

空天地一体化网络中基于联邦深度强化学习的边缘协作缓存策略

刘亮¹, 荆腾祥¹, 段洁¹, 毛武平¹, 燕洪成², 马文杰²

(1.重庆邮电大学通信与信息工程学院, 重庆 400065; 2.中国空间技术研究院, 北京 100094)

摘要: 针对偏远地区网络覆盖范围有限的问题, 将空天地一体化网络与移动边缘计算相结合, 可以实现这些地区用户请求的低时延和高可靠传输, 并能及时提供缓存服务。考虑到空天地一体化网络拓扑的动态变化和-content流行度不断更新的特点, 首先提出了一种空天地一体化边缘协作缓存的网络架构。然后, 将边缘服务器的缓存替换问题建模为马尔可夫决策过程。最后, 提出了一种联邦离散柔性演员评论家(FDSAC)算法, 其核心思想是将加权注意力机制融入联邦学习框架中, 并将双向长短期记忆网络集成到DSAC模型。以重构后的奖励函数为优化目标, 通过最大化长期负奖励的期望来学习最优的缓存替换策略。仿真结果表明, 与其他算法相比, 所提算法可以在保护用户隐私的前提下, 将用户请求的缓存命中率提高18%, 内容的访问时延降低25%。

关键词: 空天地一体化网络; 移动边缘计算; 缓存; 离散柔性演员评论家; 联邦学习

中图分类号: TN927

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025014

Federated deep reinforcement learning-based edge collaborative caching strategy in space-air-ground integrated network

LIU Liang¹, JING Tengxiang¹, DUAN Jie¹, MAO Wuping¹, YAN Hongcheng², MA Wenjie²

1. School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

2. China Academy of Space Technology, Beijing 100094, China

Abstract: To address the problem of limited network coverage in remote areas, combining space-air-ground integrated network with mobile edge computing could provide low-latency and high-reliability transmissions for user requests in these areas, as well as timely caching services. Considering the dynamic change of the topology of the space-air-ground integrated network and the content popularity being constantly updated, a network architecture of space-air-ground integrated edge collaborative caching was proposed first. Then, the cache replacement problem for edge servers was modeled as a Markov decision process. Finally, a federated discrete soft actor-critic (FDSAC) algorithm was proposed, with the core idea of integrating a weighted attention mechanism into the federated learning framework and incorporating a bi-directional long short-term memory network into the DSAC model. With the reconfigured reward function as the optimization objective, the optimal cache replacement policy was learned by maximizing the expectation of negative long-term rewards. Simulation results show that compared with other algorithm, the proposed algorithm can improve the cache hit rate of user requests by 18% and reduce the access latency of content by 25% while protecting user privacy.

Keywords: space-air-ground integrated network, mobile edge computing, caching, discrete soft actor-critic, federated learning

收稿日期: 2024-11-06; 修回日期: 2025-01-08

通信作者: 段洁, duanjie@cqupt.edu.cn

基金项目: 国家自然科学基金资助项目(No.62171070, No.61701058); 重庆市教委科学技术研究基金资助项目(No.KJQN202200615)

Foundation Items: The National Natural Science Foundation of China (No.62171070, No.61701058), The Science and Technology Research Program of Chongqing Municipal Education Commission (No.KJQN202200615)

0 引言

随着物联网的快速发展,物联网终端设备 and 应用不断增长,而偏远地区由于网络通信基础设施建设成本较高和地理限制,且终端设备的计算和存储能力有限,导致一系列时延敏感的内容请求得不到及时的处理和响应。尽管传统的云计算网络可以为这些用户请求提供强大的计算和存储能力,但是由于传输距离远,会造成较大的传输时延并消耗大量的带宽资源,进而严重影响用户的体验质量(QoE, quality of experience)^[1]。同时,用户数据上传至云数据中心可能导致严重的数据安全和隐私泄露问题。移动边缘计算(MEC, mobile edge computing)通过将云数据中心丰富的计算和存储资源下沉到离终端设备更近的网络边缘,使终端设备能够快速将计算任务和缓存服务卸载或迁移到附近的 MEC 服务器进行处理^[2],从而为物联网应用提供更加高效和及时的服务。

针对偏远地区用户的内容请求问题,移动边缘缓存技术可以有效降低用户的内容检索时延^[3]。由于热门且流行的内容可能会被多个用户频繁请求,因此从 MEC 服务器缓存和检索这些内容会比从远程云端获取数据的效率更高。当前,空天地一体化网络(SAGIN, space-air-ground integrated network)作为未来 6G 网络发展的重要支柱^[4],引起了学术界和工业界的广泛关注。在 SAGIN 中,卫星通信网络能够提供覆盖全球的通信服务,而由无人机、飞艇等组成的空中网络可以辅助卫星网络,进一步减轻卫星骨干网络的通信拥塞程度以及上行链路的传输负担。因此,可以将 SAGIN 与 MEC 相结合,在卫星或无人机上部署一系列轻量级的 MEC 服务器,并利用其存储能力将一些用户请求的热门流行内容缓存起来,从而为偏远地区用户的内容请求提供及时的响应。然而,空天地一体化协同的边缘网络具有高度的异构性和动态性,同时用户内容流行度也在不断更新,这给用户内容请求在 MEC 服务器的缓存命中带来了极大的挑战。

目前,大多数的缓存方案主要基于传统规则的优化算法或启发式算法。文献[5]首次尝试从应用程序供应商的角度研究 MEC 环境中的公平边缘数据缓存(FEDC, fair edge data caching)问题,并分别采用优化方法和近似算法求解不同规模的 FEDC 问题。文献[6]重点研究了异构物联网中不同大小

内容的概率缓存,并提出了一种改进的缓存概率转换算法导出最佳缓存概率的封闭式解决方案。文献[7]提出了一种缓存辅助的 SAGIN 移动边缘计算系统,采用基于交替优化的方法对无人机轨迹、资源分配和卸载决策进行联合优化。然而,实验结果表明,上述基于传统规则的缓存替换算法只能粗略地预测内容流行度,无法有效应对用户需求和内容流行度的快速变化,且难以适应高度动态变化的空天地协同的边缘缓存网络环境。

随着人工智能技术的不断发展,大量学者开始应用机器学习算法寻找最优的缓存方案。文献[8]提出了一种多智能体强化学习(RL, reinforcement learning)算法,最小化下一代超密集网络中的整体平均传输时延,并利用变分循环神经网络估计内容随时间变化的流行度。文献[9]采用分层架构部署物联网中的边缘缓存节点,并提出了一种基于近端策略优化算法的缓存方案,在考虑数据新鲜度和有限生命周期的前提下,提高缓存命中率并降低能量消耗。然而,上述集中式学习算法需要频繁地将大量的终端设备数据上传至中央服务器,而在需要快速预测内容流行度的应用场景中,这种方法将带来额外的传输时延,可能无法满足实时性的要求。此外,集中式学习算法必须依赖于终端用户所提供的大量本地数据做出最优决策,但这可能会引发用户数据的隐私泄露和安全风险,使用户往往不愿意直接分享他们的数据。在此场景中,联邦学习(FL, federated learning)可以通过允许用户共享他们的本地模型而不是本地数据来提供一种解决这些限制和保护用户隐私的可行方案。

综上所述,现有的缓存替换方案在面对大规模动态边缘缓存网络环境时仍存在不足。因此,本文提出了一种基于联邦离散柔性演员评论家(FD-SAC, federated discrete soft actor-critic)的缓存替换算法,通过 FL 框架来避免目前的集中式学习算法在收集用户数据时存在的隐私风险和产生的额外通信开销,并利用离散柔性演员评论家(DSAC, discrete soft actor-critic)模型应对大规模用户请求的异构性和网络的动态性,从而在保护用户隐私的前提下,提高用户内容请求的缓存命中率,并最小化用户的内容访问时延。本文的主要贡献如下。

1) 设计了一种 SAGIN 中的边缘协作缓存架构,为了提高缓存命中率,将影响用户 QoE 的因素建

模为由时延成本和缓存命中成本组成的综合系统成本。

2) 将 MEC 服务器的缓存替换问题建模为马尔可夫决策过程, 将负标量的系统成本设计为奖励函数, 以提高缓存命中率并降低用户内容访问时延。

3) 将加权注意力机制融入 FL 框架中, 并将双向长短期记忆网络 (BiLSTM, bidirectional long short-term memory) 集成到 DSAC 模型, 提出了一种 FDSAC 算法, 该算法以重构后的奖励函数为优化目标, 通过最大化长期负奖励的期望学习最优的缓存替换策略。

4) 进行了大量的仿真实验, 结果表明本文算法与现有算法相比, 在提高缓存命中率和降低内容访问时延方面有更好的性能。

1 相关工作

本节讨论分析目前的相关工作进展以及本文所提 FDSAC 算法的研究动机。

1.1 星地网络中的缓存策略

在传统的地面网络中, 内容通常被缓存在基站 (BS, base station) 中, 用于移动用户的边缘缓存。然而, 在星地网络中, 内容可以缓存在 BS、卫星和网关中。文献[10]提出了一种星地网络中的协作传输方案, 通过联合优化缓存放置和内容调度来最小化平均文件的传输时延。文献[11]研究了虚拟化低地球轨道 (LEO, low earth orbit) 星地网络中缓存放置和功率分配的联合优化问题, 并采用基于蜉蝣算法的智能优化算法确保在问题求解过程中摆脱局部最优解。文献[12]在卫星回程传输中考虑了基站和用户的协同缓存, 然后基于社交关系选择帮助节点并提出了贪婪缓存策略以最小化内容检索时延。文献[13]研究了混合星地中继网络中的缓存问题, 通过结合 2 种不同的缓存放置方案来缓解频谱短缺问题, 提高频谱效率。文献[14]研究了基站、卫星和网关协同为地面用户提供内容缓存服务的多层边缘缓存协同通信技术, 并提出了非协作式和协作式 2 种缓存策略, 以降低通信时延。然而, 上述研究是基于星地网络场景, 并且没有考虑内容访问时延和缓存命中率的联合优化问题。

