

# 基于深度强化学习的海洋移动边缘计算卸载方法

苏新<sup>1</sup>, 孟蕾蕾<sup>1</sup>, 周一青<sup>2,3</sup>, CELIMUGE Wu<sup>4</sup>

(1. 河海大学物联网工程学院, 江苏 常州 231022;

2. 中国科学院计算技术研究所处理器芯片全国重点实验室, 北京 100190;

3. 中国科学院大学计算机科学与技术学院, 北京 100190; 4. 日本电气通信大学信息理工学, 东京 182-8585)

**摘要:** 海洋信息系统网络节点之间的强异构特性为海洋移动边缘计算任务卸载优化带来了复杂高维度的限制条件, 同时复杂多样化的海事应用会导致海洋网络局部区域出现计算任务的超负荷处理。为实现海洋网络节点计算任务的最佳卸载与资源优化, 满足网络低时延、高可靠的应用服务需求, 提出基于多尺度异构特征属性的海洋网络节点分层归类方法和基于深度强化学习的海洋移动边缘计算卸载方法。仿真结果表明, 所提方法较传统方法能够在海洋信息系统下有效地降低网络节点的计算任务卸载时延, 并且能够在大规模任务流下保持海洋网络的稳健性。

**关键词:** 海洋信息系统; 边缘计算; 计算任务卸载; 功率与计算资源分配; 深度强化学习

中图分类号: TN929.52

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022197

## Maritime mobile edge computing offloading method based on deep reinforcement learning

SU Xin<sup>1</sup>, MENG Leilei<sup>1</sup>, ZHOU Yiqing<sup>2,3</sup>, CELIMUGE Wu<sup>4</sup>

1. The College of IoT Engineering, Hohai University, Changzhou 231022, China

2. State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

3. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China

4. Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan

**Abstract:** The strong heterogeneity among the network nodes of the maritime information system brings complex and high-dimensional constraints for optimizing task offloading of the maritime mobile edge computing. The complex and diverse maritime applications also lead to the overload processing of computing tasks in local areas of the maritime network. In order to optimize the task offloading and resource management of maritime network, as well as meet the maritime application service requirements of low-latency and high-reliability, a hierarchical classification method of maritime nodes based on multi-layers attributes and a novel offloading method for maritime mobile edge computing based on deep reinforcement learning were proposed. Compared with conventional methods, simulation results show that the proposed method can effectively reduce the computing task offloading delay of the marine information system, and maintain the robustness of the maritime network with large-scale task flows.

**Keywords:** maritime information system, edge computing, computing task offloading, power and computing resource allocation, deep reinforcement learning

## 0 引言

国家“十四五”规划明确提出要积极拓展海洋经济发展空间, 协同推进海洋生态保护、海洋经济

发展和海洋权益维护, 加快海洋强国建设。当前, 国家正在加大开发利用海洋多样化数字资源的基础建设投入, 构建具有自主核心技术的下一代海洋信息系统, 围绕海洋工程、资源、环境等领域攻克

收稿日期: 2022-06-13; 修回日期: 2022-09-10

基金项目: 国家重点研发计划基金资助项目 (No.2021YFE0105500)

Foundation Item: The National Key Research and Development Program of China (No.2021YFE0105500)

一批技术难题,突破海洋霸权强国的封锁,使我国全天候、全自动的海洋观测活动稳扎深蓝远海<sup>[1]</sup>。

移动边缘计算作为 5G 关键技术之一,将其应用于下一代海洋信息系统有望提供各类海洋感知数据的实时处理能力,满足高可靠、低时延海事应用的快速响应需求<sup>[2-3]</sup>。研究表明,基于移动边缘计算的海洋信息实时处理性能相比传统方式可以极大提高处理效率<sup>[4]</sup>。面向下一代海洋信息系统建设,该技术具有极高的研究意义和应用价值。

计算任务的有效卸载作为移动边缘计算的核心功能,已在陆地蜂窝网和车联网等应用场景下展开了深入研究和广泛应用<sup>[5-6]</sup>。然而,与传统陆地移动边缘计算任务卸载不同,制定海洋环境下的计算任务卸载策略将会面临全新的技术挑战。

1) 海洋网络节点离散、分布不均,计算能力与能耗敏感度存在差异,海洋网络各节点之间的强异构特性为计算任务卸载优化带来了复杂高维度的限制条件。

2) 复杂多样化的海事应用会导致海洋网络局部区域出现计算任务的超负荷处理,实现海洋网络节点计算任务的最佳卸载与资源分配是保障海事应用服务性能的关键。

通过调研已有计算任务卸载方法,结合海洋环境下的计算任务卸载技术挑战,以满足高可靠、低时延的海事应用服务为目标,本文提出了基于深度强化学习的计算任务卸载方法,联合优化海洋网络节点卸载决策与资源分配,主要学术贡献包括以下几个方面。

1) 针对海洋节点强异构下的卸载问题,刻画全新海洋信息系统下的计算任务卸载场景,利用海洋网络节点的计算能力、传输能力、可分配存储能力等多种异构特征属性对海洋网络节点进行归类分层。

2) 提出面向下一代海洋信息系统的多节点卸载模型,以最小化海洋网络节点计算任务执行时延为目标,利用人工智能与凸优化等手段,联合优化海洋网络节点计算任务卸载决策与资源分配问题,提升系统服务性能。

## 1 相关工作

欧洲电信标准化协会于 2016 年将移动边缘计算的概念扩展为多接入边缘计算(MEC, multi-access edge computing),并扩大了其研究范围,包括 LTE、固定宽带和 Wi-Fi 技术的异构组网模式<sup>[7]</sup>。MEC 技术优势主要体现在超低时延、高带宽、高可靠性以

及可扩展性等方面,可以辅助运营商灵活快速地向移动用户和企业部署新的服务。MEC 可利用计算任务卸载实现系统分布式计算框架下的边缘计算节点负载均衡,通过适配网络节点计算能力和与其对应的数据负载,实现边缘计算节点的同步处理,获取最短的应用服务时延。

面向陆地蜂窝网络,MEC 可以通过计算任务的卸载大幅降低网络的回程流量、传输成本和数据泄露风险。文献[8]研究了单服务器场景下的多用户计算卸载问题,通过一种基于软件定义网络的 Stackelberg 博弈模型,实现了任务最佳卸载。但是,文献[8]中模型计算复杂度高,不适用于任务实时卸载的相关应用场景。文献[9]面向任务实时卸载应用场景,将卸载问题转化为二阶锥规划问题,并利用一种基于逐次凸逼近的算法进行迭代求解,降低了求解方法的复杂度。然而,文献[9]尚未考虑边缘计算节点的动态移动性,导致其适用性较弱。文献[10]研究了 MEC 系统中时延和能耗之间的基本权衡,提出了一种基于迭代启发式的在线卸载算法,获得了较好的服务性能。文献[11]将计算卸载决策表示为有限时间的马尔可夫决策过程,并采用动态规划方法求解最优卸载决策,实现了最小化系统时延的目标。然而,文献[11]未考虑能耗约束会导致用户能耗过高。为同时满足用户对时延与能耗的要求,文献[12]针对蜂窝网络中边缘网络与中心云计算之间的协同卸载问题,提出了一种基于分解法和连续伪凸法的框架,并通过迭代算法降低了系统成本。

面向车联网等时延敏感型应用,MEC 技术优势主要体现在降低网络时延、提高用户的体验质量。文献[13]设计了一种基于预测卸载的传输机制,将任务提前传输至 MEC 服务器,一旦车辆进入服务器覆盖范围便可立即响应,实现了降低车辆计算任务执行时延的目标。但是,文献[13]忽略了节点负载均衡,导致部分服务器由于过载而限制其性能的提升。文献[14]则将负载均衡与卸载相结合,研究了多用户多服务器系统的资源分配问题,提升了系统性能。然而,文献[14]假设车辆匀速移动,不符合实际车联网的应用场景。针对车辆移动性问题,文献[15]研究了车辆随机移动场景下的 MEC 系统性能,提出了相应的移动感知卸载机制,降低了系统运维成本。但文献[15]采用了单跳的任务卸载方式,导致其系统模型对环境的适应性较弱。在此基础上,文献[16]研究了针对多跳车辆场景的有效联

合卸载，并制定了双端优化问题，联合优化了用户端和服务端的服务成本。但是，文献[16]仅考虑了正交多址应用场景下的车载 MEC 系统，导致其系统模型过于单一。文献[17]进一步考虑了正交多址和非正交多址 2 种应用场景下的车载 MEC 系统，提出了基于启发式算法的任务卸载机制，获得了系统最小化任务总执行时延。

移动边缘计算卸载场景实际上存在动态、随机、时变等特性，而上述面向陆地蜂窝网和车联网的任务卸载方案的主要目的是得到更好的卸载决策，缺乏对实际场景下决策算法高实时性要求的考虑。随着人工智能技术的兴起，研究者逐步将其与移动边缘计算技术进行有效结合，可以更好地解决动态、随机、时变环境下的计算任务卸载优化问题。

