函数。接下来, 分别描述这三个元素。

(1) 状态空间。为了正确表示系统状态, 状态张量应该包括 SAGIN 中的 SFC 和物理设备所包含的参数。由于本文考虑到 SFC 的可靠性, 状态张量包含 SFC 可以容忍的最小可靠性 \mathfrak{R}^s 、相关硬件可靠性以及物理节点提供的资源。因此, \mathcal{S} 表示如下:

$$\mathcal{S} = \{ \mathfrak{R}^s, C_u^P, M_u^P, D_u^P, B_{u,v}^P, v_u, \eta_u, \nu_u, \zeta_{u,v}, \forall s \in \mathcal{S}, f_i \in s, u \in \mathcal{N}^P, l \in \mathcal{L}^P \} \quad (14)$$

经计算, 状态张量的长度 $|\mathcal{S}| = 1 + 6|\mathcal{N}^P| + 2|\mathcal{L}^P|$, 这个长度会对 DDQN 的计算复杂度产生影响, 在本文后续会进行分析。

(2) 动作空间。DDQN 的智能体采取的动作是选择 VNF 备份部署位置和当前 VNF 备份是否是主实例。此操作需要分为两个部分, 一个是 VNF 放置动作, 另一个是备份后选择主实例的动作。动作空间定义如下:

$$\mathcal{A} = \{ x_u^{f_i,s}, y_u^{f_i,s}, \forall s \in \mathcal{S}, f_i \in s, u \in \mathcal{N}^P \} \quad (15)$$

其中, $y_u^{f_i,s}$ 表示部署在设备 u 上的 SFC s 中的 VNF f_i 的实例是否是主实例。同理, 可以得到动作张量的长度 $|\mathcal{A}| = 2|\mathcal{N}^P|$ 。

(3) 奖励函数。智能体从环境中获得的奖励可以激励其做出更准确的决策。该模型的目标是在遵守约束的前提下最小化经济成本。奖励是促使智能体

在执行放置动作后环境提供的相应反馈。因此, 奖励通过成本的负数表达, 奖励函数为:

$$\mathbb{R} = -\alpha \text{Cost}(s) - \beta \sum_{f_{i+1} \in s} \sum_{f_i \in s} \sum_{u \in \mathcal{N}^P} \sum_{u' \in \mathcal{N}^P} D_{(u',u)}^P x_u^{f_{i+1},s} x_{u'}^{f_i,s} \quad (16)$$

其中 $\alpha \geq 0, \beta \geq 0$ 且都为常数。式(16)中添加了包含 VNF 备份的两个设备之间的延迟, 以确保所选备份更接近地面。如果一条完整的 SFC 包含了三层网络的不同部分设备, 智能体可以借助延迟平衡项进行决策, 选择更低延迟的备份方案。换言之, 由于部署了多个备份, 如果 SFC 经过高延迟的 VNF 实例从而导致 SFC 违反延迟约束, 则会倾向于选择低延迟的备份, 从而保证最小化端到端延迟。通过训练, 所选备份之间的延迟将更低, 且智能体将更倾向于选择成本较低的操作最终实现收敛。

算法 1 输入 SFC 的软硬件参数作为 DDQN 中智能体接受的初始状态张量。算法第 2-16 行执行 DDQN 的训练步骤, 其中 θ 表示神经网络的参数。经过 ϵ -贪婪策略后生成的最佳动作需要通过第 6-8 行进行筛选, 初始选择的主实例是随机的。最后, 如果得到的备份不满足约束条件, 拒绝 SFC。

算法 1. 可靠性感知的 SFC 嵌入算法.

输入: 最低可容忍可靠性 \mathfrak{R}^s , 网络拓扑参数 $\mathcal{A}^P = \{C_u^P, M_u^P, D_u^P, B_{u,v}^P, D_{u,v}^P, \nu_u\}$, SFC 参数 $\mathcal{A}^V = \{\varpi_{f_i}, \varrho_{f_i}, \zeta_{f_i, f_{i+1}}, \phi^{f_i, s}, D_s\}$, 单位资源的代价 $v_u, \eta_u, \zeta_{u,v}$

输出: 嵌入动作 $a = (x_u^{f_i,s}, y_u^{f_i,s})$

1. 初始化回放集 \mathcal{B}
 2. FOR each episode DO
 3. 初始化状态 s_t
 4. FOR $s \in \mathcal{S}$ DO
 5. FOR $f_i \in s$ DO
 6. 根据 ϵ -贪婪规则选择动作 a
 7. 随机选择 $y_u^{f_i,s} = 1$
 8. $y_u^{f_i,s} = 0, \forall u' \neq u$
 9. 执行动作并获取奖励
 10. 将经验 (s_t, a, r, s_t') 放入 \mathcal{B}
 11. 更新网络参数 θ
 12. $s_t = s_t'$
 13. END FOR
 14. END FOR
 15. END FOR
 16. IF $R^s < \mathfrak{R}^s$ THEN
 17. 拒绝 SFC s
 18. END IF
- 返回: a