1.2 无人机辅助网络中的缓存策略

为了提高缓存效率, 无人机已经被用作一种更灵活的飞行基站, 以动态缓存流行内容并跟踪相关

用户的移动模式, 提供更加高效的服务。文献[15]提出了一种无人机辅助网络的概率缓存策略, 可以显著提高缓存命中率从而减少存储资源的消耗。文献[16]提出了一种 MEC 和无人机辅助的缓存策略, 为特定地理区域的用户提供更快的内容检索, 该策略考虑了无人机有限的存储容量和用户偏好, 以提高用户的 QoE。文献[17]提出了一种用于支持缓存的多无人机网络的 QoE 引导内容交付框架, 通过创建地面移动用户集群, 找到所有无人机的最佳位置和高度以及优化缓存策略来最大限度地提高缓存用户的 QoE。文献[18]提出了一种无人机辅助网络中基于决斗强化学习的内容缓存方案, 在最大化用户的 QoE 的同时最小化无人机的平均发射功率。文献[19]研究了启用缓存的无人机通信网络的性能, 并提出了一种优化缓存策略, 以最大化网络空间吞吐量, 减少回程通信链路的负担。然而, 上述研究仅通过无人机来辅助内容缓存, 通信覆盖范围有限。同时, 其有限的存储容量和负载能力会导致缓存效率降低和内容访问时延增加。

1.3 空天地一体化网络中的缓存策略

基于 SAGIN 三层网络体系结构, 可以为不同类型的服务提供灵活的网络接入和多样化的传输方式。文献[20]提出了一种由卫星和无人机组成的 SAGIN 移动边缘缓存物联网系统, 并采用智能优化算法降低系统的总体通信成本, 保证系统的高可用性。文献[21]结合随机森林模型提出了一种新的分层分布式编码缓存方案, 以解决 SAGIN 资源利用率低、存储复杂和流量繁忙的问题。文献[22]针对内容的时空流行度, 提出了一个基于深度 Q 网络的内容选择方法以优化缓存命中率。文献[23]提出了一种基于缓存的卫星-无人机-车辆一体化网络, 考虑到内容流行度、缓存大小、请求频率和不同路段移动性的影响, 设计了一种能量感知的编码缓存策略, 提高网络的资源利用效率。文献[24]研究了 SAGIN 中多媒体服务的联合缓存和资源分配, 同样采用交替迭代优化框架寻找最优解以最小化服务时延。然而, 上述研究都采用集中式学习算法解决缓存问题, 需要在数据中心持续收集大量用户数据以学习参数模型, 这可能导致网络拥塞和用户隐私泄漏, 并且它们没有考虑多跳相邻卫星或多跳无人机之间的协作缓存。

因此, 为了进一步增强本文所提缓存方案的效

率 and 安全性, 基于 FL 框架设计了 FDSAC 算法, 通过将加权注意力机制融入 FL 框架, 并将 BiLSTM 集成到 DSAC 模型中, 有效实现 SAGIN 中内容协作缓存命中率和内容访问时延的联合优化。

2 系统模型

2.1 物理网络

本文所设计的空天地一体化三层边缘协作缓存架构如图 1 所示。

卫星节点集合表示为 $V = \{v_1, v_2, \dots, v_L\}$, 无人机节点集合表示为 $M = \{m_1, m_2, \dots, m_N\}$, 地面用户设备集合表示为 $U = \{u_1, u_2, \dots, u_k\}$, MEC 服务器集合表示为 $E = \{e_1, e_2, \dots, e_{L+N}\}$ 。部署在卫星或无人机上的 MEC 服务器具有一定的缓存能力, 卫星 MEC 服务器的缓存大小为 C_v , 无人机 MEC 服务器的缓存大小为 C_m , 且 $C_m < C_v$ 。地面用户设备发出的内容请求可以由无人机或卫星 MEC 服务器提供相应的内容服务。地面用户设备请求的所有内容的集合为 $\mathbb{F} = \{1, 2, \dots, F\}$, 假设每个内容的大小为 $C_f, f \in \mathbb{F}$ 。一般情况下, 假设云服务器拥有足够的容量存储集合 \mathbb{F} 中的所有内容。在 MEC 服务器中,

假设内容 f 的局部流行度为 p_f , 则全局流行度可以表示为 $P = \{p_1, p_2, \dots, p_F\}$ 。对于任意的 MEC 服务器, 地面用户设备在时隙 t 请求内容 f 的概率为 $p_f^t = \frac{u_f^t}{\sum_{j \in \mathbb{F}} u_j^t}$, 并且有 $\sum_{f \in \mathbb{F}} p_f^t = 1$ 。其中, u_f^t 表示地面用户设备在时间隙 t 请求内容 f 的次数, $\sum_{j \in \mathbb{F}} u_j^t$ 表示地面设备的总请求次数。此外, 假设在每个时隙 t , 用户设备只会发出一个内容请求。

2.2 协作缓存模型

由于地面用户设备的请求内容会随时间变化, 当地面用户设备移动到无人机或卫星覆盖的范围并向它们发出内容请求时, 如果此时相应的无人机没有缓存用户请求的内容, 则会将该请求发送给其集群网络内相邻 n_m 跳的无人机, 如果相邻 n_m 跳内的无人机缓存了该内容, 也属于缓存命中, 否则缓存命中失败。如果此时相应的卫星没有缓存用户请求的内容, 则请求会进一步发送给其相邻 n_s 跳的卫星, 如果相邻 n_s 跳内的卫星缓存了该内容, 则属于缓存命中, 否则缓存命中失败。无人机缓存命中失败后会将请求转发给卫星去检索该内容, 而卫星缓

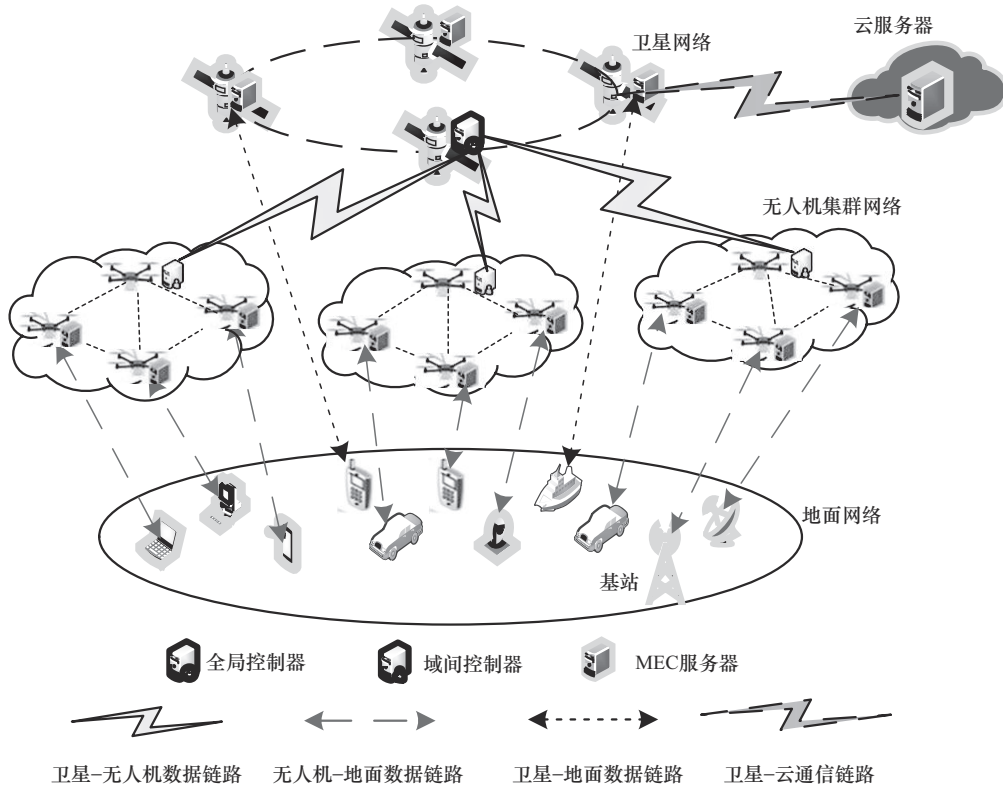


图 1 空天地一体化三层边缘协作缓存架构

存命中失败最终会将请求转发给云服务器进行处理。假设地面用户设备只能向无人机或卫星上的一个 MEC 服务器发送内容请求。因此,地面用户设备所请求的内容平均缓存命中率可表示为

$$p_{\text{hit}_f}^{\text{ave}} = \frac{\sum_u \sum_f p_f^i (\gamma_{u,f}^m + \gamma_{u,f}^v)}{U} \quad (1)$$

其中, U 是地面总用户设备数; p_f^i 是用户请求内容的流行度; $\gamma_{u,f}^m$ 是一个二进制变量, 表示地面用户设备 u 请求的内容 f 是否存储在无人机 MEC 服务器 m 中及其相邻 $n_m \in \mathbf{N}^+$ (正整数集) 跳内的无人机上, 即是否缓存命中; $\gamma_{u,f}^v$ 也是一个二进制变量, 表示地面用户设备 u 请求的内容 f 是否存储在卫星 MEC 服务器 v 中及其相邻 n_s 跳内的卫星上, 即是否缓存命中。

$\gamma_{u,f}^m$ 和 $\gamma_{u,f}^v$ 的计算式分别如式(2)和式(3)所示。

$$\gamma_{u,f}^m = \begin{cases} 1, \gamma_{u,f}^{m,1} = 1 \text{ 或 } \gamma_{u,f}^{m,n_m} = 1 \\ 0, \text{其他} \end{cases} \quad (2)$$

$$\gamma_{u,f}^v = \begin{cases} 1, \gamma_{u,f}^{v,1} = 1 \text{ 或 } \gamma_{u,f}^{v,n_s} = 1 \\ 0, \text{其他} \end{cases} \quad (3)$$