文献[18]研究了车联网中的任务卸载和资源分配问题，并提出了一种基于 Q-Learning 的任务卸载和资源分配算法，但是文献[18]仅考虑了单一的 MEC 系统。文献[19]针对蜂窝网中的多个 MEC 系统，提出了一种基于 Q-Learning 的联合通信和计算资源分配机制，优化了满足能量约束条件下的任务处理时延，并通过仿真实验验证了所提方法具有较好的环境适应性。文献[20]面向动态卸载场景，利用一种基于软件定义网络（SDN, software defined network）边缘云的新型 Q-Learning 优化框架制定卸载决策和资源分配，能够快速适应梯度更新和样本数量较少的全新通信环境。虽然文献[18-20]利用 Q-Learning 方法在特定场景下达到了较好的卸载效果，但是当优化问题的状态和动作空间过大且高维连续时，存储  $Q$  值的内存空间将呈指数级增长，在搜寻最优卸载决策时也会产生大量的时间开销。为此，研究者利用深度学习技术解决了传统强化学习存在状态空间中的高维问题。文献[21]考虑了一个包括陆基、车辆以及无人机的混合移动边缘计算平台，以最小化终端设备能耗为目标，提出了一种基于深度学习的混合在线卸载算法框架，但是每次仅限输入一个设备信息，不适用于实际应用场景。文献[22]研究了车辆-车辆（V2V, vehicle to vehicle）与车辆-基础设施（V2I, vehicle to infrastructure）之间的联合卸载问题，提出了一种多智能体深度强化学习框架，达到了同时满足 V2I 和 V2V 链路时延要求的目标，但是该方法收敛速度较慢。文献[23]考虑了多用户和多服务器的 MEC 场景中的任务卸载决策问题，提出了一种基于深度强化学习的在线

卸载算法，解决了计算任务卸载时长的优化问题。

本文聚焦下一代海洋信息系统，面向多种海事应用并行、海洋网络拓扑较大且快变等特性，提出基于深度强化学习的海洋移动边缘计算卸载方法。截至目前，研究者针对海洋移动边缘计算已经提出了若干计算任务卸载算法，在一定程度上满足了海洋网络低时延高可靠的应用服务需求。

文献[24]分析了海洋通信时延和能耗之间的折中关系，提出了一种阶段性联合优化算法，在能量有限与时延敏感条件下优化了计算与通信资源的分配机制。文献[25]提出了一种基于海洋网络连通概率的边缘计算节点选取方法，根据海洋近岸和远岸的网络节点密度差异，建立了基于近海和远海场景的任务卸载模型，并分别利用遗传算法和粒子群优化算法进行求解。文献[26]利用混合整数非线性规划分离优化目标，有效地分配传输功率并通过改进传统人工鱼群算法制定了卸载决策。然而，上述文献均未充分考虑海洋网络节点之间的强异构特性。同时，启发式算法不仅在寻优时需要大量迭代，而且面对复杂卸载环境时算法计算能力会大幅下降，求解质量不能得到很好的保证。

表 1 通过应用场景、系统模型、卸载算法、节点异构性、实现效率、适用规模等方面对上述文献进行了对比与总结。由表 1 可知，传统 MEC 卸载算法<sup>[8-17, 24-26]</sup>计算复杂度较高，实现效率始终低于人工智能算法。基于强化学习的人工智能卸载算法<sup>[18-20]</sup>虽然能够达到较高的卸载效率，但存在维数灾难，只适用于小规模应用场景。基于深度强化学习的卸载算法<sup>[21-23]</sup>虽然能够克服维数灾难，但过于专注中心化的卸载策略，对新环境的适应性较弱。通过对比还可以发现目前针对海洋网络 MEC 研究仍然存在以下缺陷：对综合考虑端-边协同架构中的计算任务卸载和资源分配问题的研究较少，并且已有文献尚未充分考虑海洋网络节点之间的强异构特性；没有对 MEC 系统中的节点进行分层归类，在高维度时收敛速度慢，无法满足海洋监测网中低时延、高可靠的任务卸载需求，并且对快速变化的环境适应性较弱。

## 2 海洋移动边缘计算任务卸载

### 2.1 海洋网络节点分层归类

图 1 刻画了基于海洋网络的计算任务卸载模型。考虑到海洋网络节点的强异构特性，需要依据海洋网络节点特征属性对其进行归类分层，包括计

表 1 MEC 卸载模型比较

文献	应用场景	系统模型	卸载算法	是否基于人工智能	是否考虑异构性	节点是否分层	实现效率	适用规模	是否存在维数灾难
文献[8]	蜂窝网	单服务器	Stackelberg 博弈	否	否	否	低	小	是
文献[9]	蜂窝网	多服务器	逐次逼近	否	是	否	低	小	是
文献[10]	蜂窝网	单服务器	启发式	否	否	否	低	小	是
文献[11]	蜂窝网	单服务器	动态规划	否	否	否	较低	小	是
文献[12]	蜂窝网	多服务器	迭代算法	否	是	否	较低	小	是
文献[13]	车联网	多服务器	预测卸载	否	是	否	低	小	是
文献[14]	车联网	多服务器	启发式	否	是	否	较低	小	是
文献[15]	车联网	多服务器	移动感知	否	是	否	低	小	是
文献[16]	车联网	多服务器	Lyapunov 优化	否	是	否	较低	小	是
文献[17]	车联网	多服务器	启发式	否	否	否	较低	小	是
文献[18]	车联网	单服务器	Q-Learning	是	否	否	较高	小	是
文献[19]	蜂窝网	多服务器	Q-Learning	是	否	否	较高	小	是
文献[20]	蜂窝网	多服务器	Q-Learning, SDN	是	是	否	较高	小	是
文献[21]	混合 MEC 网络	多服务器	DNN	是	是	否	较高	大	否
文献[22]	车联网	多服务器	DRL	是	是	否	高	大	否
文献[23]	蜂窝网	多服务器	DRL	是	是	否	高	大	否
文献[24]	海洋网络	单服务器	启发式	否	否	否	低	小	是
文献[25]	海洋网络	多服务器	粒子群优化、遗传算法	否	是	否	低	小	是
文献[26]	海洋网络	单服务器	人工鱼群算法	否	是	否	低	小	是

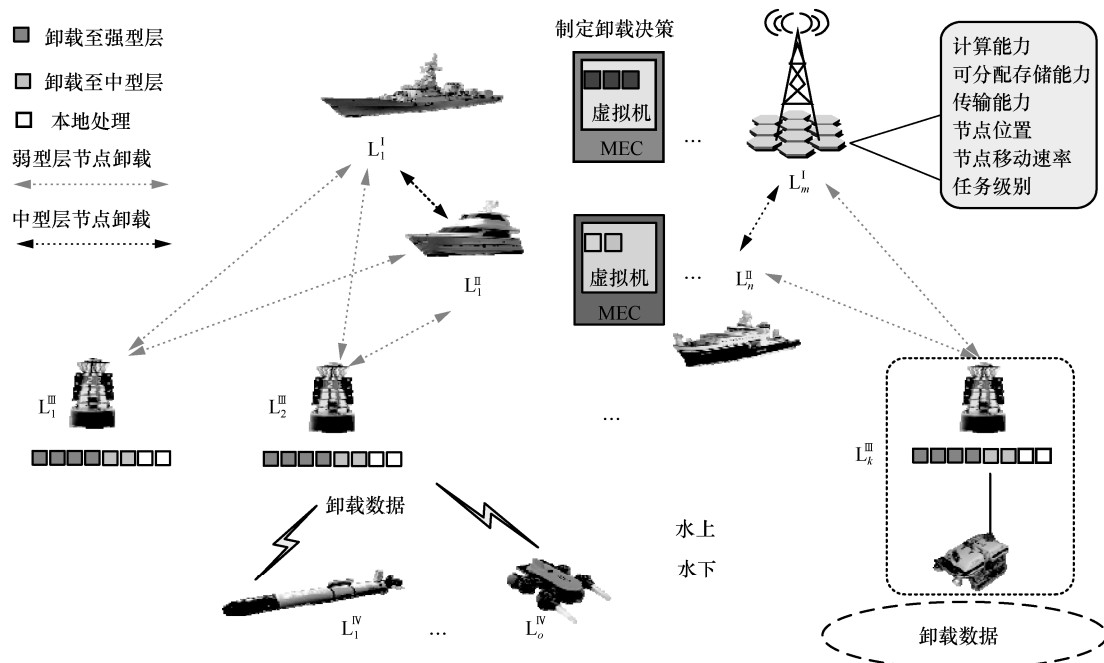


图 1 基于海洋网络的计算任务卸载模型

算能力、可分配存储能力、传输能力、节点位置以及节点移动速率；并且可将节点归类划分为强型层、中型层、弱型层和独立层，分别用  $L_m^I$ 、 $L_n^{II}$ 、 $L_k^{III}$  和  $L_o^{IV}$  表示，其中， $m \in \{1, 2, \dots, M\}$ ， $n \in \{1, 2, \dots, N\}$ ， $k \in \{1, 2, \dots, K\}$ ， $o \in \{1, 2, \dots, O\}$ 。具体分层归类策略如下。