4.3 SFC 重嵌入算法

当SFC请求到达时, 监控器首先获取整个网络的当前状态。监控器根据网络的状态判断是否达到重嵌入阈值即满足式(1)。由于监控器此时会阻塞流, 因此它需要以尽可能快的速度解决重新嵌入问题。由于贪婪算法的算法时间复杂度相比于进行推理的RASE算法更快, 监控器应使用速度更快的贪婪算法, 而不是让训练好的智能体进行重新嵌入, 以最小化重新嵌入时间并降低整体嵌入成本。贪婪策略GSR如算法2所示。由于当网络负载达到瓶颈时会触发重新嵌入, 因此应在重新嵌入后最小化备份数量以避免由于使用RASE训练的智能体而嵌入到同一位置的情况。令 μ 为最小备份数, 可以得到定理2, 其中 \mathfrak{R} 表示SFC s 的最小可容忍可靠性。

贪婪策略GSR需要聚合同一设备的归一化CPU和内存资源总数。监视器选择前 μ 个设备进行备份, 并选择资源最大的设备作为主实例。GSR的细节在算法2中描述。SAGIN中设备的负载状态首先由监视器在SFC请求流到达时获取。第2-3行计算资源利用率并检查是否超过阈值。如果达到阈值, 则阻塞后续的SFC, 并按照第5-13行执行重新嵌入。将符合条件的设备放入一个集合中, 然后在第9行进行备份选择。

算法2. 贪婪的SFC重嵌入算法

```

输入: SFC  $s$ 
输出: 嵌入动作  $a = (x_u^{f_i, s}, y_u^{f_i, s})$ 
1. 获取网络的状态
2. 计算  $\left(\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varpi_{f_i}\right) / C_u^P, \left(\sum_{s \in \mathcal{S}} \sum_{f_i \in s} \varrho_{f_i}\right) / M_u^P$ 
3. IF 存在设备满足式(1)THEN
4. 阻塞后续SFC
5. 所有满足条件设备构成集合B
6. FOR  $v \in B$  DO
7. 归一化每个设备的  $C_u^P, M_u^P$ 
8. 设备总资源  $\Upsilon_u = C_u^P + M_u^P$ 
9. 根据top-k算法选出  $\mu$  个设备备份
10. 更新设备资源量
11. END FOR
12. END IF
返回:  $a$ 

```

4.4 算法分析

RASE算法采用离线训练的方式进行训练, 训练完成后通过训练好的模型进行在线推理部署, 为新到达的SFC部署备份并选择主实例。离线训练的计算复杂度与训练数据的大小和训练周期成正

比。由于训练过程是离线的, 本文重点分析在线运行时的计算复杂度。假设神经网络的隐层数为 \mathcal{X} , 每个隐含层包含 \mathcal{Y} 个神经元。

定理3. 在HRSER问题中, 在线过程的运行时间为:

$$\mathcal{O}(|\mathcal{N}^P| + \max \{ \mathcal{Y}(|\mathcal{N}^P| + |\mathcal{L}^P| + \mathcal{X}\mathcal{Y}), |\mathcal{B}|(\mu + |\mathcal{N}^P| \log \mu) \})$$

证明. 算法1描述了RASE的训练过程。根据图3, 当新的SFC到来时, 它会根据网络负载计算选择执行算法1或算法2。因此, 模型的最坏计算复杂度应该是两个算法中较大的一个。首先, 网络负载需要监视器扫描整个网络的设备并计算得到, 其运行时间为 $\mathcal{O}(|\mathcal{N}^P|)$ 。

其次, 算法1的复杂度取决于神经网络的推理过程。长度为 $1 + 6|\mathcal{N}^P| + 2|\mathcal{L}^P|$ 的状态张量作为神经网络的输入层进入。而动作由大小为 $2|\mathcal{N}^P|$ 的张量表示。因此该过程的运行时间为 $\mathcal{O}(\mathcal{Y}(|\mathcal{N}^P| + |\mathcal{L}^P| + \mathcal{X}\mathcal{Y}))$ 。同时, 设满足算法2条件的设备数量为 $|\mathcal{B}|$, 则需要执行 $|\mathcal{B}|$ 次top- k 算法。因此, 算法2的计算复杂度为 $\mathcal{O}(|\mathcal{B}|(\mu + |\mathcal{N}^P| \log \mu))$ 。综上, 模型复杂度为

$$\mathcal{O}(|\mathcal{N}^P| + \max \{ \mathcal{Y}(|\mathcal{N}^P| + |\mathcal{L}^P| + \mathcal{X}\mathcal{Y}), |\mathcal{B}|(\mu + |\mathcal{N}^P| \log \mu) \})$$