其中, 二进制变量 $\gamma_{u,f}^{m,1} = 1$ 表示在相应的无人机 m 上缓存命中内容 f , 否则 $\gamma_{u,f}^{m,1} = 0$; 二进制变量 $\gamma_{u,f}^{m,n_m} = 1$ 表示在无人机 m 相邻 n_m 跳内的无人机协作缓存命中内容 f , 否则 $\gamma_{u,f}^{m,n_m} = 0$; 二进制变量 $\gamma_{u,f}^{v,1} = 1$ 表示在相应的卫星 v 上缓存命中, 否则 $\gamma_{u,f}^{v,1} = 0$; 二进制变量 $\gamma_{u,f}^{v,n_s} = 1$ 表示在卫星 v 相邻 n_s 跳内的卫星协作缓存命中内容 f , 否则 $\gamma_{u,f}^{v,n_s} = 0$ 。

$\gamma_{u,f}^{m,1}$ 、 $\gamma_{u,f}^{m,n_m}$ 、 $\gamma_{u,f}^{v,1}$ 和 $\gamma_{u,f}^{v,n_s}$ 的计算式如式(4)~式(7)所示。

$$\gamma_{u,f}^{m,1} = \begin{cases} 1, \beta_f^m z_u^m = 1, f \in \mathbb{F} \\ 0, \beta_f^m z_u^m = 0, f \in \mathbb{F} \end{cases} \quad (4)$$

$$\gamma_{u,f}^{m,n_m} = \begin{cases} 1, \beta_f^{m,n_m} z_u^m = 1 \text{ 且 } \gamma_{u,f}^{m,1} \neq 1, f \in \mathbb{F} \\ 0, \beta_f^{m,n_m} z_u^m = 0 \text{ 或 } \gamma_{u,f}^{m,1} = 1, f \in \mathbb{F} \end{cases} \quad (5)$$

$$\gamma_{u,f}^{v,1} = \begin{cases} 1, \beta_f^v z_u^v = 1, f \in \mathbb{F} \\ 0, \beta_f^v z_u^v = 0, f \in \mathbb{F} \end{cases} \quad (6)$$

$$\gamma_{u,f}^{v,n_s} = \begin{cases} 1, \beta_f^{v,n_s} z_u^v = 1 \text{ 且 } \gamma_{u,f}^{v,1} \neq 1, f \in \mathbb{F} \\ 0, \beta_f^{v,n_s} z_u^v = 0 \text{ 或 } \gamma_{u,f}^{v,1} = 1, f \in \mathbb{F} \end{cases} \quad (7)$$

其中, 二进制变量 β_f^m 表示请求内容 f 与无人机 MEC 服务器缓存之间的关系, 如果内容 f 先前已经缓存在无人机 m 上, 那么 $\beta_f^m = 1$, 否则, $\beta_f^m = 0$, 二进制变量 β_f^{m,n_m} 同理; 二进制变量 z_u^m 表示地面用户设备与无人机 m 之间的关联性, 如果地面用户设备 k 在无人机 m 的覆盖范围内, 即满足 $l_{u,m} = \|\text{loc}_u - \text{loc}_m\| \leq R_1$, 则有 $z_u^m = 1$, 否则 $z_u^m = 0$, loc_u 、 loc_m 和 $l_{u,m}$ 分别表示地面用户设备 u 的位置坐标信息、无人机 m 的位置坐标信息和地面用户设备 u 与无人机 m 的距离信息, R_1 表示无人机 m 对地面用户设备的通信覆盖范围的半径; 二进制变量 β_f^v 表示请求内容 f 与卫星 MEC 服务器缓存之间的关系, 如果内容 f 先前缓存在卫星 v 上, 则有 $\beta_f^v = 1$, 否则, $\beta_f^v = 0$, β_f^{v,n_s} 同理; 二进制变量 z_u^v 表示地面用户设备与卫星 v 之间的关联性, 如果地面用户设备 u 在卫星 v 的覆盖范围内, 即满足 $l_{u,v} = \|\text{loc}_u - \text{loc}_v\| \leq R_2$, 则有 $z_u^v = 1$, 否则 $z_u^v = 0$, loc_v 表示卫星 v 的位置坐标信息, $l_{u,v}$ 表示地面用户设备 u 与卫星 v 的距离信息, R_2 表示卫星 v 对地面用户设备的通信覆盖范围的半径。

2.3 内容访问时延模型

假设地面用户设备向相应的无人机发送内容请求时, 考虑到地面用户设备之间的相互干扰和高斯白噪声, 根据香农公式, 地面用户设备到无人机的上行传输速率为

$$R_{u,m} = B_{u,m} \log \left[1 + \frac{h_{u,m} p_{u,m}}{\sum_{j \in \mathcal{U} \setminus \{u\}} h_{u,m} p_j + N_0 B_{u,m}} \right] \quad (8)$$

其中, $p_{u,m}$ 表示地面用户设备 u 到无人机 m 的上行链路发射功率, $h_{u,m}$ 表示地面用户设备 u 与无人机 m 之间的信道功率增益, $B_{u,m}$ 表示地面用户设备 u 与无人机 m 之间的信道带宽, p_j 表示其他地面用户设备 j 到无人机 m 的发射功率, N_0 表示高斯白噪声的功率谱密度。

当地面用户设备在相应的无人机覆盖范围内, 并且用户设备所请求的内容在无人机或相邻 n_m 跳内的无人机缓存命中时, 内容访问时延计算式如式(9)所示。

$$T_{u,m}^f = \psi \frac{\gamma_{u,f}^{m,1} C_f}{R_{u,m}} + (1 - \psi) \left(\frac{\gamma_{u,f}^{m,1} C_f}{R_{u,m}} + \frac{n_m \gamma_{u,f}^{m,n_m} C_f}{R_{m,m}} \right) \quad (9)$$

其中, $\psi \in \{0,1\}$ 是一个二进制变量, 当 $\psi = 1$ 时, 表示内容请求在相应的无人机上缓存命中, 否则未能在本地无人机缓存命中, 会将请求发送给其相邻 n_m 跳内的无人机, $R_{m,m}$ 为无人机之间的传输速率, $n_m \leq \frac{cR_{m,m}(h_{m,v}R_{m,v} + cC_f)}{cR_{m,v}(d_m R_{m,m} + cC_f)}$, c 为光速, d_m 为相邻无

人机之间的距离, $h_{m,v}$ 为无人机到卫星的距离, $R_{m,v}$ 为无人机到卫星的传输速率。

当地面用户设备向相应的卫星发送内容请求时, 地面用户设备到卫星的上行传输速率如式(10)所示。

$$R_{u,v} = B_{u,v} \text{lb} \left[1 + \frac{h_{u,v} P_{u,v}}{\sum_{i \in U \setminus \{u\}} h_{u,v} P_i + N_0 B_{u,v}} \right] \quad (10)$$

其中, $p_{u,v}$ 表示地面用户设备 u 到卫星 v 的上行链路发射功率, $h_{u,v}$ 表示地面用户设备 u 与卫星 v 之间的信道功率增益, $B_{u,v}$ 表示地面用户设备 u 与卫星 v 之间的信道带宽, p_i 表示其他地面用户设备 i 到卫星 v 的发射功率。

当用户设备在相应的卫星覆盖范围内, 并且用

$$T_{u,ave}^f = \frac{\sum_u \sum_f p_f^t [\alpha T_{u,m}^f + (1 - \alpha) t_{m,v} + \beta T_{u,v}^f + (1 - \alpha) (1 - \beta) t_{v,c}]}{U} \quad (12)$$

其中, $\alpha \in \{0,1\}$ 和 $\beta \in \{0,1\}$ 都是二进制变量, 当 $\alpha = 1$ 时, 表示内容请求在无人机上缓存命中, 同样, 当 $\beta = 1$ 时, 表示内容请求在卫星上缓存命中。

$t_{m,v} = \frac{h_{m,v}}{c} + \frac{C_f}{R_{m,v}}$ 为无人机到卫星的传输时延,

$t_{v,c} = \frac{d_c}{c} + \frac{C_f}{R_{v,c}}$ 为卫星到云服务器的传输时延。

2.4 缓存替换模型

当地面用户设备的请求内容在无人机或卫星上的 MEC 服务器缓存命中失败后, MEC 服务器应该替换缓存的内容, 以提高用户设备请求内容的缓存命中率 and 降低内容的平均访问时延。将无人机或卫星上的 MEC 服务器的缓存替换建模为马尔可夫决策过程, 由状态空间、动作空间和奖励函数组成。

1) 状态空间。由于无人机和卫星上的 MEC 服务器是通过观察地面用户设备的请求内容以及请求是否缓存命中来作出相应的替换动作。因此, 状态空间由 MEC 服务器的缓存状态和地面用户设备的请求状态组成。假设地面用户设备在每个时隙 t 仅

户设备所请求的内容在卫星或相邻 n_s 跳内的卫星缓存命中时, 内容访问时延计算式如式(11)所示。

$$T_{u,v}^f = \zeta \frac{\gamma_{u,f}^{v,1} C_f}{R_{u,v}} + (1 - \zeta) \left(\frac{\gamma_{u,f}^{v,1} C_f}{R_{u,v}} + \frac{n_s \gamma_{u,f}^{v,n_s} C_f}{R_{v,v}} \right) \quad (11)$$

其中, $\zeta \in \{0,1\}$ 是一个二进制变量, 当 $\zeta = 1$ 时, 表示内容请求在相应的卫星上缓存命中, 否则未能在本地卫星缓存命中, 会将请求发送给其相邻 n_s 跳内的卫星, $R_{v,v}$ 为卫星之间的传输速率, d_v 为相邻卫星之间的距离, d_c 为卫星到云服务器的距离, $R_{v,c}$ 为卫星到云服务器的传输速率, $n_s \leq \frac{cR_{v,v}(d_c R_{v,c} + cC_f)}{cR_{v,c}(d_v R_{v,v} + cC_f)}$ 。