强型层节点移动速率相对缓慢稳定，计算能力和可分配存储能力强、电量充沛，可有效快速地地完成海量复杂的计算任务，主要由海岸基站和大型船舶等组成。中型层节点移动速率较快，计算能力和可分配存储能力较强，电量较充足，可处理复杂度一般的海事任务，主要由柴油驱动的中型船舶组

成。弱型层节点计算能力和可分配存储能力弱，蓄电池电量受限，可处理普通的海事任务，主要由电力驱动的小型船舶以及海上浮标等组成。部分海上浮标可通过有线连接浅海区域水下机器人展开相关业务，因此将其视为一个节点整体。独立层节点主要位于水下，通过水声通信方式互联。若进行大规模的数据卸载，会产生较大时延，因此本文不考虑水下卸载过程，水下任务只在本地处理，该层节点独立于其他层的节点。

如图 1 所示，同层节点性能相似，相互之间无须卸载计算任务。那么，本文设定基于海洋网络的计算任务卸载规律可描述为：弱型层 ( $L_k^{\text{III}}$ )  $\rightarrow$  中型层 ( $L_n^{\text{II}}$ )  $\rightarrow$  强型层 ( $L_m^{\text{I}}$ ) 和  $L_k^{\text{III}} \rightarrow L_m^{\text{I}}$ ，且不可反向卸载。强型层  $L_m^{\text{I}}$  节点具有较强的计算能力，可将其设置为最终制定卸载决策的计算中心节点。同时，当  $L_m^{\text{I}}$  和  $L_n^{\text{II}}$  均在  $L_k^{\text{III}}$  节点的通信范围内时，可以优先选择  $L_m^{\text{I}}$  作为候选卸载节点；当仅有  $L_n^{\text{II}}$  在  $L_k^{\text{III}}$  节点的通信范围内时，则可选择  $L_n^{\text{II}}$  作为候选卸载节点；当  $L_m^{\text{I}}$  和  $L_n^{\text{II}}$  均不在  $L_k^{\text{III}}$  节点的通信范围内时，则进行本地处理。

## 2.2 海洋移动边缘计算卸载时延模型

本文面向海洋网络研究了  $L_k^{\text{III}}$  节点的卸载过程，具体地，每个  $L_k^{\text{III}}$  节点可向  $L_m^{\text{I}}$  节点或  $L_n^{\text{II}}$  节点卸载海事应用任务，并通过二元组  $D_k=(d_k, c_k)$  表示其不可分割的每个海事应用子任务，其中， $d_k$  为完成计算任务所需的输入数据大小， $c_k$  为完成计算任务所需的 CPU 周期数。

$L_k^{\text{III}}$  节点选择在本地执行计算任务的时延为

$$t_k^{\text{III-local}} = \frac{c_k}{f_k^{\text{III}}} \quad (1)$$

其中， $f_k^{\text{III}}$  为  $L_k^{\text{III}}$  节点的本地处理速率。

若  $L_k^{\text{III}}$  节点将计算任务卸载至强型层  $L_m^{\text{I}}$  节点进行处理，则所产生的时延包括计算任务卸载的上行链路传输时延、在 MEC 服务器端的计算任务处理时延和计算任务处理结果反馈的下行传输时延。由于处理后的计算结果远小于原始卸载数据，可以忽略计算结果反馈时延。为防止节点之间相互干扰，本文使用正交频分多址接入 (OFDMA, orthogonal frequency division multiple access) 进行通信，当每个  $L_k^{\text{III}}$  节点所分配的信道带宽为  $B_k^{\text{III}}$  时， $L_k^{\text{III}}$  节点

任务上传至  $L_m^{\text{I}}$  的数据传输速率可以表示为

$$r_{k,m}^{\text{III,I}} = B_k^{\text{III}} \text{lb} \left( 1 + \frac{P_{k,m}^{\text{III,I}} G_{k,m}^{\text{III,I}}}{N_0} \right) \quad (2)$$

其中， $P_{k,m}^{\text{III,I}}$  为  $L_k^{\text{III}}$  节点的发送功率， $G_{k,m}^{\text{III,I}}$  为信道增益， $N_0$  为高斯白噪声功率。 $L_k^{\text{III}}$  节点任务上传至  $L_m^{\text{I}}$  的传输时延为

$$t_{\text{up}(k,m)}^{\text{III,I}} = \frac{d_k}{r_{k,m}^{\text{III,I}}} \quad (3)$$

同时，计算任务在  $L_m^{\text{I}}$  上的处理时延为

$$t_{\text{exe}(k,m)}^{\text{III,I}} = \frac{c_k}{f_{k,m}} \quad (4)$$

其中， $f_{k,m}$  为  $L_m^{\text{I}}$  分配给  $L_k^{\text{III}}$  节点的处理速率。因此， $L_k^{\text{III}}$  节点将计算任务卸载至  $L_m^{\text{I}}$  节点的时延为

$$t_{k,m}^{\text{off-I}} = t_{\text{up}(k,m)}^{\text{III,I}} + t_{\text{exe}(k,m)}^{\text{III,I}} \quad (5)$$

类似  $L_k^{\text{III}}$  对  $L_m^{\text{I}}$  的卸载过程， $L_k^{\text{III}}$  将计算任务卸载至中型层  $L_n^{\text{II}}$  节点的时延可表示为

$$t_{k,m}^{\text{off-II}} = t_{\text{up}(k,m)}^{\text{III,II}} + t_{\text{exe}(k,m)}^{\text{III,II}} \quad (6)$$

由于每个  $L_k^{\text{III}}$  节点任务可选择在本地、 $L_m^{\text{I}}$  或  $L_n^{\text{II}}$  进行处理，因此， $L_k^{\text{III}}$  节点任务执行时延可表示为

$$t_k^{\text{III}} = t_k^{\text{III-local}} \alpha_k^{\text{III}} + \sum_{n=1}^N t_{k,n}^{\text{off-II}} \alpha_{k,n}^{\text{II}} + \sum_{m=1}^M t_{k,m}^{\text{off-I}} \alpha_{k,m}^{\text{I}} \quad (7)$$

其中， $\alpha_k^{\text{III}}$ 、 $\alpha_{k,n}^{\text{II}}$  和  $\alpha_{k,m}^{\text{I}}$  为二元变量，表示计算任务的卸载决策。若  $L_k^{\text{III}}$  将任务在本地执行，则  $\alpha_k^{\text{III}} = 1$ ，否则  $\alpha_k^{\text{III}} = 0$ 。若  $L_k^{\text{III}}$  将任务卸载至  $L_n^{\text{II}}$  节点，则  $\alpha_{k,n}^{\text{II}} = 1$ ，否则  $\alpha_{k,n}^{\text{II}} = 0$ 。若  $L_k^{\text{III}}$  将任务卸载至  $L_m^{\text{I}}$  节点，则  $\alpha_{k,m}^{\text{I}} = 1$ ，否则  $\alpha_{k,m}^{\text{I}} = 0$ 。

当网络中心节点根据卸载决策确定  $\alpha^{\text{I}}$ 、 $\alpha^{\text{II}}$ 、 $\alpha^{\text{III}}$ 、上行链路传输功率  $P$  以及相应的 MEC 服务器计算资源  $F$  时，执行完成所有  $L_k^{\text{III}}$  节点的计算任务所需要的总时延为

$$T(\alpha^{\text{I}}, \alpha^{\text{II}}, \alpha^{\text{III}}, P, F) = \sum_{k=1}^K \varpi_k t_k^{\text{III}} \quad (8)$$

其中， $\varpi_k \in (0, 1]$  为权重因子，用于指定资源提供者对节点的偏好，满足时延敏感性节点的需求<sup>[27]</sup>， $\varpi_k$  的值可以根据计算任务的关键性来设置。

## 2.3 海洋移动边缘计算卸载时延优化问题

依据以上分析，本文关注的海洋网络  $L_k^{\text{III}}$  节点总

的计算任务执行时延优化问题可以描述为

$$\begin{aligned}
& \text{P1} \quad \min_{\alpha^I, \alpha^{II}, \alpha^{III}, P, F} T(\alpha^I, \alpha^{II}, \alpha^{III}, P, F) \\
& \text{s.t.} \quad C_1: \alpha_k^{III}, \alpha_{k,n}^{II}, \alpha_{k,m}^I \in \{0, 1\}, \forall k, m, n \\
& \quad C_2: \alpha_k^{III} + \sum_{n=1}^N \alpha_{k,n}^{II} + \sum_{m=1}^M \alpha_{k,m}^I = 1, \forall k \\
& \quad C_3: \alpha_{k,m}^I \leq \alpha_{k,m}^{I, \max}, \forall k, m \in M \\
& \quad C_4: \alpha_{k,n}^{II} \leq \alpha_{k,n}^{II, \max}, \forall k, n \\
& \quad C_5: P_{k,m}^{III, I} \geq 0, \forall k, m \\
& \quad C_6: \sum_{m=1}^M \alpha_{k,m}^I P_{k,m}^{III, I} \leq P_{k, \max}^{III}, \forall k \\
& \quad C_7: P_{k,n}^{III, II} \geq 0, \forall k, n \\
& \quad C_8: \sum_{n=1}^N \alpha_{k,n}^{II} P_{k,n}^{III, II} \leq P_{k, \max}^{III}, \forall k \\
& \quad C_9: f_{k,m} \leq 0, \forall k, m \\
& \quad C_{10}: \sum_{k=1}^K f_{k,m} \leq F_{\max}^I, \forall m \\
& \quad C_{11}: f_{k,n} \leq 0, \forall k, n \\
& \quad C_{12}: \sum_{k=1}^K f_{k,n} \leq F_{\max}^{II}, \forall n
\end{aligned} \quad (9)$$