证毕。

算法复杂度为多项式, 网络规模也能控制在有限范围, 因此能够快速响应新到达的SFC请求。

5 仿真实验

本节首先给出模拟实验的参数设置, 然后介绍RASE在不同条件下的训练过程, 最后分析本文所提出的算法和不同基线算法的SFC接受率和经济成本的对比结果。

5.1 参数设置

RASE算法和贪婪策略GSR在一台拥有8核第11代Intel(R) Core(TM) i7-11700 @ 2.50 GHz、64 GB内存和NVIDIA Geforce RTX 3090 GPU的计算机上模拟运行。

实验所使用的所有数据均来自Alibaba cluster-trace-v2018数据集^[33]。从数据集中选择了15个物理设备, 其中包括9个地面设备、4个空中设备和2个轨道卫星设备。根据设备资源的异构型, 空天

设备的参数通过对三种已有服务器设备的物理资源数据分别加[0, 20]和[20, 40]范围内服从正态分布的随机数得到, 从而形成了SAGIN三层结构节点的资源。其中, 设备的内存单位设置为GB, 而传输带宽为Mbps。与Zeng等人论文中的理论一致, 物理设备的可靠性设置为[0.999, 0.99999]范围内服从均匀分布的随机值。计算、内存和带宽的单位资源代价为[0.1, 1]元范围内均匀分布的数值。根据SFC长度最大不超过7, 所选的SFC长度范围为[4, 7], SFC中每个VNF所需的计算、内存和带宽资源均来自上述数据集中的真实数据。软件可靠性如前文对物理硬件可靠性的描述, 在区间[0.999, 0.99999]内服从均匀分布。同时, 最大可容忍SFC可靠性来自集合{0.99, 0.995, 0.999, 0.9995, 0.9999}^[29]。由于模型的复杂性, 以及强化学习的状态向量即神经网络的输入层较大, 因此RASE中的策略网络和目标网络共有三层, 每层隐藏层有1024个神经元。其他参数见表2。

表2 其他参数说明

参数	说明	值
lr	学习率	0.01
T	Batch容量	64
γ	折扣因子	0.98
$ \mathcal{B} $	DDQN回放集 \mathcal{B} 大小	100 000
ϵ	贪婪策略初始化参数	0.99
\aleph	训练目标网络更新所间隔代数	50
α	奖励中代价权重	0.6
β	奖励中延迟权重	0.4

本文将通过RASE和GSR混合算法与以下一些基线方法进行比较来说明它的一些优势。以下是对基线算法的简单介绍:

(1) RuleDRL: 作者在动态环境中执行可靠的感知SFC配置。RuleDRL考虑在多个设备上备份以增强可靠性^[29]。

(2) SADDQN: 本文中, 作者提出了一种在DDQN中加入自注意力层来解决SFC嵌入问题时对于物理拓扑注意力不够导致性能浪费问题的方法^[34]。

(3) FRL: 本文提出一种称为联邦强化学习(Federated Reinforcement Learning, FRL)的算法。FRL是一种联邦学习(Federated Reinforcement, FL)和DRL的混合方法, 用于解决SAGIN异构网络中的SFC嵌入挑战^[35]。

与上述三种基线算法相比, 本文提出的解决方案特别考虑了SAGIN的环境特性, 并通过混合算法采用备份方法来提高可靠性。该方案可以使SFC具有高可靠性, 而成本接近基线算法。

5.2 训练过程

首先, 在不同数量的隐藏层神经元下评估学习过程。结果如图4所示。设定训练过程中 $|\mathcal{N}^P|=15$, SFC数量 $|\mathcal{S}|=10$, SFC长度范围记为 $|\mathcal{N}^V|$, 大小在区间[4, 7]内。总训练集规模为1000。