当地面用户设备的服务请求在相应的无人机与其相邻 n_m 跳内的无人机都缓存命中失败后, 则会将请求转发给卫星 MEC 服务器去检索该内容。如果地面用户设备的服务请求在相应的卫星与其相邻 n_s 跳内的卫星都缓存命中失败后, 会将请求进一步转发给云服务器去检索。因此, 地面用户设备的平均内容访问时延可以表示为

发出一个内容请求, 则在时隙 t 时, 地面用户设备的请求状态为 $\text{req}_u^t = \{0,1,2,\dots,F\}$, $\text{req}_u^t = 0$ 表示地面用户设备 u 没有向相应的无人机或卫星发出内容请求, $\text{req}_u^t = f$ 表示地面用户设备 u 向相应的无人机或卫星发出内容 f 。无人机或卫星上 MEC 服务器的内容缓存状态为 $c_f^t \in \{0,1\}$, $f \in \mathbb{F}$, $c_f^t = 1$ 表示在相应的无人机或卫星上缓存了内容 f , 否则, $c_f^t = 0$ 。因此, 在时隙 t 的状态 s_f^t 可以表示为 req_u^t 和 c_f^t 的组合, 即 $s_f^t = [\text{req}_u^t, c_f^t]$ 。

2) 动作空间。随着用户请求内容的流行度不断更新, 缓存在无人机或卫星 MEC 服务器中的内容由于流行度较低可能导致缓存命中失败。此时, 需要考虑将原先的内容进行替换。设 a_f^t 为无人机或卫星上 MEC 服务器的缓存替换动作, 其中 $a_f^t \in \{0,1\}$, $a_f^t = 1$ 表示缓存的内容 f 应该被替换, 如果 $a_f^t = 0$ 则不会替换缓存在无人机或卫星上的内容 f 。

3) 奖励函数。为了最大化地面用户设备请求的缓存命中率和最小化内容平均访问时延, 将奖励

函数定义为缓存命中成本和时延成本的加权和。由于它们是负相关的,因此将缓存命中成本表示为 $\phi_f^t = (1 - p_{\text{hit}_f}^{\text{ave}}) \in [0,1]$ 。所以奖励函数计算式如式(13)所示。

$$R_f^t(s_f^t, a_f^t) = \lambda_1 \phi_f^t + \lambda_2 T_{u,\text{ave}}^t \quad (13)$$

其中, λ_1 和 λ_2 分别为缓存命中成本和时延成本的权重,也用于平衡缓存命中成本和时延成本的量纲, $\lambda_1, \lambda_2 \in [0,1]$, 且 $\lambda_1 + \lambda_2 = 1$ 。

3 问题刻画

通过最小化由缓存命中成本和时延成本组成的系统综合成本,提高地面用户设备请求的缓存命中率并降低用户请求的平均访问时延。因此,需要为无人机和卫星上的每个 MEC 服务器设计一个策略来生成最优的替换动作。为了简化计算,将奖励函数重构为 $r_f^t = -R_f^t(s_f^t, a_f^t)$, 通过最大化长期累积的负奖励期望来最小化系统成本,从而学习最优的策略 π^* , 即将问题刻画为

$$\begin{aligned} & \max_{\pi} \mathbb{E} \left[\sum_{i=t}^{\infty} \gamma^{i-t} r_f^i \pi \right] \\ \text{s.t. C1: } & \sum_{f \in \mathbb{F}} C_f \leq C_v, \sum_{f \in \mathbb{F}} C_f \leq C_m \\ \text{C2: } & \sum_{u \in U} B_{u,v} \leq B_v, \sum_{u \in U} B_{u,m} \leq B_m \\ \text{C3: } & \psi, \zeta \in \{0,1\} \\ \text{C4: } & \sum_{f \in \mathbb{F}} a_f^t = 1 \\ \text{C5: } & a_f^t \in \{0,1\} \end{aligned} \quad (14)$$

其中, $\gamma \in (0,1]$ 作为奖励的折扣因子; 约束条件 C1 表示缓存在无人机和卫星上的内容不能超过其最大的缓存能力; C2 表示无人机 m 和卫星 v 能接入的最大地面用户设备数, 其中 B_v 表示分配给卫星 v 的最大通信带宽, B_m 表示分配给无人机 m 的最大通信带宽; C3 表示地面用户设备的内容请求是否在本地卫星或本地无人机缓存命中; C4 表示无人机或卫星 MEC 服务器在 t 时刻最多只能替换一个内容; C5 表示替换动作的二进制变量。该问题是一个混合整数非线性规划问题, 它是 NP 难的, 常规方法很难求解。

4 基于联邦深度强化学习的边缘缓存方法

首先, 本文提出了一种柔性演员评论家 (SAC,

soft actor critic) 的多智能体离散变体, 采用加权注意力机制控制 FL 聚合步骤中的模型权重, 同时保护用户的隐私安全。然后, 基于 BiLSTM 设计内容流行度预测模型, 并将它与 DSAC 模型聚合以实现更高的预测精度。最后, 提出 FDSAC 算法来学习最优的缓存替换策略。

4.1 注意力加权的 FL 框架

考虑到用户的隐私安全问题, FL 通过使用分布式 MEC 服务器的本地数据来协作训练全局模型。与集中式训练方式相比, 它可以显著降低用户的隐私风险, 并大幅降低通信成本。然而, 在复杂、动态的 SAGIN 环境中, 典型的 FL 面临着局部模型质量不均衡的挑战。为了应对这一挑战, 本文结合注意力机制来优化聚合权重, 提出了一种注意力加权的 FL 框架, 如图 2 所示。FL 模型的训练是通过多轮通信完成的, 每轮通信都包含以下步骤。

1) 模型部署。在每个无人机和卫星边缘节点上部署 DSAC 模型, 并且在云端部署全局模型, 全局模型进行预训练产生初始化的全局参数。

2) 模型参数下发。当智能体 (即 DSAC 模型) 在各个 MEC 节点上部署完成后, 云服务器将以广播的方式将全局参数下发给这些智能体。

3) 模型参数训练。MEC 节点上的边缘智能体根据自身存储的历史数据对策略参数 θ 和柔性 Q 函数参数 ϕ 进行训练。

4) 上传更新后的参数。将各智能体在边缘本地训练好的策略参数 θ 上传到云服务器。

5) 注意力加权的模型参数聚合。云服务器通过计算从所有边缘智能体接收到的局部策略参数 θ 的加权和, 生成全局策略参数 θ_{global} 。

由于各个 MEC 服务器的计算能力和缓存的数据质量都有所差异, 因此根据经典 FL 框架中平均地集成每个边缘模型的参数是不合理的。通过引入注意力加权的聚合机制, 采用 2 个与 MEC 服务器相关的指标作为衡量标准评估局部模型对全局模型的贡献, 以此来赋予局部模型策略参数的不同权重值。第一个指标就是 MEC 服务器的活动水平。由于卫星的飞行速度相对无人机较快, 离地面的距离也较远。这意味着地面用户设备更倾向于将自身的请求数据发送到无人机 MEC 服务器上, 因此对于无人机来说更有可能训练出高质量模型。换句话说, 不同活动级别的 MEC 服务器对全局模型的贡

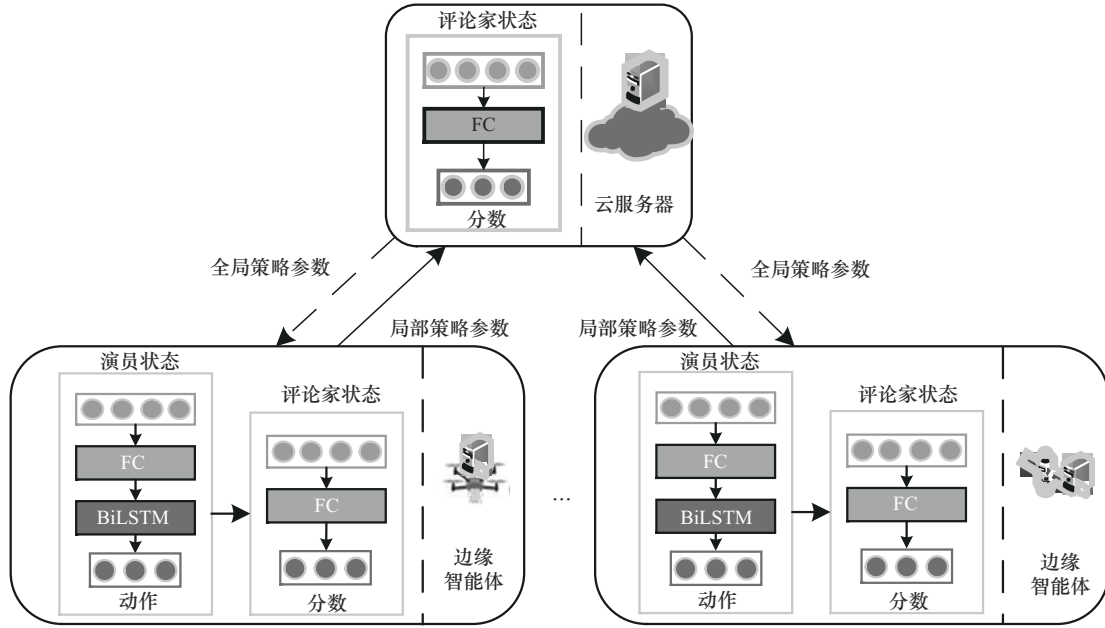


图2 注意力加权联邦学习框架

献各不相同。

因此，通过观察窗口内各 MEC 服务器的历史请求数量来评估各 MEC 服务器的活动水平。MEC 服务器 e_i 的活动级别可以表示为

$$a_{r+1}^{e_i} = \frac{\|d_{t-\rho:t}^{e_i}\|}{\sum_{i=1}^{L+N} \|d_{t-\rho:t}^{e_i}\|}, e_i \in E \quad (15)$$