其中,  $P_{k, \max}^{III}$  表示  $L_k^{III}$  节点的最大传输功率,  $F_{\max}^{II}$  和  $F_{\max}^I$  分别表示  $L_n^{II}$  和  $L_m^I$  节点的最大计算能力。 $\alpha_{k,m}^{I, \max}$  和  $\alpha_{k,n}^{II, \max}$  为状态变量, 若  $L_m^I$  在  $L_k^{II}$  或  $L_k^{III}$  的传输范围内,  $\alpha_{k,m}^{I, \max}$  或  $\alpha_{k,n}^{II, \max}$  的取值为 1, 否则为 0。

式(9)的约束  $C_1$  表示任务执行方式为二进制卸载, 约束  $C_2$  表示每个任务只能选择一种执行方式, 约束  $C_3$  和  $C_4$  表示  $L_k^{III}$  节点的传输范围限制, 约束  $C_5$ 、 $C_6$ 、 $C_7$  和  $C_8$  限定节点发送功率, 约束  $C_9$ 、 $C_{10}$ 、 $C_{11}$  和  $C_{12}$  限定节点计算能力。

基于式(9)可知优化问题 P1 是一个混合整数非线性规划问题<sup>[28-29]</sup>。面向多种海事应用并行、网络拓扑大而快变的海洋计算任务卸载场景, 考虑海洋网络独有的节点强异构特性, 优化问题 P1 是一个大规模、高维度且具有诸多限制条件的复杂问题, 即使能够描述出这种海洋移动边缘计算的卸载模型及问题, 求解过程也会相当困难。为此, 本文面向海洋移动边缘计算任务卸载, 设计了一种基于凸优化-深度强化学习的计算任务卸载策略, 从而避免解决使用传统方法难以求解的问题。

### 3 基于深度强化学习的计算任务卸载方法

优化问题 P1 中  $P$  和  $F$  的最优值  $P^*$  和  $F^*$  是关于  $\alpha^I, \alpha^{II}, \alpha^{III}$  的函数, 因此问题 P1 可以简化为

$$\begin{aligned}
& \min_{\alpha^I, \alpha^{II}, \alpha^{III}} \left( \min_{P, F} T(\alpha^I, \alpha^{II}, \alpha^{III}, P, F) \right) \\
& \text{s.t.} \quad C_1 \sim C_{12}
\end{aligned} \quad (10)$$

其中, 约束  $C_1 \sim C_4$  中的  $\alpha^I, \alpha^{II}, \alpha^{III}$  与约束  $C_5 \sim C_{12}$  中的  $P, F$  相互解耦。式(10)的求解等效于求解如下的主优化问题 P2。

$$\begin{aligned}
& \text{P2} \quad \min_{\alpha^I, \alpha^{II}, \alpha^{III}} T^*(\alpha^I, \alpha^{II}, \alpha^{III}) \\
& \text{s.t.} \quad C_1 \sim C_{12}
\end{aligned} \quad (11)$$

其中,  $T^*(\alpha^I, \alpha^{II}, \alpha^{III})$  是对应  $P$  和  $F$  的最优解函数, 具体表达式为

$$\begin{aligned}
& T^*(\alpha^I, \alpha^{II}, \alpha^{III}) = \min_{P, F} T(\alpha^I, \alpha^{II}, \alpha^{III}, P, F) \\
& \text{s.t.} \quad C_5 \sim C_{12}
\end{aligned} \quad (12)$$

上述问题的分解不会改变原问题的最优解, 同时式(12)具有变量可分离的结构, 即  $P$  和  $F$  所对应的目标函数和约束彼此解耦。因此, 式(12)可以分解为 2 类子优化问题, 其中,  $F^*$  为子优化问题 P3 的最优解,  $P^*$  为子优化问题 P4 的最优解。本文针对任意卸载决策  $(\alpha^I, \alpha^{II}, \alpha^{III})$ , 利用相关算法推导出  $F^*$  和  $P^*$  的具体表达式。

$$\begin{aligned}
& \text{P3} \quad \min_{P, F} T(\alpha^I, \alpha^{II}, \alpha^{III}, P) \\
& \text{s.t.} \quad C_5 \sim C_8
\end{aligned} \quad (13)$$

$$\begin{aligned}
& \text{P4} \quad \min_{P, F} T(\alpha^I, \alpha^{II}, \alpha^{III}, F) \\
& \text{s.t.} \quad C_9 \sim C_{12}
\end{aligned} \quad (14)$$

#### 3.1 计算资源分配

$L_k^{III}$  节点向  $L_m^I$  节点进行卸载时所对应的计算资源子优化问题 P3 的具体表达式为

$$\begin{aligned}
& \min_F \sum_{m=1}^M \sum_{k=1}^K \frac{\varpi_k C_k}{f_{k,m}} \alpha_{k,m}^I \\
& \text{s.t.} \quad f_{k,m} \geq 0, \forall k, m \\
& \quad \sum_{k=1}^K f_{k,m} \leq F_{\max}^I, \forall m
\end{aligned} \quad (15)$$

式(15)约束为凸, 目标函数记为  $\Gamma(\alpha^I, F)$ , 并且  $\Gamma(\alpha^I, F)$  关于  $f_{k,m}$  的二阶导数可以表示为

$$\frac{\partial^2 \Gamma(\boldsymbol{\alpha}^1, F)}{\partial f_{k,m}^2} = \frac{2\bar{\omega}_k c_k}{f_{k,m}^3} \alpha_{k,m}^1 \geq 0, \forall k, m \quad (16)$$

由此可知, 式(15)为凸优化问题, 可以使用 KKT (Karush-Kuhn-Tucker) 条件进行求解, 其对应的拉格朗日函数可以表示为

$$L(\Gamma(s^1, F), \nu) = \sum_{m=1}^M \sum_{k=1}^K \frac{\bar{\omega}_k c_k}{f_{k,m}} \alpha_{k,m}^1 + \sum_{m=1}^M \nu_m \left( \sum_{k=1}^K f_{k,m} - F_{\max}^1 \right) \quad (17)$$

其中,  $\nu = [\nu_1, \dots, \nu_M]$  为拉格朗日乘子。计算拉格朗日函数  $L(\Gamma(s^1, F), \nu)$  关于  $f_{k,m}$  的一阶导数为

$$\frac{\partial L(\Gamma(\boldsymbol{\alpha}^1, F), \nu)}{\partial f_{k,m}} = -\frac{\bar{\omega}_k c_k}{f_{k,m}^2} \alpha_{k,m}^1 + \nu_m, \forall k, m \quad (18)$$

$$\text{令 } \frac{\partial L(\Gamma(\boldsymbol{\alpha}^1, F), \nu)}{\partial f_{k,m}} = 0, \text{ 可得计算资源问题最}$$

优解为

$$(f_{k,m})^* = \sqrt{\frac{\bar{\omega}_k c_k \alpha_{k,m}^1}{\nu_m}}, \forall k, m \quad (19)$$

当  $\nu_m^* > 0$  时, 有式(20)成立

$$\sum_{k=1}^K (f_{k,m})^* = F_{\max}^1 \quad (20)$$

将式(18)代入式(19)可以得到拉格朗日乘子的具体表达式为

$$\nu_m^* = \left( \frac{\sum_{k=1}^K \sqrt{\bar{\omega}_k c_k \alpha_{k,m}^1}}{F_{\max}^1} \right)^2, \forall m \quad (21)$$

最后, 将式(21)代入式(19), 便可以得到式(15)的最优解为

$$(f_{k,m})^* = \frac{F_{\max}^1 \sqrt{\bar{\omega}_k c_k \alpha_{k,m}^1}}{\sum_{k=1}^K \sqrt{\bar{\omega}_k c_k \alpha_{k,m}^1}}, \forall k, m \quad (22)$$

同理, 可以求出  $L_k^{\text{III}}$  节点向  $L_n^{\text{II}}$  节点进行卸载时所对应的计算资源子优化问题 P3 的最优解为

$$(f_{k,n})^* = \frac{F_{\max}^{\text{II}} \sqrt{\bar{\omega}_k c_k \alpha_{k,n}^{\text{II}}}}{\sum_{k=1}^K \sqrt{\bar{\omega}_k c_k \alpha_{k,n}^{\text{II}}}}, \forall k, n \quad (23)$$