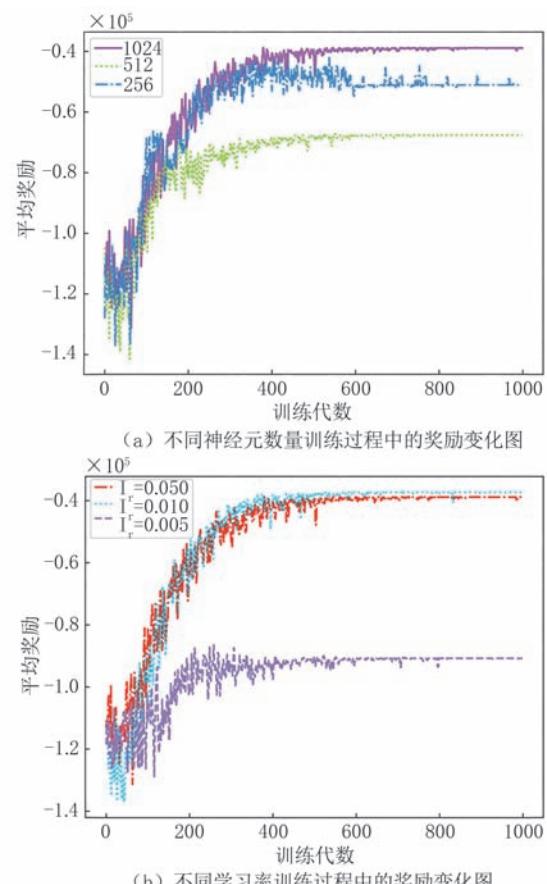


图4 训练过程

在图4(a)中, RASE在不同数量的神经元之间都能表现出收敛性。不同的神经元数量表示神经网络的不同拟合能力, 导致在不同的奖励上收敛, 从而产生不同的奖励和代价。其中, 1024个神经元具有最大的奖励。在隐藏层中有512个神经元的情况下, 智能体倾向于优先考虑额外的备份, 以提高可靠性并减少延迟。然而, 更多备份导致相关的成本增加, 因此智能体获得的奖励更低。不同数量的神经元的收敛速度非常接近, 在第400次左右达到收敛。经过分析, 偶尔会出现嵌入式备份数量较少且成本较低的情况。然而, 这样的解决方案可能会违反约

束,因此需要重新训练。以上现象是由于RASE的输入层规模超过260,而隐含层神经元数量为256和512时,RASE的两个神经网络规模较小,无法学习到有效的信息,因此会产生欠拟合问题。

图4(b)展示了不同学习率下的学习过程,当学习率为0.01时,获得的奖励最高。当学习率过小或过大时,网络都无法收敛到最优值,而学习率过大或过小,网络可能会陷入局部最优解而无法跳出,影响模型稳定性,同时,不同的学习率也会由于步长大小影响训练速率。

表3描述了不同阈值 Ψ 下的SFC接受率。触发重新嵌入的难度是由于阈值 Ψ 决定。从表中可以看出,当阈值达到95%时重新嵌入仍然可以使SFC接受率达到100%。但是当 Ψ 为99%时,剩余的1%资源不足以满足SFC的要求,导致SFC被拒绝。为了降低重嵌入的触发概率,提高系统响应时间,在后续实验中选取 $\Psi=0.90$ 。

表3 不同阈值的SFC接受率

网络负载阈值 Ψ	SFC接受率
80%	100%
85%	100%
90%	100%
95%	100%
99%	92.1%

5.3 SFC接受率和成本

本节评估RASE与GSR混合算法、RuleDRL、SADDQN和FRL在相同条件下的表现及优缺点。仿真实验对比方法的设置中,采用了和混合算法相同的超参数设置,例如相同的网络拓扑,学习率等。对于RuleDRL,其应用场景为边缘环境,因此将其直接应用于SAGIN中,并没有详细区分不同层的异构性,参数设置相同。而FRL算法需要异构拓扑环境,本文则通过将原有网络拓扑划分为多份,每份环境分别包含了不同规模的三层设备,SADDQN算法则相比较RASE算法额外添加自注意力层,隐含层的神经元数量与输入输出规模两者相同。

仿真使用SFC接受率作为衡量指标之一。每次SFC请求进入时,都会有两个决策:接受和拒绝。SFC接受率表示在一段时间内请求到达后被接受的比例。在图中展示了在不同SFC长度和数量的条件下,SFC接受率和成本的变化和区别。图5(a)具体展示了不同方法SFC接受率随着SFC长度变化的变化,而图5(b)则显示了SFC接受率在不同数