其中， $\|d_{t-\rho:t}^{e_i}\|$ 表示观察窗口 $[t-\rho, t]$ 内 MEC 服务器 e_i 中上传的所有内容的请求总数， ρ 表示观察窗口的大小。

第二个指标是边缘训练的周期数。事实上，不同 MEC 服务器的计算能力不同，这会导致计算相同训练任务的时间不同。从侧面也表明了不同 MEC 服务器的训练周期数不同。由于训练周期数是决定模型收敛性的关键因素，因此对于训练周期数较大的局部训练模型应赋予较大的权重。由于模型更新随着训练周期数的增加而减慢，因此采用 \log 函数来计算边缘模型参数训练周期数的权重。MEC 服务器 e_i 训练时期的权重通过式(16)计算。

$$q_{r+1}^{e_i} = \frac{\log(e_{r+1}^{e_i})}{\log(\max\{e_{r+1}\})}, e_{r+1} = \{e_{r+1}^{e_1}, e_{r+1}^{e_2}, \dots, e_{r+1}^{e_{L+N}}\} \quad (16)$$

其中， $e_{r+1}^{e_i}$ 表示 MEC 服务器 e_i 在第 $(r+1)$ 次通信轮数中的边缘训练周期数。值得注意的是，边缘训

练周期数的最大值 ($\max\{e_{r+1}\}$) 是根据训练结果选择的，根据注意力加权聚合机制，MEC 服务器通过式(17)聚合它们各自训练的模型参数。

$$\theta_{r+1}^{\text{global}} = \sum_{i=1}^{L+N} a_{r+1}^{e_i} q_{r+1}^{e_i} \theta_{r+1}^{e_i} \quad (17)$$

4.2 基于 BiLSTM 的内容流行度预测模型

在人们的日常工作中，由于大多数内容请求模式在时间域上是周期性的，如一天或一周。因此，采用 BiLSTM 预测每个 MEC 服务器覆盖区域的内容流行度。基于 BiLSTM 的内容流行度预测模型框架如图 3 所示。在该模型中，以长度为 ρ 时间段的观察窗口内的内容请求的历史数量作为输入，可以表示为 $D = [D_{t-\rho}, D_{t-\rho+1}, \dots, D_t]$ 。将历史数据输入模型后，通过 3 个自参数化控制门，即输入门 g_i^t 、遗忘门 g_f^t 和输出门 g_o^t 来控制长、短时间段历史信息的记忆。3 个控制门的表达式分别如式(18)~式(20)所示。同时，LSTM 单元的长期记忆 c_t 通过输入门、遗忘门和输出门等机制，负责在时间序列中传递和更新关键信息。而 LSTM 单元的短期记忆 h_t 表示隐藏状态，作为当前时间步的输出。 c_t 和 h_t 的计算式分别如式(21)和式(22)所示。

$$g_i^t = \delta(w_i[h_{t-1}, D_t] + b_i) \quad (18)$$

$$g_f^t = \delta(w_f[h_{t-1}, D_t] + b_f) \quad (19)$$

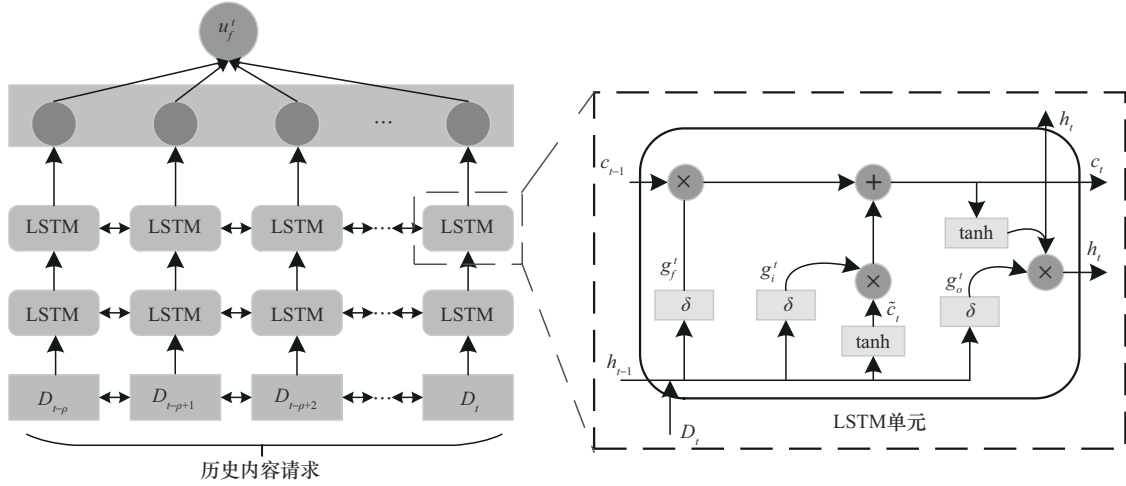


图3 基于BiLSTM的内容流行度预测模型框架

$$g_o^t = \delta(w_o[h_{t-1}, D_t] + b_o) \quad (20)$$

$$c_t = g_c^t c_{t-1} + g_i^t \tanh(w_c[h_{t-1}, D_t] + b_c) \quad (21)$$

$$h_t = g_o^t \tanh(c_t) \quad (22)$$

其中, w_i, w_f, w_o, w_c 和 b_i, b_f, b_o, b_c 分别表示要学习的 BiLSTM 的权重和偏差, δ 表示激活函数, \tanh 表示激活函数 $\frac{e^x - e^{-x}}{e^x + e^{-x}}$.

然后, 采用全连接网络融合 BiLSTM 的输出, 可表示为 $o = h_1 \oplus h_2 \oplus \dots \oplus h_t$. 最后, 在激活操作后, 可以得到 t 时隙内容 f 的请求数为

$$u_f^t = \delta(o) \quad (23)$$

4.3 基于 FDSAC 算法的缓存替换策略

1) 离散柔性演员评论家算法。SAC 是一种具有最大熵目标的高级深度强化学习 (DRL, deep reinforcement learning) 算法, 然而常见的 SAC 仅适用于连续动作的仿真实验设置。为了克服 SAC 对连续动作的限制, 设计了一种基于 SAC 的多智能体离散变体, 以更高效地处理高维环境中的离散决策。而 DSAC 的目标就是通过找到一个随机策略 π 来最大化预期累积奖励和熵, 即

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t (r^{t+1} + \mu \mathcal{H}(\pi(\cdot | s^t))) \right] \quad (24)$$

其中, $\mathcal{H}(\pi(\cdot | s^t)) = \mathbb{E}[-\log \pi(a | s^t)]$ 表示熵项, 可以用来衡量随机变量的不确定性, μ 表示平衡熵项与奖励相对重要性的温度参数, 一般较大的 μ 会使智能体在决策过程中进行更加随机的探索。

为了评估所有有限离散动作, 柔性 Q 函数应输出每个可能动作的 Q 值。因此, 柔性 Q 函数定义为

$$Q(s^t, a^t) = r^t + \gamma \mathbb{E}[V(s^{t+1})] \quad (25)$$

其中, $V(s^t) = \pi(\cdot | s^t) \left[Q(s^t, a^t) - \mu \log \pi(\cdot | s^t) \right]$ 表示具有熵增累积收益的柔性价值函数。

在策略更新过程中, 与传统的通过最大化 Q 值的离线策略不同, DSAC 更新的策略 π_{new} 与 $Q(s^t, a^t)$ 的指数分布成正比。

给定策略 π_{old} 和具有有限动作空间的柔性 Q 函数 $Q_{\pi_{\text{old}}}(s^t, a^t)$, 新策略 π_{new} 如式(26)所示。

$$\pi_{\text{new}} = \arg \min_{\pi} D_{\text{KL}} \left(\pi^t(\cdot | s^t) \parallel \frac{\exp\left(\frac{Q_{\pi_{\text{old}}}(s^t, a^t)}{\mu}\right)}{Z_{\pi_{\text{old}}}(s^t, a^t)} \right) \quad (26)$$

其中, $D_{\text{KL}}(\cdot)$ 表示 KL (Kullback-Leibler) 散度, $Z_{\pi_{\text{old}}}(s^t, a^t)$ 表示用于标准化 Q 值的分布。最小化策略分布和柔性 Q 函数指数分布之间的 KL 散度相当于最大化目标函数, 并且对于新策略 π_{new} 都有 $Q_{\pi_{\text{new}}}(s^t, a^t) \geq Q_{\pi_{\text{old}}}(s^t, a^t)$ 成立。

与典型强化学习的策略迭代求解类似, 可以迭代求解柔性 Q 函数和策略更新 2 个过程, 最终找到最优策略。该方案要确保最优策略在状态-动作空间中是离散的, 并且可以在有限迭代内找到。在本文中, 通过使用函数逼近器 (即深度神经网络 (DNN, deep neural network)) 来寻找 DSAC 模型

的最优策略，并且将 $Q_\phi(s^t, a^t)$ 和 $\pi_\theta(a^t | s^t)$ 分别表示为 DNN 的参数化柔性 Q 函数和参数化策略函数，其中 ϕ 和 θ 是 DNN 的参数。最后，通过最小化柔性贝尔曼残差的均方误差作为 $Q_\phi(s^t, a^t)$ 的目标函数，即

$$J_Q(\phi) = E_{s^t \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\phi(s^t, a^t) - \hat{Q}(s^t, a^t) \right)^2 \right] \quad (27)$$

其中， \mathcal{D} 表示回放缓冲区，用来存储历史的训练数据（即 $d^t = \{s^t, a^t, r^t, s^{t+1}\} \in \mathcal{D}$ ）， $\hat{Q}(s^t, a^t)$ 表示目标柔性 Q 函数。通过梯度下降法对 $J_Q(\phi)$ 进行梯度更新来最小化损失函数，即

$$\hat{\nabla}_\phi J_Q(\phi) = \nabla Q_\phi(s^t, a^t) \left(Q_\phi(s^t, a^t) - (r^t + \gamma V(s^{t+1})) \right) \quad (28)$$