### 3.2 数据传输功率分配

$L_k^{\text{III}}$  节点向  $L_m^{\text{I}}$  节点进行卸载时所对应的数据传输功率子优化问题 P4 的具体表达式为

$$\min_P R(\gamma, P) = \sum_{m=1}^M \sum_{k=1}^K \frac{\bar{\omega}_k d_k}{B_k^{\text{III}} \text{lb} \left( 1 + \frac{P_{k,m}^{\text{III,I}} G_{k,m}^{\text{III,I}}}{N_0} \right)} \alpha_{k,m}^{\text{I}} \quad (24)$$

$$\text{s.t. } P_{k,m}^{\text{III,I}} \geq 0, \forall k, m$$

$$\sum_{m=1}^M s_{k,m}^{\text{I}} P_{k,m}^{\text{III,I}} \leq P_{k,\max}^{\text{III}}, \forall k \quad (24)$$

由于每个  $L_k^{\text{III}}$  节点的数据传输功率是相互独立的, 因此可对每个  $L_k^{\text{III}}$  节点的数据传输功率进行单独优化。此外, 每个  $L_k^{\text{III}}$  节点只能由  $L_m^{\text{I}}$  或  $L_n^{\text{II}}$  中的一个有效节点进行服务, 即满足  $\sum_{m=1}^M \alpha_{k,m}^{\text{I}} \leq 1$ 。当  $\alpha_{k,m}^{\text{I}} = 1$  时, 式(24)可进一步化简为

$$\min_P R(\gamma, P) = \frac{\bar{\omega}_k d_k}{B_k^{\text{III}} \text{lb} \left( 1 + \frac{P_{k,m}^{\text{III,I}} G_{k,m}^{\text{III,I}}}{N_0} \right)} \quad (25)$$

$$\text{s.t. } 0 \leq P_{k,m}^{\text{III,I}} \leq P_{k,\max}^{\text{III}}, \forall k, m$$

上述优化问题中的目标函数在定义域内是严格单调递减的, 故最优解为  $p_{k,m}^* = P_{k,\max}^{\text{III}}$ 。考虑到卸载决策  $\alpha_{k,m}^{\text{I}}$  的取值为 0 或 1, 故  $p_{k,m}^*$  的统一表达式为  $p_{k,m}^* = \alpha_{k,m}^{\text{I}} P_{k,\max}^{\text{III}}$ 。同理, 可以求出  $L_k^{\text{III}}$  节点向  $L_n^{\text{II}}$  节点进行卸载时所对应的数据传输功率子优化问题 P4 的最优解为

$$p_{k,n}^* = \alpha_{k,n}^{\text{II}} P_{k,\max}^{\text{III}} \quad (26)$$

### 3.3 卸载决策生成

当确定 P3 和 P4 的最优解后, 将其代入式(12)可以得到  $T^*(\boldsymbol{\alpha}^{\text{I}}, \boldsymbol{\alpha}^{\text{II}}, \boldsymbol{\alpha}^{\text{III}})$  的具体表达式为

$$T^*(\boldsymbol{\alpha}^{\text{I}}, \boldsymbol{\alpha}^{\text{II}}, \boldsymbol{\alpha}^{\text{III}}) = \sum_{k=1}^K \frac{\bar{\omega}_k d_k}{f_{k,m}^{\text{III}}} \alpha_{k,m}^{\text{III}} + \Gamma(s^{\text{II}}, F^*) + \Gamma(\boldsymbol{\alpha}^{\text{I}}, F^*) + R(\boldsymbol{\alpha}^{\text{II}}, P^*) + R(\boldsymbol{\alpha}^{\text{I}}, P^*) \quad (27)$$

将式(27)代入式(11), 进而主优化问题 P2 可以更新为

$$\begin{aligned} \min_{\boldsymbol{\alpha}^{\text{I}}, \boldsymbol{\alpha}^{\text{II}}, \boldsymbol{\alpha}^{\text{III}}} T^*(\boldsymbol{\alpha}^{\text{I}}, \boldsymbol{\alpha}^{\text{II}}, \boldsymbol{\alpha}^{\text{III}}) \\ \text{s.t. } \alpha_k^{\text{III}}, \alpha_{k,n}^{\text{II}}, \alpha_{k,m}^{\text{I}} \in \{0, 1\}, \forall k, m, n \\ \alpha_k^{\text{III}} + \sum_{n=1}^N \alpha_{k,n}^{\text{II}} + \sum_{m=1}^M \alpha_{k,m}^{\text{I}} = 1, \forall k \\ \alpha_{k,m}^{\text{I}} \leq \alpha_{k,m}^{\text{I,max}}, \alpha_{k,n}^{\text{II}} \leq \alpha_{k,n}^{\text{II,max}}, \forall k, m, n \end{aligned} \quad (28)$$

通过分析可知式(28)为整数规划问题。由于传统算法求解整数规划问题存在维数灾难, 本文针对上述优化问题提出一种基于深度强化学习并行计算在线卸载 (OOPC-DRL, online offloading of par-

allel computing based on deep reinforcement learning) 算法, 可以规范高效地求解卸载策略, 以最小化  $L_k^m$  节点计算任务执行总时延。

OOPC-DRL 计算任务卸载策略架构如图 2 所示, 从图 2 可知, OOPC-DRL 主要由生成卸载决策矩阵集、资源分配、生成最佳卸载决策矩阵和更新卸载经验 4 个阶段交替完成。生成卸载决策矩阵集阶段主要依赖于并行的深度神经网络 (DNN, deep neural network), DNN 根据当前时刻状态实时生成  $\Omega$  个卸载决策矩阵, 并且通过经验存储器在线学习定期更新参数。其中, 当前时刻状态  $s_t$  包括海事应用任务数据量的大小  $D_k$ 、 $L_k^m$  节点的相关参数集合  $\{\alpha_{k,m}^{I,max}, \alpha_{k,m}^{II,max}, f_k^{III}, P_{k,max}^{III}\}$ 、 $L_n^I$  和  $L_m^I$  节点的相关参数集合  $\{F_{max}^{II}, F_{max}^I\}$ , 以及系统总带宽、信道增益和背景噪声功率。

资源分配阶段将生成卸载决策矩阵集阶段所生成的决策矩阵分别代入子优化问题 P3 和 P4 求解其最优计算资源和数据传输功率, 实现系统的最佳资源分配。生成最佳卸载决策矩阵阶段将前 2 个阶段所产生的卸载决策矩阵集、最优计算资源和最优数据传输功率代入主优化问题 P2, 通过求解 P2 生成最佳卸载决策矩阵。更新卸载经验阶段使用经验回放机制<sup>[30]</sup>, 将同一时刻的状态与最佳卸载决策矩阵合并作为卸载经验存储于经验存储器中, 并在训练时随机采样作为训练样本。与使用整个数据样本集相比, OOPC-DRL 算法降低了更新复杂度; 通过降低训练样本之间的相

关性, 加快了网络收敛速度; 重复使用历史数据, 减少了迭代更新方差。此时, 可以使用 Adam 随机梯度下降优化算法更新 DNN 中各网络参数<sup>[31]</sup>, 因此, 平均交叉熵损失函数表示可以为

$$C(\theta_\Omega) = -\frac{1}{|\omega|} \sum_t \left( (\alpha_t^*)^T \log f_{\theta_{\Omega}}(s_t) - (\mathbf{1} - \alpha_t^*)^T \log (1 - f_{\theta_{\Omega}}(s_t)) \right) \quad (29)$$

其中,  $|\omega|$  表示随机采样的样本数据批量大小。

OOPC-DRL 算法具体表述如算法 1 所示。

**算法 1** OOPC-DRL 算法

输入 神经网络参数  $\theta_\Omega$ , 经验存储器  $B$ , 当前状态  $s_t$

输出 卸载决策矩阵  $(\alpha^I, \alpha^{II}, \alpha^{III})$ , 计算资源  $F^*$ , 数据传输功率  $P^*$

初始化 神经网络参数  $\theta_\Omega \sim N(0,1)$ , 经验存储器的大小

- 1) for  $t=1:T$
- 2) 将当前时刻状态  $s_t$  进行归一化
- 3) DNN 根据归一化的  $s_t$  产生  $\Omega$  个候选动作矩阵, 并存入候选动作矩阵集  $C$
- 4) 代入资源分配子优化问题最优解, 即式(22)、式(23)、式(26), 得出计算资源和数据传输功率最优解  $F^*$ 、 $P^*$
- 5) 代入式(27)求解  $T^*(s_t, \alpha_t)$
- 6) 生成最佳卸载决策矩阵  $\alpha_t^* = \arg \min$