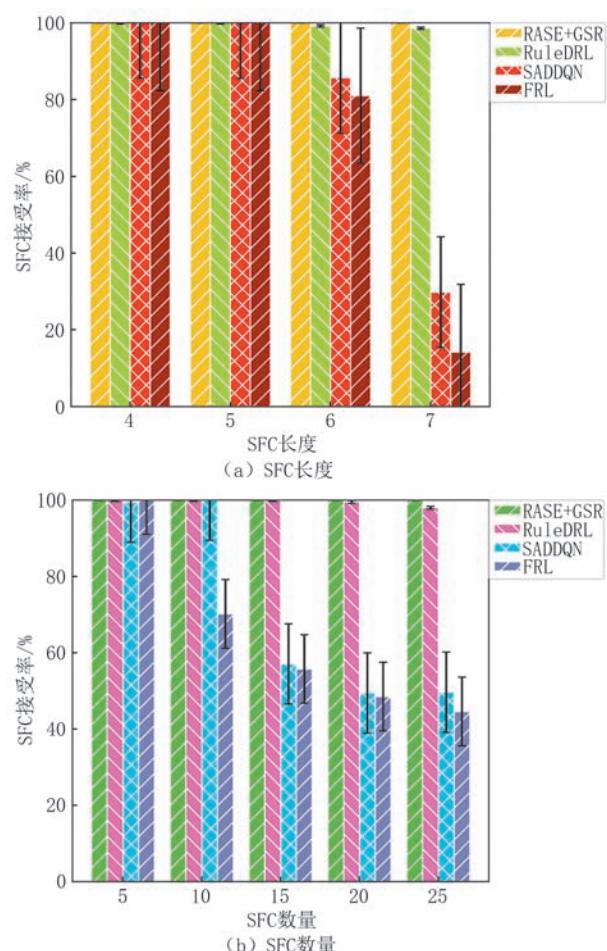


图5 SFC接受率在不同自变量下的变化

量的SFC的情况下值。图6(a)和(b)则分别介绍了经济成本随着SFC长度和SFC数量变化而变化的值。

如图5(a)所示,当SFC长度增加时,混合算法的接受率达到仍然能保持100%。这归因于在资源不足的情况下算法会激活GSR,从而节省了大量的资源。对于RuleDRL、SADDQN和FRL,当SFC长度为4和5时,它们的接受率也达到100%。然而,随着长度增加到6和7,SADDQN和FRL的接受率急剧下降。具体而言,不同长度的RuleDRL、SADDQN和FRL的接受率分别为99.2%、85.7%、81.0%、98.6%、29.8%和14.2%。这是由于随着SFC长度的增加,两种方法中缺少备份方法,极大降低了SFC的可靠性,导致SFC被拒绝。SFC由一系列VNF构成一条链,当链中的一个节点服务失败,整个链都会无法提供服务。

在图5(b)中展示了不同数量的SFC到达时SFC接受率的变化,其中SFC长度设置为5。如图所示,所提出的算法的SFC接受率始终保持在

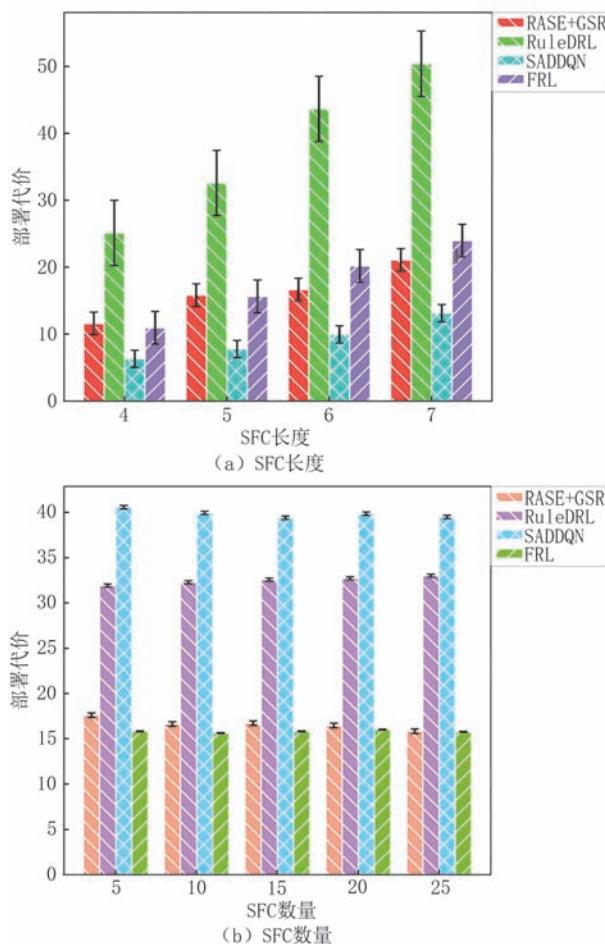


图6 代价在不同自变量下的变化

100%。具体来说,RuleDRL保持接近100%的SFC接受率。然而,SADDQN的接受率分别为99.4%、100%、57.07%、49.45%和49.64%,而FRL的接受率为100%、70.2%、55.73%、48.5%和44.6%。对于后两种算法,当SFC数量超过一定阈值时,边缘设备资源不足会导致后续到达的SFC因资源不足或可靠性受损而被拒绝。