类似地， $\pi_\theta(a^t | s^t)$ 的目标函数是最小化 KL 散度，即

$$J_\pi(\theta) = E_{s^t \sim \mathcal{D}} \left[D_{\text{KL}} \left[\pi_\theta(\cdot | s^t) \parallel \frac{\exp\left(\frac{Q_\theta(s^t, a^t)}{\mu}\right)}{Z_\theta(s^t, a^t)} \right] \right] \quad (29)$$

由于 $Z_\theta(\cdot)$ 是配分函数且与 θ 无关，因此可以省略。此外，该策略输出的是精确的动作分布，而非连续动作的 SAC 所使用的重参数化技术。因此， $\pi_\theta(a^t | s^t)$ 的损失函数可以表示为

$$J_\pi(\theta) = E_{s^t \sim \mathcal{D}} \left[\pi_\theta(a^t | s^t)^T \left(\mu \log \pi_\theta(a^t | s^t) - Q_\theta(s^t, a^t) \right) \right] \quad (30)$$

同样，通过梯度下降法对 $\pi_\theta(a^t | s^t)$ 进行梯度更新来最小化损失函数，即

$$\hat{\nabla} J_\pi(\theta) = \nabla_\theta \pi_\theta(a^t | s^t)^T \left(\mu \log \pi_\theta(a^t | s^t) - Q_\theta(s^t, a^t) \right) + \pi_\theta(a^t | s^t)^T \mu \nabla_\theta \log \pi_\theta(a^t | s^t) \quad (31)$$

2) 神经网络的设计。将演员网络设计为五层神经网络，由一个输入层、2个隐藏层、一个 BiLSTM 层和一个输出层组成。而评论家网络设计为四层神经网络，与演员网络相比，缺少一个 BiLSTM 层。DSAC 的演员网络和评论家网络都是由全连接网络构建的，但是全连接网络无法完全获得隐藏的内容请求模式。此外，历史请求的状态信息会

增加输入空间，使得 DSAC 难以学习最优策略。因此，可以利用具有记忆机制的 BiLSTM 改进演员网络以捕获隐藏的内容请求模式。具体来说，在隐藏层和输出层之间添加一个 BiLSTM 层，而单层的 BiLSTM 由 2 个 LSTM 组合而成，一个负责正向去处理输入序列，另一个负责反向处理序列，处理完成后将 2 个 LSTM 的输出拼接起来。最后，由全连接层的输出和隐藏矩阵 $\mathbf{H}(t)$ 组合起来作为输入，其中隐藏矩阵负责动态记录和观察动作的完整历史信息。因此，使用 BiLSTM 可以更好地捕捉较长距离的依赖关系。

3) 算法训练。在本文所提 FDSAC 的缓存替换算法中，每个 MEC 服务器都部署一个智能体（即本地 DSAC 模型），该智能体独立训练本地模型并协作聚合模型参数。与现有的将所有本地模型聚合到全局模型的 FL 不同，本文将 DSAC 的演员网络和评论家网络进行分离，并且仅将所有本地评论家网络的参数聚合到云服务器中的全局评论家网络。这是因为理论上所有评论家网络对于当前网络环境都应该有相同的评价标准，但由于用户请求的异构性，演员网络应该有所不同。

首先，各边缘智能体通过观察其本地的状态 s_f^t 来生成相应的替换动作，并计算出相应的奖励值 r_f^t 。然后，将当前的奖励值 r_f^t 、状态 s_f^t 、动作 a_f^t 和下一状态 s_f^{t+1} 作为一个四元组 $d^t = \{s_f^t, a_f^t, r_f^t, s_f^{t+1}\}$ 存储在经验回放缓冲区内。接下来，基于批量大小为 N 的元组数据更新和学习演员网络和评论家网络。经过多次迭代后，当训练周期数 episode 等于聚合数量的倍数时，所有 MEC 服务器上的智能体上传它们评论家网络的策略参数 θ 到云服务器中进行加权聚合。最后，云服务器将更新后的全局策略参数 θ_{global} 下发给 MEC 服务器。否则，将继续进行多轮次的迭代，直到达到模型参数训练的精度时，输出最优的缓存替换策略 π^* 。具体算法流程如算法 1 所示。

算法 1 联邦离散柔性演员评论家 (FDSAC) 算法

输入 地面用户设备的一组内容请求 $\text{req}_u = \{1, 2, \dots, F\}$

输出 最优缓存替换策略 π^*

1) 初始化策略参数 θ ，柔性 Q 函数参数 ϕ ，全局策略参数 θ_{global} ，聚合数量 Δ ，最大训练轮

- 次 K 和经验回放缓冲区 \mathcal{D}
- 2) for episode=1 to K do
 - 3) for t to T do
 - 4) for $e_i \in E$ do
 - 5) 通过观察各自的本地状态 s_f^t 做出替换动作 $a_f^t \leftarrow \pi_\theta(\cdot | s_f^t)$
 - 6) 然后得到当前的奖励值 r_f^t 和下一状态 s_f^{t+1}
 - 7) 将四元组 $d^t = \{s_f^t, a_f^t, r_f^t, s_f^{t+1}\}$ 存储到经验回放缓冲区 \mathcal{D} 中
 - 8) 从各自的经验回放缓冲区 \mathcal{D} 中随机选择一批大小为 \mathcal{N} 的元组数据进行本地训练
 - 9) 基于梯度下降法, 根据式(28)和式(31)分别更新演员网络中的柔性 Q 函数参数 ϕ 和评论家网络中的策略参数 θ
 - 10) end for
 - 11) end for
 - 12) if episode mod $\Delta = 0$ then
 - 13) 对各 MEC 服务器的局部参数 θ 进行加权处理并上传到云服务器上
 - 14) 根据联邦聚合算法对加权局部参数 θ 进行聚合, 并通过式(17)更新全局策略参数 θ_{global}
 - 15) 然后将更新后的全局策略参数 θ_{global} 下发给所有 MEC 服务器
 - 16) end if
 - 17) end for

5 性能分析

5.1 仿真设置

本文所有的仿真实验都是在 64 位的 Windows 10 系统上进行的, 该系统配备 16 GB RAM、Intel i7 CPU 和 GTX1050-Ti GPU。编程工具采用 Python 3.6, 并通过 Python NetworkX3.1 库来生成网络拓扑。神经网络的训练使用开源的 PyTorch 框架。设置一个具有 50 个地面终端设备的节点和 10 个 MEC 服务器节点的场景, 其中卫星 MEC 服务器节点和无人机 MEC 服务器节点的个数分别为 3 个和 7 个^[25], 并将两者的节点覆盖范围分别设置为半径等于 5 km 和 1 km 的圆形区域。地面用户设备到卫星的上行链路发射功率设置为 30 dBm, 信道带宽设置为

20 MHz^[26]。地面用户设备到无人机的上行链路发射功率设置为 20 dBm, 信道带宽设置为 10 MHz^[27]。卫星和无人机上 MEC 服务器的缓存大小分别设置为 150 Mbit 和 100 Mbit。此外, 本文采用合成的数据集, 其考虑 50 个终端用户发送的 200 个内容请求的场景。每个内容请求的大小设置为 5~10 Mbit, 终端设备所发送的内容请求都是从内容目录中随机挑选的, 并且每个用户的请求序列都遵循 Zipf 分布。具体的仿真参数设置如表 1 所示。

表 1 仿真参数设置

参数名称	值
地面用户设备到卫星的发射功率/dBm	30
地面用户设备与卫星之间的信道带宽/MHz	20
地面用户设备到无人机的发射功率/dBm	20
地面用户设备与无人机之间的信道带宽/MHz	10
地面用户设备的内容请求大小/Mbit	[5,10]
卫星 MEC 服务器的缓存大小/Mbit	150
无人机 MEC 服务器的缓存大小/Mbit	100
神经元的数量	128
学习率	0.000 3
优化器的类型	Adam
折扣因子 γ	0.99
小批量的大小 \mathcal{N}	256
聚合数量 Δ	100
温度参数 μ	0.3
经验回放缓冲区 \mathcal{D} 的大小	10 000

为了验证 FDSAC 算法的有效性, 设计和选择与本文研究方向紧密相关的 3 种对比算法在本文设计的相同场景中进行仿真实验对比。

1) 深度 Q 网络 (DQN, deep Q-network)^[28]。DQN 是一种经典的 DRL 算法, 它可以在小规模设置中获得接近最优的结果。

2) 深度双 Q 网络 (DDQN, double deep Q-network)^[29]。DDQN 是一种双深度 Q 网络算法, 它通过将选择动作和评估动作分割开来避免过度估计的问题, 从而达到更好的收敛效果。

3) 近端策略优化 (PPO, proximal policy optimization)^[30]。PPO 是一种策略梯度方法, 通过限制每次更新的步长, 有效确保了策略更新的稳定性和有效性。

5.2 计算复杂度分析

本节提供了上述对比算法和FDSAC算法的计算复杂度的对比分析。DQN算法在执行时的主要计算步骤是通过神经网络进行前向传播来估算每个动作的Q值，其计算量就是所有层之间的计算量之和。因此，DQN在每个时间步执行时的计算复杂度可以表示为 $O(\sum_{i=0}^{L-1} x_i x_{i+1})$ ，其中L表示神经网络的层数， x_i 和 x_{i+1} 分别表示第i层和第i+1层的神经元数量。DDQN与DQN的核心区别是它通过引入2个深度Q网络来避免Q值的过度估计问题，其决策过程包含2次神经网络的前向传播，而每次前向传播的计算复杂度均为 $O(\sum_{i=0}^{L-1} x_i x_{i+1})$ 。因此，DDQN在每个时间步执行时的总计算复杂度可以表示为 $O(\sum_{i=0}^{L-1} x_i x_{i+1})$ 。PPO与DQN在每个时间步的计算复杂度相似，也是 $O(\sum_{i=0}^{L-1} x_i x_{i+1})$ 。对于FDSAC算法，其计算复杂度主要由策略网络和价值函数网络的前向传播过程决定。因此，FDSAC算法在每个时间步做出决策的总计算复杂度可以表示为 $O(\sum_{i=0}^{L_A-1} x_i x_{i+1} + \sum_{i=0}^{L_C-1} y_i y_{i+1})$ ，其中 L_A 和 L_C 分别表示演员网络和评论家网络的网络层数， x_i 和 y_i 分别表示对应网络第i层的神经元数量。