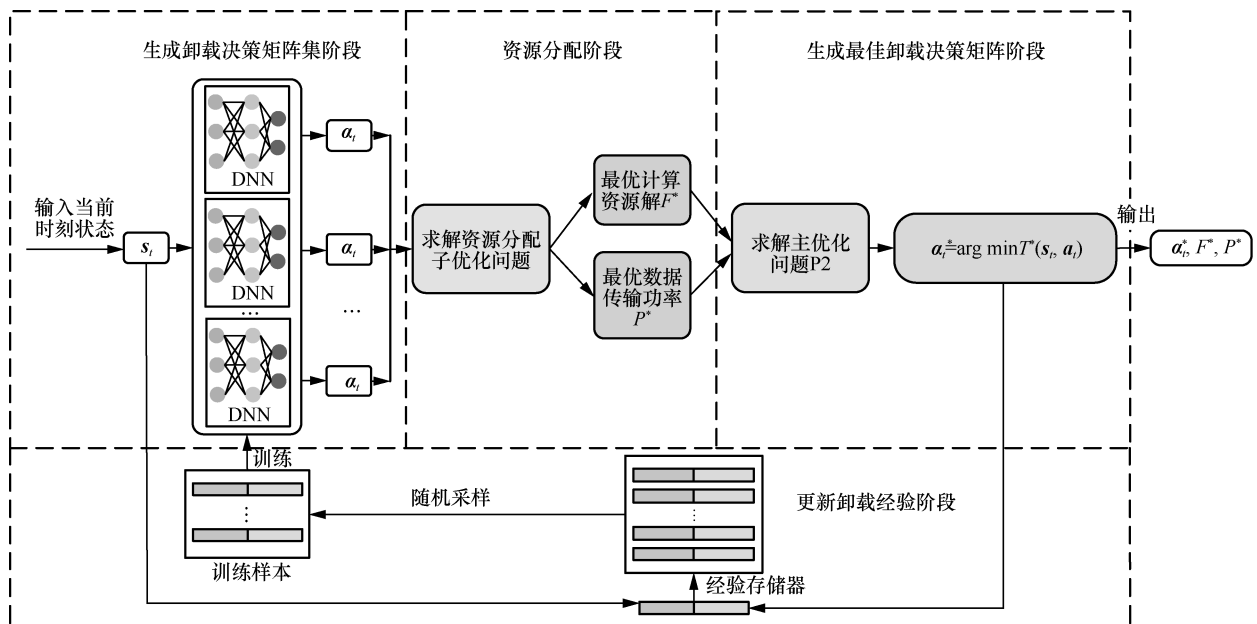


图 2 OOPC-DRL 计算任务卸载策略架构

$T^*(s_t, \alpha_t)$

- 7) 将  $(s_t, \alpha_t^*)$  存储到经验存储器中
- 8) if 经验存储器  $B$  的大小  $|B| > |\omega|$
- 9) 从经验存储器中随机选取  $\Omega$  批训练数据样本
- 10) 使用 Adam 随机梯度下降算法更新神经网络参数  $\theta_{\Omega_t}$
- 11) end if
- 12)  $t=t+1$ , 进入下一状态  $s_{t+1}$
- 13) end for
- 14) 输出  $(\alpha^I, \alpha^{II}, \alpha^{III}), F^*, P^*$

#### 4 实验结果分析与讨论

基于海洋网络的计算任务卸载模型，本节在 Python3.6 和 Tensorflow 2.0.0 环境下对 OOPC-DRL 算法进行了仿真实验。海洋网络仿真场景主要由强型层、中型层以及弱型层节点构成。网络环境下的计算任务卸载仿真参数如表 2 所示。OOPC-DRL 算法中 DNN 的相关设置参数参考文献[32]，如表 3 所示。

表 2 网络环境下的计算任务卸载仿真参数

参数	数值
系统总带宽/MHz	20
$L_m^I$ 节点处理能力/GHz	200
$L_n^{II}$ 节点处理能力/GHz	[10,50]
$L_k^{III}$ 节点处理能力/GHz	[2,3]
$L_k^{III}$ 节点数量/个	10
$L_k^{III}$ 节点输入数据的大小/MB	[10,30]
$L_k^{III}$ 节点完成计算内容的 CPU 周期数/GHz	[30,50]
上行链路最大发射功率/W	[0.05,0.1]
背景噪声功率/dBm	-70

表 3 DNN 的相关设置参数

参数	数值
神经元层数/层	4
学习率	0.01
神经网络训练间隔	10
经验存储器容量/个	1 000
批规模/个	128
并行 DNN 数量/个	4

为了分析 OOPC-DRL 算法的收敛性能，本文将增益定义为

$$\text{Gain} = \frac{\min_{(\alpha^I, \alpha^{II}, \alpha^{III}) \in \{0,1\}^N} T^*(\alpha^I, \alpha^{II}, \alpha^{III}, P, F)}{T^*(\alpha^I, \alpha^{II}, \alpha^{III}, P, F)} \quad (30)$$

其中，分子是通过枚举所有可行的卸载动作得到的最优解。

图 3 和图 4 分别展示了 OOPC-DRL 算法的训练损失函数曲线和增益曲线。其中，训练损失随着训练步数的增加而降低，增益随着训练步数的增加而趋近于 1。在训练初始时，DNN 需要不断探索动作，增益曲线会存在较大程度的波动。当训练步数在 2 000 步时，训练损失和增益基本同时收敛；当训练步数大于 2 000 步时，训练损失降低到 0.04，增益收敛到 0.98，这说明本文所提 OOPC-DRL 算法可实现在有限训练步数内的稳定收敛，且能快速收敛到最优解。

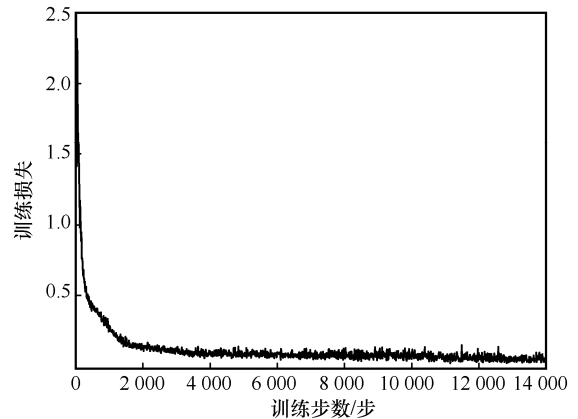


图 3 基于 OOPC-DRL 算法的损失函数曲线

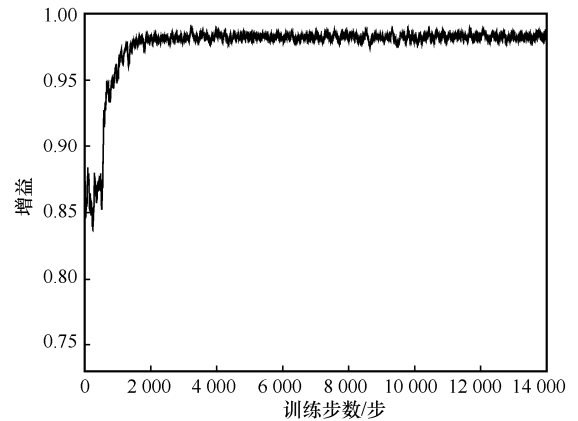


图 4 基于 OOPC-DRL 算法的增益曲线

图 5 展示了 DNN 数量对 OOPC-DRL 算法收敛性能的影响。总体来看，随着训练步数的增加，增益逐渐收敛到 1，且 DNN 数量越多，收敛效果越明显。然而，当只使用一个 DNN 时，OOPC-DRL 算法无法从其自身生成的数据中获取任何信息，并且无法收敛。因此，OOPC-DRL 算法至少需要 2 个 DNN。OOPC-DRL 算法的计算复杂度主要来自利用 DNN 生成卸载决策矩阵阶段，当 DNN 数量增加时，增益曲线虽然有所改善，但是效果并不明显，且算法的复杂

度大幅度增加, 因此, 在设置 DNN 数量时, 需要权衡其性能和复杂度。所以, 本文将 DNN 数量设置为 4。

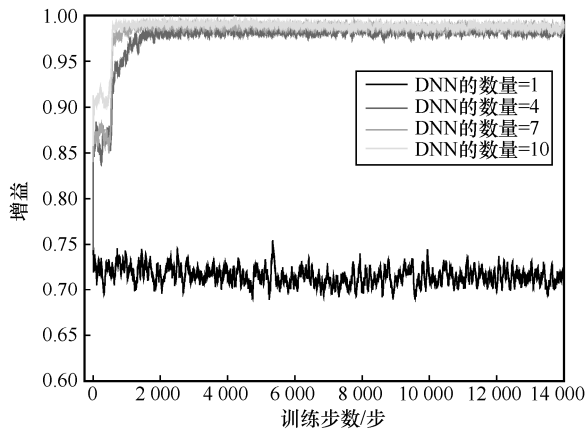


图 5 DNN 数量对 OOPC-DRL 算法收敛性能的影响

图 6 进一步说明了学习率对 OOPC-DRL 算法收敛性能的影响。从图 6 可以看出, 随着训练步数增加, OOPC-DRL 算法在不同学习率下的增益均逐渐提升。学习率越大, OOPC-DRL 算法的增益曲线收敛越快。然而, 当学习率为 0.1 时, 增益值反而降低, 这是由于学习率过高导致网络陷入了局部最优的困局。相反, 当学习率过低时(取值为 0.000 1), 算法收敛速度较慢, 大约在训练 4 000 步后达到收敛。因此结合这一实际情况, 本文在剩余的仿真实验中将学习率设置为 0.01。