图6显示了不同条件下的经济成本。在图6(a)中,所提出的算法在不同SFC长度下的成本是可以接受的,分别为11.6020、15.8521、16.6751和21.0933。相比之下,RuleDRL分别付出25.12、32.59、43.66和50.43的成本。SADDQN的成本分别为6.327、7.793、9.973和13.156。虽然混合算法的成本略高于SADDQN,与FRL相当,但SFC接受率能够基本达到100%,提高了系统的服务质量。图6(b)显示了不同SFC数量下相关算法的成本。从图中可以看出,与其他三种解决方案相比,本文提出的方法实现了较低的成本,并且可以保证100%的接受率,具有显著的优势。比较这四个图可以发

现,虽然RuleDRL可以在不同变量的变化下维持较高的SFC接受率,但本文所提出的算法可以实现约50%的成本节省。由于本文所提出的框架能够在网络负载过重的情况下及时切换部署方案,因此在能保证SFC接受率的前提下,当部署备份过多,系统会及时以最小成本进行部署,保障SFC可靠性,相比于RuleDRL,本文所提方案就可以极大节约成本。

6 结语

本文深入研究了SAGIN中异构可靠SFC嵌入与重嵌入问题。现有工作由于没有考虑到SAGIN的环境特性,不能很好地应用于HRSER问题。复杂的网络拓扑和设备异构性导致嵌入SFC时需要考虑多方面的挑战。资源的动态性和稀缺性可能需要在SFC嵌入后进行频繁调整。为了解决这些挑战,提出了混合算法,该算法结合了DDQN和贪婪策略来提高SFC的接受率。针对SAGIN的异构性,提出了一种基于DDQN的算法RASE。该算法通过考虑备份,增强了SFC的可靠性。此外,RASE使用DDQN来选择每个VNF的主实例。为了应对后期高网络负载,进一步提出了一种贪婪算法GSR进行重嵌入,有效地提高了SFC的接受率。此外,还从理论上证明了重新嵌入的SFC备份数量的下界。最后,通过在真实数据集上进行模拟实验,提出的RASE和GSR算法对不同的神经元数量和学习率表现出了鲁棒性。与现有的优秀算法相比,混合算法的SFC接受率接近100%,且成本较低,仅为具有类似SFC接受率的算法RuleDRL所需成本的一半。

致谢 褒心感谢编辑们和评审专家们对本文提出的宝贵意见和建议!

参考文献

- [1] Xiong Ze-Kai. Research on deployment strategy of service function chain in space-ground integrated network architecture [Master Dissertation]. Guang Xi University, Guangxi, 2023 (in Chinese)
(熊泽凯. 空天地一体化网络构架中服务功能链的部署策略研究 [硕士学位论文]. 广西大学, 广西, 2023)
- [2] Wu Yan-Yan, Wu Song. Research on resource management of edge computing in air-space-ground-sea integrated network. Mobile Communications, 2024, 48(09):124-131 (in Chinese)