5.3 仿真结果分析

图4为不同训练轮次下系统平均奖励值的比较。从图4中可以看出，随着训练轮次的增加，4种算法的系统平均奖励值在逐步上升。这是因为模型参数在不断训练，所积累的奖励值越来越多。但是当训练轮次达到1500次后，本文所提FDSAC算法的系统平均奖励值开始趋于稳定，这就意味着算法开始收敛了。而其他3种对比算法，当训练轮次达到2000次后，系统平均奖励值趋于稳定且算法也逐步收敛。这是由于FDSAC算法采用了分布式的FL框架，能够加快模型的训练速度，从而使得算法快速收敛。此外，从图4中还可以看出，FDSAC算法的系统平均奖励值始终高于其他3种对比算法。这也表明了FDSAC算法的性能优于其他3种对比算法，能够为用户请求提供优越的缓存策略。

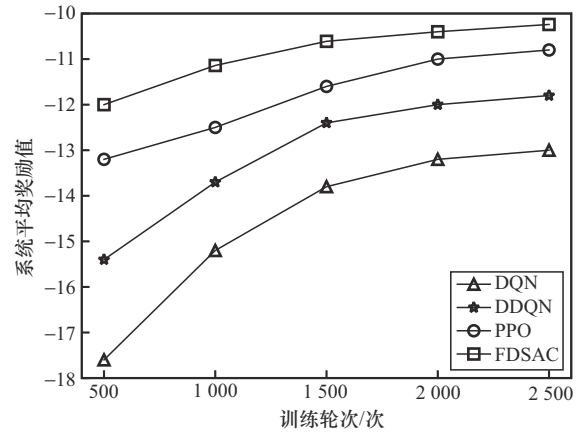


图4 不同训练轮次下系统平均奖励值的比较

图5展示了4种算法在不同地面终端设备数量下的平均内容访问时延。从图5中可以看出，随着地面终端设备数量的增多，系统的平均内容访问时延也会急剧增加。这是由于随着地面终端设备数量的增多，用户所发起的内容请求也会随之增多。在面对大量的用户请求时，网络可能会出现流量拥塞的情况，故MEC服务器无法对内容请求做出及时响应。因此，就会造成内容请求在MEC服务器经历较长的等待时延。此外，从图5中还可以看出，本文所提FDSAC算法的平均内容访问时延是最小的，而DQN算法是最大的。这是因为FDSAC算法融合了加权联邦学习框架和DSAC模型，在面对大规模的用户请求时，能够快速做出相应的缓存决策。而DQN算法只能在小规模环境设置中获得接近最优的决策，无法应对大规模的用户请求。同时，也体现出了FDSAC算法在内容访问时延方面的性能是优于其他3种对比算法的。

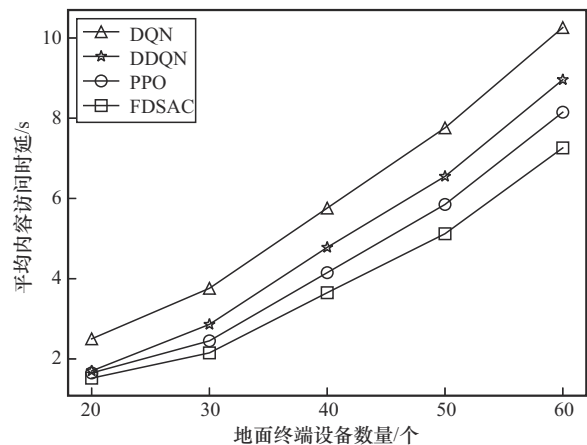


图5 不同地面终端设备数量下的平均内容访问时延

图6展示了不同地面终端设备数量下的内容平均缓存命中率。从图6中可以看出,随着地面终端设备数量的增多,4种算法下的内容平均缓存命中率在不断降低。这是因为每个MEC服务器的缓存大小有限,只能缓存内容目录中部分内容。当面对大量的用户请求时,MEC服务器由于缓存的内容过少导致缓存命中率降低了。另外,随着内容流行度的不断变化,MEC服务器的缓存内容未及时更新,导致无法满足新内容请求,使得缓存命中率下降。从图6中还可以看出,本文所提FDSAC算法下的内容平均缓存命中率要高于其他3种对比算法。这是由于FDSAC算法融合了多MEC服务器协作缓存的思想以及在DSAC模型中嵌入了BiLSTM层,可以对内容的流行度进行有效的预测,从而提高内容平均缓存命中率。

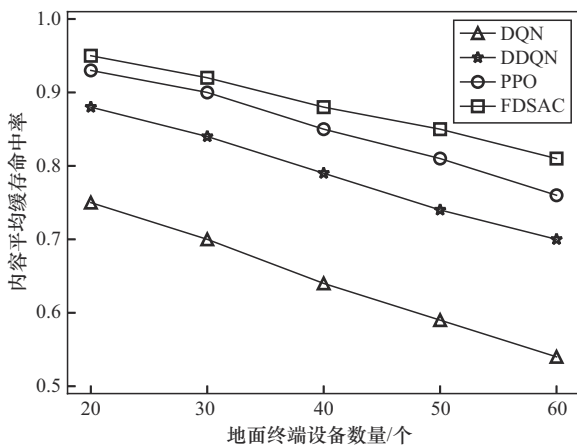


图6 不同地面终端设备数量下的内容平均缓存命中率

图7将本文所提FDSAC算法在不同地面终端设备数量下的系统综合成本与其他3种对比算法进行了比较。从图7中可以看出,随着地面终端设备数量的增多,系统综合成本在急剧上升。这是因为系统综合成本由时延成本和缓存命中成本组成,随着地面终端设备发起的内容请求越来越多,会造成内容请求在MEC服务器上的处理时延较长。并且随着内容请求的增多,会造成边缘缓存命中率降低,从而导致缓存命中成本的提高。此外,从图7中还可以看出,FDSAC算法下的系统综合成本要低于其他3种对比算法。这是由于FDSAC算法集成了DSAC模型和加权联邦学习框架,可以应对大量的用户请求,从而提高用户请求在MEC服务器的响应速度和缓存命中率。

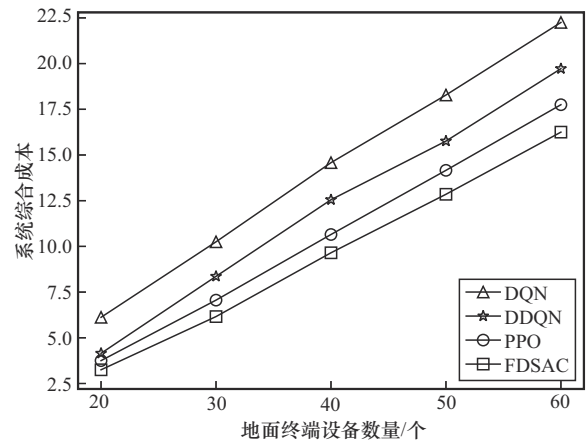


图7 不同地面终端设备数量下的系统综合成本

图8展示了不同MEC服务器数量下的系统综合成本。从图8中可以看出,随着MEC服务器的数量增多,4种算法下的系统综合成本在不断降低。这是由于当MEC服务器的数量增加时,可以扩大MEC服务器在内容目录中存储的缓存内容。当地面用户设备发起大量的内容请求时,可以得到快速的响应,从而降低了时延。同时,多MEC服务器之间可以相互协作,提高了用户请求在边缘缓存的命中率,从而降低了缓存命中成本。从图8中还可以看出,FDSAC算法在不同MEC服务器数量下的系统综合成本低于其他3种对比算法。这是由于FDSAC算法采用了加权FL的框架,各个MEC服务器通过协作训练模型参数,加快了模型的收敛速度,使MEC服务器能够快速做出缓存决策,从而降低系统综合成本。

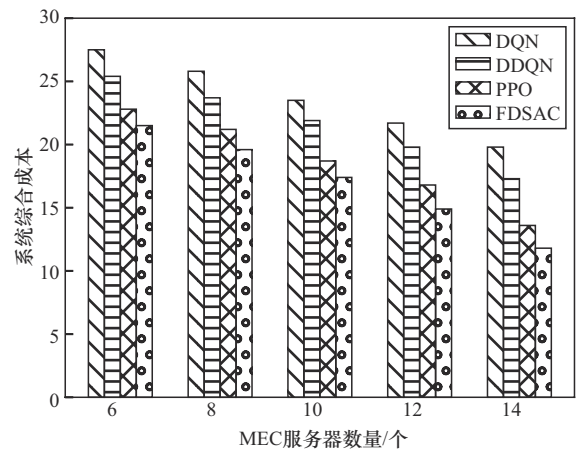


图8 不同MEC服务器数量下的系统综合成本

图9展示了不同MEC服务器缓存大小下的系统综合成本。随着MEC服务器缓存大小的增加,

意味着 MEC 服务器可以缓存更多内容目录中的内容。从图 9 中可以看出, 随着 MEC 服务器缓存大小的增加, 4 种算法下的系统综合成本在不断降低。特别是与其他 3 种对比算法相比较, FDSAC 算法的系统综合成本下降的速度更快。这是由于 FDSAC 算法能够充分利用 MEC 服务器的缓存大小做出最优的缓存替换策略, 从而最大化网络的资源利用率以减少系统综合成本。此外, 从图 9 中还可以看出, FDSAC 算法下的系统综合成本要远小于其他 3 种对比算法, 这表明 FDSAC 算法所做出的缓存决策要优于其他 3 种对比算法。