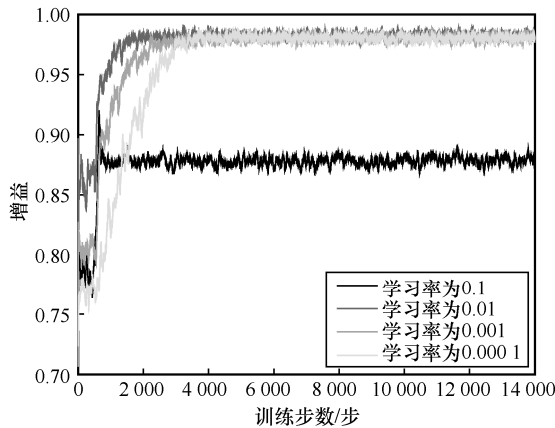


图 6 学习率对 OOPC-DRL 算法收敛性能的影响

为了验证 OOPC-DRL 算法有效性, 本文将其与以下 5 种现有策略进行对比。

- 1) 本地计算策略(简称 Local 策略)。节点的全部计算任务均在本地处理, 不进行卸载处理。
- 2) 基于资源分配的随机卸载策略(简称 Random 策略)<sup>[33]</sup>。每当节点遇到新状态时, 随机生成卸载动作, 利用本文中的子问题优化方法分配计算资源。

3) 基于分组交叉学习的粒子群优化(GCL-PSO, group cross learning based particle swarm optimization)算法<sup>[25]</sup>。利用文献[25]中的 GCL-PSO 算法生成卸载动作, 通过本文的子问题优化方法分配计算资源。

4) 基于变异操作的人工鱼群算法(MO-AFSA, artificial fish swarm algorithm based on mutation operation)<sup>[26]</sup>。利用文献[26]中的 MO-AFSA 生成卸载动作, 通过本文的子问题优化方法分配计算资源。

5) 枚举(Enumerate)策略。对所有卸载决策进行穷举搜索, 并选择最优卸载决策, 但耗时过长, 不能满足在实际情况中的实时性。

图 7 展示了不同节点数量下各策略平均计算任务执行时延对比。从图 7 中可知, 除 Local 策略外, 其他策略的平均计算任务执行时延总体变化趋势均随着  $L_k^m$  节点数量的增大而增加。这是因为当系统中  $L_k^m$  节点数量增大时, 单个  $L_k^m$  节点分配的带宽和边缘服务器分配的计算资源相应减少, 计算任务卸载传输时延和处理时延增加, 导致平均计算任务执行时延增加。然而, 本文所提出的 OOPC-DRL 算法平均计算任务执行时延表现最优。Local 策略计算任务处理时延只与  $L_k^m$  节点的性能有关, 并且每个  $L_k^m$  节点的计算能力远小于边缘服务器节点。因此, Local 策略在不同节点数量下的平均计算任务执行时延最高。Random 策略具有很强的随机性, 在不同  $L_k^m$  节点数量下的平均计算任务执行时延波动最大。由于寻优时容易陷入局部最优, MO-AFSA 和 GCL-PSO 这 2 种策略较 OOPC-DRL 算法相比平均计算任务执行时延依次增加。通过以上对比可以看出, OOPC-DRL 算法平均计算任务执行时延接近具有最佳性能的 Enumerate 策略, 在降低  $L_k^m$  节点平均计算任务执行时延方面具有较好的表现。

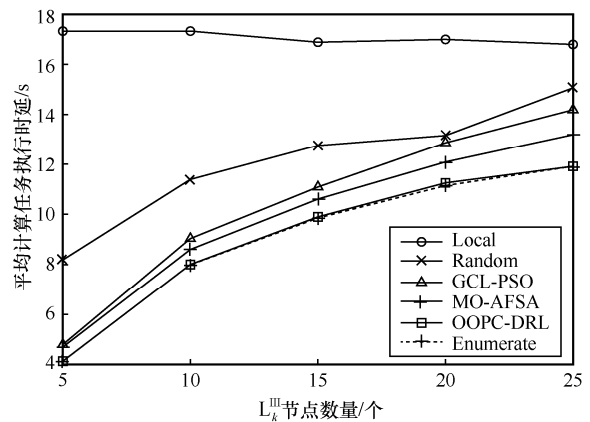


图 7 不同节点数量下各策略平均计算任务执行时延对比

图 8 展示了不同输入数据下各策略平均计算任务执行时延对比。从图 8 中可以看出，随着输入数据的增大，除 LOCAL 策略外，其他 5 种策略的平均计算任务执行时延都有不同程度的增加，但是 OOPC-DRL 算法相比其他策略表现更加优异，这是因为输入数据的大小只影响  $L_k^m$  节点的任务传输时延。Random 策略由于存在随机性，其增加幅度最大，GCL-PSO 和 MO-AFSA 策略容易陷入局部最优，寻优效果也不佳。OOPC-DRL 算法在不同输入数据下的平均计算任务执行时延接近 Enumerate 策略，表现依然最好。在输入数据明显增大时，OOPC-DRL 算法相较其他策略提升更明显，再次说明该策略能够更好地应对资源有限情况下的任务分配问题，即能够在有限的资源情况下最小化系统时延。

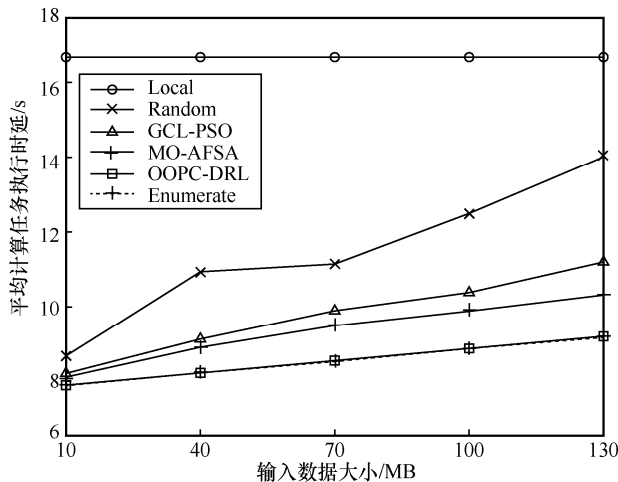


图 8 不同输入数据下各策略平均计算任务执行时延对比

图 9 展示了不同任务计算量下各策略平均计算任务执行时延对比。所有策略产生的平均计算任务执行时延均随着  $L_k^m$  节点任务计算量的增大而增加，而 Enumerate 策略和 OOPC-DRL 算法的平均计算任务执行时延的增加幅度明显低于其他 4 种策略。这是由于任务计算量增大时，节点需要花费更多的时间进行计算处理，而  $L_k^m$  节点的计算能力最弱，故 Local 策略的平均计算任务执行时延增加幅度最大。OOPC-DRL 算法的平均计算任务执行时延虽然随着  $L_k^m$  节点任务计算量的增大而增加，但是始终非常接近能够产生最佳性能的 Enumerate 策略，再次阐明了 OOPC-DRL 算法在有限的资源情况下最小化系统时延的能力。

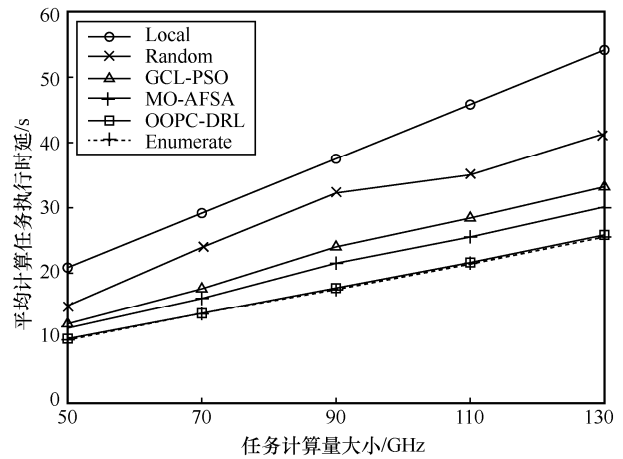


图 9 不同任务计算量下各策略平均计算任务执行时延对比

图 10 展示了不同计算能力下各策略平均计算任务执行时延对比(以固定输入数据大小为 20 MB，任务计算量的大小为 40 GHz 为例)。仿真结果表明，除 Local 策略之外，其他策略的平均计算任务执行时延总体均随着边缘服务器计算能力的增大而降低，提高了  $L_k^m$  节点任务执行效率。其中，OOPC-DRL 算法在不同边缘服务器计算能力下的平均计算任务执行时延始终接近 Enumerate 策略，均小于其他 4 种策略，且随着边缘服务器计算能力越来越强，OOPC-DRL 与其他 4 种策略的差距也越来越大。再次表明了本文所提算法随着边缘服务器计算能力的变化可以实时调整卸载策略，以确保系统具有更好的动态适应性。值得注意的是，Random 策略是随机生成卸载动作的，仿真时会出现 50 GHz 下对应的平均计算任务执行时延高于 40 GHz 的情况。