- (武燕燕, 吴松. 空天地海一体化网络边缘计算的资源管理研究. *移动通信*, 2024, 48(09):124-131)
- [3] Liu Xue-Fang, Mao Wei-Hao, Yang Qing-Hai. Air-space-ground integrated network resource allocation algorithm based on deep reinforcement learning. *Journal of Electronics and Information Technology*, 2021, 46(07): 2831-2841 (in Chinese)
(刘雪芳, 毛伟灏, 杨清海. 基于深度强化学习的空天地一体化网络资源分配算法. *电子与信息学报*, 2021, 46(07):2831-2841)
- [4] White Paper on 6G-oriented Space-Ground Converged Network Technology, <http://221.179.172.81/images/20240204/2971707048250617.pdf> (in Chinese)
(面向6G的天地一体融合网络技术白皮书, <http://221.179.172.81/images/20240204/2971707048250617.pdf>)
- [5] Pei J, Hong P, Xue K, et al. Resource aware routing for service function chains in SDN and NFV-enabled network. *IEEE Transactions on Services Computing*, 2018, 14(4): 985-997
- [6] Liu Y, Lu Y, Li X, et al. On dynamic service function chain reconfiguration in IoT networks. *IEEE Internet of Things Journal*, 2020, 7(11): 10969-10984
- [7] Fan Qi-Lin, Niu Yue, Yin Hao. GAT-IL: A service function chain deployment method based on graph attention network and imitation learning. *Journal of Electronics*, 2024, 52(08): 2811-2823 (in Chinese)
(范琪琳, 牛岳, 尹浩等. GAT-IL:一种基于图注意力网络与模仿学习的服务功能链部署方法[J]. *电子学报*, 2024,52(08):2811-2823)
- [8] Wu Jing-Wen, Jiang Ling-Yun. Cross-domain deployment algorithm of service function chain. *Computer Engineering and Design*, 2023, 44(08): 2263-2271 (in Chinese)
(武静雯, 江凌云. 服务功能链的跨域部署算法. *计算机工程与设计*, 2023,44(08):2263-2271)
- [9] Kibalya G, Serrat J, Gorricho J L, et al. A deep reinforcement learning-based algorithm for reliability-aware multi-domain service deployment in smart ecosystems. *Neural Computing and Applications*, 2023, 35(33): 23795-23817
- [10] Hubenko V P, Raines R A, Mills R F, et al. Improving the global information grid's performance through satellite communications layer enhancements. *IEEE Communications Magazine*, 2006, 44(11): 66-72
- [11] Pultarova T. Telecommunications-space tycoons go head to head over mega satellite network . *Engineering & Technology*, 2015, 10(2): 20-20
- [12] Liu Y, Lu Y, Qiao W, et al. Reliability-aware service chaining mapping in NFV-enabled networks. *Etri Journal*, 2019, 41(2): 207-223
- [13] Qu L, Khabbaz M, Assi C. Reliability-aware service chaining in carrier-grade softwarized networks. *IEEE Journal on Selected Areas in Communications*, 2018, 36(3): 558-573
- [14] Karimzadeh-Farshbafan M, Shah-Mansouri V, Niyato D. A dynamic reliability-aware service placement for network function virtualization (NFV). *IEEE Journal on Selected Areas in Communications*, 2019, 38(2): 318-333
- [15] Erbati M M, Schiele G. A novel reliable low-latency service function chaining to enable URLLC in NFV//Proceedings of the 2022 IEEE Ninth International Conference on Communications and Networking. Online, 2022: 1-8
- [16] Huang M, Liang W, Shen X, et al. Reliability-aware virtualized network function services provisioning in mobile edge computing. *IEEE Transactions on Mobile Computing*, 2019, 19(11): 2699-2713
- [17] Li J, Liang W, Huang M, et al. Reliability-aware network service provisioning in mobile edge-cloud networks. *IEEE Transactions on Parallel and Distributed Systems*, 2020, 31(7): 1545-1558
- [18] Jia J, Yang L, Cao J. Reliability-aware dynamic service chain scheduling in 5G networks based on reinforcement learning// Proceedings of the 2021-IEEE Conference on Computer Communications. Online, 2021: 1-10
- [19] Fu X, Yu F R, Wang J, et al. Dynamic service function chain embedding for NFV-enabled IoT: A deep reinforcement learning approach. *IEEE Transactions on Wireless Communications*, 2019, 19(1): 507-519
- [20] Hsu Y H, Tsai T R, Yeh T C, et al. Deep reinforcement learning based mobility-aware SFC embedding for MEC in 5G and beyond//Proceedings of the 2023 IEEE Wireless Communications and Networking Conference. Glasgow, Scotland, UK, 2023: 1-6
- [21] Wang T, Fan Q, Li X, et al. Drl-sfcp: Adaptive service function chains placement with deep reinforcement learning// Proceedings of the 2021-IEEE International Conference on Communications. Online, 2021: 1-6
- [22] Solozabal R, Ceberio J, Sanchoyerto A, et al. Virtual network function placement optimization with deep reinforcement learning. *IEEE Journal on Selected Areas in Communications*, 2019, 38(2): 292-303
- [23] Yan Z, Ge J, Wu Y, et al. Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks. *IEEE Journal on Selected Areas in Communications*, 2020, 38(6): 1040-1057
- [24] Chen Y T, Liao W. Mobility-aware service function chaining in 5G wireless networks with mobile edge computing// Proceedings of the 2019 IEEE International Conference on Communications. Shanghai, China, 2019: 1-6
- [25] Jin P, Fei X, Zhang Q, et al. Latency-aware VNF chain deployment with efficient resource reuse at network edge// Proceedings of the 2020 IEEE Conference Computer Communications. Online , 2020: 267-276
- [26] Emu M, Yan P, Choudhury S. Latency aware VNF deployment at edge devices for IoT services: An artificial neural network based approach//Proceedings of the 2020 IEEE International Conference on Communications Workshops. Online , 2020: 1-6
- [27] Yang L, Jia J, Lin H, et al. Reliable dynamic service chain scheduling in 5G networks. *IEEE Transactions on Mobile Computing*, 2022, 22(8): 4898-4911
- [28] Tu H, Zhao G, Xu H, et al. A robustness-aware real-time SFC routing update scheme in multi-tenant clouds. *IEEE/ACM Transactions on Networking*, 2022, 30(3): 1230-1244