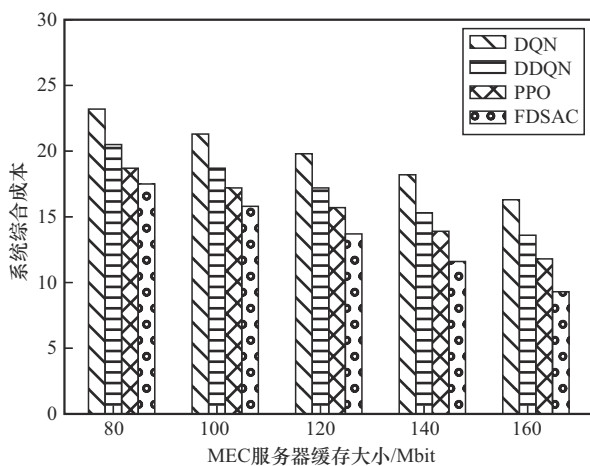


图9 不同 MEC 服务器缓存大小下的系统综合成本

6 结束语

本文研究了动态且复杂的空天地一体化 MEC 网络中内容请求的缓存替换策略。首先, 提出了一种空天地一体化边缘协作缓存的网络架构, 并刻画了 MEC 服务器的协作缓存模型和内容访问时延模型。其次, 将边缘服务器的缓存替换问题建模为马尔可夫决策过程, 以最大化长期负奖励的期望作为优化目标去学习最优的缓存替换策略。然后, 设计了一种加权注意力机制来聚合 FL 框架中模型参数的权重, 并将 BiLSTM 集成到 DSAC 模型中实现了对内容流行度更好的预测。最后, 提出了一种 FDSAC 算法以做出最优的缓存替换策略。仿真结果表明, 本文算法与其他经典的 DRL 算法相比, 在提高边缘缓存命中率、降低内容访问时延方面具有更好的性能, 为未来 6G 空天地一体化网络中边缘协作缓存方案的设计提供了值得借鉴的参考。

参考文献:

- [1] NAWAZ F, MOHSIN A, JANJUA N K. Service description languages in cloud computing: state-of-the-art and research issues[J]. *Service Oriented Computing and Applications*, 2019, 13(2): 109-125.
- [2] 黄永明, 郑冲, 张征明, 等. 大规模无线通信网络移动边缘计算和缓存研究[J]. *通信学报*, 2021, 42(4): 44-61.
HUANG Y M, ZHENG C, ZHANG Z M, et al. Research on mobile edge computing and caching in massive wireless communication network[J]. *Journal on Communications*, 2021, 42(4): 44-61.
- [3] MASOOD A, TUAN D Q, LAKEW D S, et al. A review on AI-enabled content caching in vehicular edge caching and networks[C]//Proceedings of the 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIC). Piscataway: IEEE Press, 2023: 713-717.
- [4] 张海君, 陈安琪, 李亚博, 等. 6G 移动网络关键技术[J]. *通信学报*, 2022, 43(7): 189-202.
ZHANG H J, CHEN A Q, LI Y B, et al. Key technologies of 6G mobile network[J]. *Journal on Communications*, 2022, 43(7): 189-202.
- [5] ZHOU J W, CHEN F F, HE Q, et al. Data caching optimization with fairness in mobile edge computing[J]. *IEEE Transactions on Services Computing*, 2023, 16(3): 1750-1762.
- [6] ZHANG S B, LIU J J. Optimal probabilistic caching in heterogeneous IoT networks[J]. *IEEE Internet of Things Journal*, 2020, 7(4): 3404-3414.
- [7] YOO S, JEONG S, KIM J, et al. Cache-assisted mobile-edge computing over space-air-ground integrated networks for extended reality applications[J]. *IEEE Internet of Things Journal*, 2024, 11(10): 18306-18319.
- [8] CHEN S W, YAO Z, JIANG X F, et al. Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks[J]. *IEEE Transactions on Communications*, 2021, 69(4): 2441-2456.
- [9] WU H D, NASEHZADEH A, WANG P. A deep reinforcement learning-based caching strategy for IoT networks with transient data[J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(12): 13310-13319.
- [10] LIANG Q F, LIU Y, TANG W J. Joint cache placement and content scheduling in integrated LEO satellite-terrestrial networks[C]//Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2022: 642-648.
- [11] ZHANG H J, XU J L, LIU X N, et al. Joint optimization of caching placement and power allocation in virtualized satellite-terrestrial network[J]. *IEEE Transactions on Wireless Communications*, 2023, 22(11): 7932-7943.
- [12] JIANG C X, LI Z. Decreasing big data application latency in satellite link by caching and peer selection[J]. *IEEE Transactions on Network Science and Engineering*, 2020, 7(4): 2555-2565.
- [13] AN K, LI Y S, YAN X J, et al. On the performance of cache-enabled hybrid satellite-terrestrial relay networks[J]. *IEEE Wireless Communications Letters*, 2019, 8(5): 1506-1509.
- [14] ZHU X M, JIANG C X, KUANG L L, et al. Cooperative multilayer edge caching in integrated satellite-terrestrial networks[J]. *IEEE Transactions on Wireless Communications*, 2022, 21(5): 2924-2937.
- [15] JI J Q, ZHU K, NIYATO D, et al. Probabilistic cache placement in UAV-assisted networks with D2D connections: performance analysis and trajectory optimization[J]. *IEEE Transactions on Communications*,

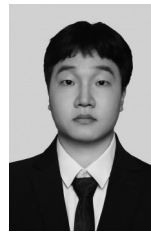
- 2020, 68(10): 6331-6345.
- [16] ZHANG M Z, EI-HAJJAR M, NG S X. Intelligent caching in UAV-aided networks[J]. IEEE Transactions on Vehicular Technology, 2022, 71(1): 739-752.
- [17] BERA A, MISRA S, CHATTERJEE C. QoE analysis in cache-enabled multi-UAV networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(6): 6680-6687.
- [18] ANOKYE S, AYEPAH-MENSAH D, SEID A M, et al. Deep reinforcement learning-based mobility-aware UAV content caching and placement in mobile edge networks[J]. IEEE Systems Journal, 2022, 16(1): 275-286.
- [19] WU Y, LI J D, LIU J Y, et al. Enabling efficient caching in high mobility UAV communications network under limited backhaul[J]. China Communications, 2022, 19(10): 207-219.
- [20] GU S S, WANG Y, WANG N N, et al. Intelligent optimization of availability and communication cost in satellite-UAV mobile edge caching system with fault-tolerant codes[J]. IEEE Transactions on Cognitive Communications and Networking, 2020, 6(4): 1230-1241.
- [21] BAO J R, PENG X Y, LIU C, et al. Multilayered decentralized coded caching with nonuniform popularity and multilevel cache capacity in space-air-ground integrated networks[J]. IEEE Internet of Things Journal, 2024, 11(8): 13913-13926.
- [22] DU W R, WEI Z X, LI C, et al. On a deep reinforcement learning-based content caching strategy in 6G space-air-ground integrated networks[C]//Proceedings of the 2023 15th International Conference on Communication Software and Networks (ICCSN). Piscataway: IEEE Press, 2023: 204-208.
- [23] GU S S, SUN X Y, YANG Z H, et al. Energy-aware coded caching strategy design with resource optimization for satellite-UAV-vehicle-integrated networks[J]. IEEE Internet of Things Journal, 2022, 9(8): 5799-5811.
- [24] YIN F F, LIU Q H, LIU D P, et al. Joint coded caching and resource allocation for multimedia service in space-air-ground integrated networks[J]. IEEE Transactions on Communications, 2024, 72(11): 6839-6853.
- [25] TANG F X, HOFNER H, KATO N, et al. A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(1): 276-289.
- [26] DENG R Q, DI B Y, CHEN S Z, et al. Ultra-dense LEO satellite offloading for terrestrial networks: how much to pay the satellite operator?[J]. IEEE Transactions on Wireless Communications, 2020, 19(10): 6240-6254.
- [27] GAN Y H, HE Y J. Trajectory optimization and computing offloading strategy in UAV-assisted MEC system[C]//Proceedings of the 2021 Computing, Communications and IoT Applications (ComComAp). Piscataway: IEEE Press, 2021: 132-137.
- [28] WANG X F, WANG C Y, LI X H, et al. Federated deep reinforcement learning for Internet of things with decentralized cooperative edge caching[J]. IEEE Internet of Things Journal, 2020, 7(10): 9441-9455.
- [29] KHAMARI S, RACHEDI A, AHMED T, et al. Adaptive deep reinforcement learning approach for service migration in MEC-enabled vehicular networks[C]//Proceedings of the 2023 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2023: 1075-1079.
- [30] LYU Z W, WANG Y, LIU M, et al. Service-driven resource management in vehicular networks based on deep reinforcement learning[C]//

Proceedings of the 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications. Piscataway: IEEE Press, 2020: 1-6.

[作者简介]



刘亮 (1979-), 男, 重庆人, 博士, 重庆邮电大学副教授, 主要研究方向为空天地一体化网络、边缘智能计算等。



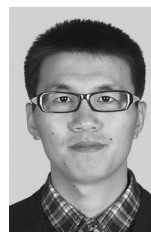
荆腾祥 (2000-), 男, 河南焦作人, 重庆邮电大学硕士生, 主要研究方向为空天地一体化网络、机器学习等。



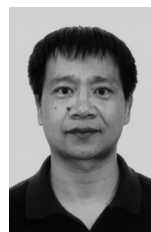
段洁 (1982-), 女, 四川内江人, 博士, 重庆邮电大学副教授, 主要研究方向为空天地一体化网络、网络路由、网络优化等。



毛武平 (1998-), 男, 江西吉安人, 重庆邮电大学硕士生, 主要研究方向为空天地一体化网络、移动边缘计算等。



燕洪成 (1985-), 男, 山东淄博人, 中国空间技术研究院高级工程师, 主要研究方向为航天器电子信息系统。



马文杰 (1973-), 男, 山东高密人, 中国空间技术研究院高级工程师, 主要研究方向为卫星总体及星座总体技术。