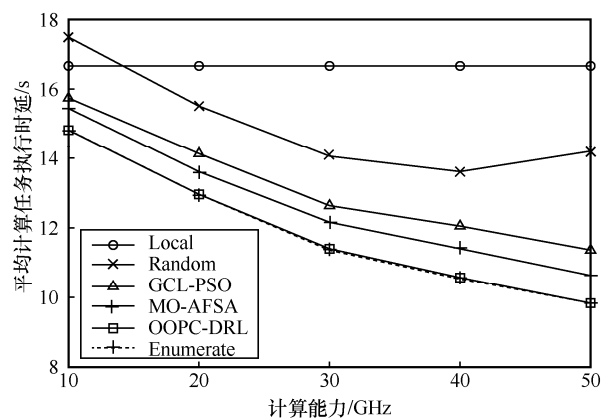


图 10 不同计算能力下各策略平均计算任务执行时延对比

### 5 结束语

本文以最小化海洋网络节点任务执行时延为

目标, 利用海洋网络节点的强异构特征属性对其进行归类分层, 并提出了一种基于深度强化学习的海洋网络计算任务卸载策略。实验结果表明, 与传统算法相比, 本文所提出的 OOPC-DRL 计算任务卸载算法能够在海洋信息系统下有效地降低网络节点的计算任务卸载时延, 可以更好地满足对实时性要求较高的海事应用服务需求, 并且能够在大规模任务流下保持海洋网络的稳健性。未来研究工作将针对多维度的水上水下节点之间的任务卸载需求, 深入分析并考虑排队等待时延对节点任务卸载的影响。

### 参考文献:

- [1] 瞿逢重, 来杭亮, 刘建章, 等. 海洋物联网关键技术研究与应用[J]. 电信科学, 2021, 37(7): 25-33.  
QU F Z, LAI H L, LIU J Z, et al. Research and application on key techniques of marine IoT[J]. Telecommunications Science, 2021, 37(7): 25-33.
- [2] YANG T, FENG H L, YANG C M, et al. Multivessel computation offloading in maritime mobile edge computing network[J]. IEEE Internet of Things Journal, 2019, 6(3): 4063-4073.
- [3] CHEN S Y, LUO S Y, WANG Q, et al. An intelligent task offloading algorithm (iTOA) for UAV edge computing network[J]. Digital Communications and Networks, 2021, 6(4): 433-443.
- [4] LIANG M Z, SU X, LIU X F, et al. Intelligent ocean convergence platform based on IoT empowered with edge computing[J]. Journal of Internet Technology, 2020, 21(1): 235-244.
- [5] ZHOU Y Q, LIU H, PAN Z G, et al. Two-stage cooperative multicast transmission with optimized power consumption and guaranteed coverage[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(2): 274-284.
- [6] SHU W N, LI Y. Joint offloading strategy based on quantum particle swarm optimization for MEC-enabled vehicular networks[J]. Digital Communications and Networks, 2022: doi.org/10.1016/j.dcan.2022.03.009.
- [7] LIU L, ZHOU Y Q, ZHUANG W H, et al. Tractable coverage analysis for hexagonal macrocell-based heterogeneous UDNs with adaptive interference-aware CoMP[J]. IEEE Transactions on Wireless Communications, 2019, 18(1): 503-517.
- [8] ANBALAGAN S, KUMAR D, RAJA G, et al. SDN assisted Stackelberg game model for LTE-WiFi offloading in 5G networks[J]. Digital Communications and Networks, 2019, 5(4): 268-275.
- [9] EL-HABER E, NGUYEN T M, ASSI C. Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds[J]. IEEE Transactions on Communications, 2019, 67(5): 3407-3421.
- [10] NING Z L, DONG P R, KONG X J, et al. A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of things[J]. IEEE Internet of Things Journal, 2019, 6(3): 4804-4814.
- [11] JIANG F, WEI F M, WANG J X, et al. Delay-aware energy minimization offloading scheme for mobile edge computing[C]//Proceedings of 2020 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2020: 717-722.
- [12] HU X Y, WANG L F, WONG K K, et al. Edge and central cloud computing: a perfect pairing for high energy efficiency and low-latency[J]. IEEE Transactions on Wireless Communications, 2020, 19(2): 1070-1083.
- [13] ZHANG K, MAO Y M, LENG S P, et al. Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading[J]. IEEE Vehicular Technology Magazine, 2017, 12(2): 36-44.
- [14] DAI Y Y, XU D, MAHARJAN S, et al. Joint load balancing and offloading in vehicular edge computing and networks[J]. IEEE Internet of Things Journal, 2019, 6(3): 4377-4387.
- [15] HUY H V, HO T M, LE L B. Mobility-aware computation offloading in MEC-based vehicular wireless networks[J]. IEEE Communications Letters, 2020, 24(2): 466-469.
- [16] DU J B, YU F R, CHU X L, et al. Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1079-1092.
- [17] GU B, ZHOU Z Y. Task offloading in vehicular mobile edge computing: a matching-theoretic framework[J]. IEEE Vehicular Technology Magazine, 2019, 14(3): 100-106.
- [18] MA X T, ZHAO J H, LI Q P, et al. Reinforcement learning based task offloading and take-back in vehicle platoon networks[C]//Proceedings of 2019 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2019: 1-6.
- [19] XU S Y, LIU Q C, GONG B, et al. RJCC: reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT[J]. IEEE Internet of Things Journal, 2020, 7(9): 8059-8076.
- [20] KIRAN N, PAN C Y, WANG S H, et al. Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks[J]. Journal of Communications and Networks, 2020, 22(1): 1-11.
- [21] JIANG F B, WANG K Z, DONG L, et al. Deep-learning-based joint resource scheduling algorithms for hybrid MEC networks[J]. IEEE Internet of Things Journal, 2020, 7(7): 6252-6265.
- [22] WANG R Y, JIANG X, ZHOU Y J, et al. Multi-agent reinforcement learning for edge information sharing in vehicular networks[J]. Digital Communications and Networks, 2022, 8(3): 267-277.
- [23] LI Y, WANG T S, WU Y, et al. Optimal dynamic spectrum allocation-assisted latency minimization for multiuser mobile edge computing[J]. Digital Communications and Networks, 2022, 8(3): 247-256.
- [24] YANG T T, FENG H L, GAO S, et al. Two-stage offloading optimization for energy-latency tradeoff with mobile edge computing in maritime Internet of things[J]. IEEE Internet of Things Journal, 2020, 7(7): 5954-5963.
- [25] 苏新, 薛溟阳, 周一青, 等. 面向海洋观测监测传感网的计算卸载方

法研究[J]. 通信学报, 2021, 42(5): 149-163.

SU X, XUE H Y, ZHOU Y Q, et al. Research on computing offloading method for maritime observation monitoring sensor network[J]. Journal on Communications, 2021, 42(5): 149-163.

- [26] 苏新, 王子怡, 王宇鹏, 等. 海洋观测监测传感器网络多接入边缘计算卸载方法[J]. 物联网学报, 2021, 5(1): 36-52.

SU X, WANG Z Y, WANG Y P, et al. Multi-access edge computing offloading in maritime monitoring sensor networks[J]. Chinese Journal on Internet of Things, 2021, 5(1): 36-52.

- [27] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.

- [28] MENG A Q, WEI G D, ZHAO Y, et al. Green resource allocation for mobile edge computing[J]. Digital Communications and Networks, 2022: doi.org/10.1016/j.dcan.2022.03.001.

- [29] LIU L, ZHOU Y Q, GARCIA V, et al. Load aware joint CoMP clustering and inter-cell resource scheduling in heterogeneous ultra dense cellular networks[J]. IEEE Transactions on Vehicular Technology, 2018, 67(3): 2741-2755.

- [30] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

- [31] WANG H D, ZHOU J, LIU W, et al. BGD-based Adam algorithm for time-domain equalizer in PAM-based optical interconnects[J]. Optics Letters, 2020, 45(1): 141.

- [32] 张鹏, 田辉, 赵鹏涛, 等. 多智能体协作场景下基于强化学习值分解的计算卸载策略[J]. 通信学报, 2021, 42(6): 1-15.

ZHANG P, TIAN H, ZHAO P, et al. Computation offloading strategy in multi-agent cooperation scenario based on reinforcement learning with value-decomposition[J]. Journal on Communications, 2021, 42(6): 1-15.

- [33] ZHANG W W, WEN Y G, WU D O. Collaborative task execution in mobile cloud computing under a stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2015, 14(1): 81-93.

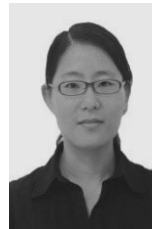
## [作者简介]



苏新(1986-), 男, 河北霸州人, 博士, 河海大学教授, 主要研究方向为移动通信、边缘/雾计算、智慧海洋等。



孟蕾蕾(1996-), 女, 甘肃定西人, 河海大学硕士生, 主要研究方向为移动通信、边缘/雾计算、智慧海洋等。



周一青(1975-), 女, 浙江杭州人, 中国科学院计算技术研究所研究员, 主要研究方向为通信与计算融合、移动边缘计算、存储通信、干扰管控等。



CELIMUGE Wu(1979-), 男, 博士, 日本电气通信大学教授, 主要研究方向为无线网络、物联网系统、边缘计算等。