- [29] Zeng Y, Qu Z, Guo S, et al. Ruledrl: Reliability-aware SFC provisioning with bounded approximations in dynamic environments. *IEEE Transactions on Services Computing*, 2023, 16(5): 3651-3664
- [30] Zhang P, Zhang Y, Kumar N, et al. Dynamic SFC embedding algorithm assisted by federated learning in space-air-ground-integrated network resource allocation scenario. *IEEE Internet of Things Journal*, 2022, 10(11): 9308-9318
- [31] Pei J, Hong P, Xue K, et al. Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 30(10): 2179-2192
- [32] Li H, Kong Z, Chen Y, et al. Slice-based service function chain embedding for end-to-end network slice deployment. *IEEE Transactions on Network and Service Management*, 2023, 20(3): 3652-3672
- [33] Service function chaining use cases in mobile networks, <http://tools.ietf.org/html/draft-haefner-sfc-use-case-mobility-00> 2014.
- [34] Wang Z, Guo S, Liu G. Sa-ddqn: Self-attention mechanism based ddqn for sfc deployment in nfv/mec-enabled networks// *Proceedings of the 2022 IEEE 28th International Conference on Parallel and Distributed Systems*. Online, 2023: 720-727
- [35] Huang H, Zeng C, Zhao Y, et al. Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 2021, 39(8): 2558-2571

附录.

定理1. HRSER 问题是 NP 难的。

证明. 此处证明将会首先特殊化该问题, 并通过极限示例且使用背包问题证明上述定理。假设 $|\mathcal{N}^V|=1, \mathfrak{R}^s=\nu_u=\varsigma_l=0, M_u^p=B_l^p=\infty, \forall s \in \mathcal{S}$ 。在此极限情况下, HRSER 问题可以转化为 0-1 背包问题, 目标是在给定集合中选择物品从而实现最大总价值, 且总重量不超过背包容量。由于 0-1 背包问题已知是 NP 难问题, 因此 HRSER 问题也是 NP 难问题。

证毕。

定理2. 当 SFC 的最小可容忍可靠性为 \mathfrak{R}^s

时, 最小备份数 μ 为 $\frac{\log_2 \left[1 - (\mathfrak{R}^s)^{\frac{1}{len}} \right]}{\log_2 (1 - \nu_{min} \phi_{min})}$ 。

证明. 当约束(4)满足时可得到

$$\prod_{f_i \in s} R^{f_i, s} \geq \mathfrak{R}^s, \forall s \in \mathcal{S}.$$

由于可靠性最低的 VNF 必须部署更多备份, 可

以在此认为 μ 是可靠性最低的 VNF 所需的备份数量。同时, 若所有 VNF 软件可靠性都等于最低可靠性, 并根据约束条件(5)和(6), 可得到以下式子,

$$\left(1 - (1 - \nu_{min} \phi_{min})^{\mu} \right)^{len} \geq \mathfrak{R}^s.$$

其中, len 为 SFC 的最短长度。从而可以将其简化如下,

$$1 - (1 - \nu_{min} \phi_{min})^{\mu} \geq (\mathfrak{R}^s)^{\frac{1}{len}}.$$

然后, 通过两边取对数, 可以得到:

$$\mu \geq \frac{\log_2 \left[1 - (\mathfrak{R}^s)^{\frac{1}{len}} \right]}{\log_2 (1 - \nu_{min} \phi_{min})}.$$

由于 μ 是整数, 因此可以得到以下结果:

$$\mu = \frac{\log_2 \left[1 - (\mathfrak{R}^s)^{\frac{1}{len}} \right]}{\log_2 (1 - \nu_{min} \phi_{min})}.$$

证毕。



QIN Kai-Xin, M. S. candidate. His research interests include network function virtualization, and edge computing.

GUO Song-Tao, Ph. D., professor. His research interests include wireless ad hoc networks and mobile edge computing.

LIU Gui-Yan, Ph. D., associate researcher. Her research interests include data center networks, software defined networking, network function virtualization and 5G network.

Background

The research problem of this paper belongs to the field of

network optimization in the context of space-air-ground integrated networks. Specifically, it focuses on the embedding

and re-embedding of Service Function Chains (SFCs), which is a crucial aspect for enhancing the efficiency, quality of service and flexibility of network services.

Internationally, significant progress has been made in understanding the fundamental concepts and challenges related to SFC embedding. However, the dynamic and heterogeneous nature of the space-air-ground network environment poses unique difficulties that are still being actively explored. Current solutions often struggle to adapt to rapid changes in network conditions and the complex requirements of diverse services.

This study aims to develop more efficient and adaptive algorithms for SFC embedding and re-embedding, taking into

account the specific characteristics of the space-air-ground network. We strive to improve the overall performance, quality of service and resource utilization of the network, enabling more reliable and flexible service provision.

This research is part of the Reliable Deployment and Orchestration of Mobile Service Function Chains for Heterogeneous Edge Networks. Our research group has previously conducted studies in the area of SFC embedding in edge network, network resource management and optimization. The results have provided valuable insights and methodologies that will be further extended and refined in this research to address the challenges of SFC embedding and re-embedding in the space-air-ground network.