

面向物联网的云边端协同计算中任务卸载与资源分配算法研究

施建锋* 陈忻阳 李宝龙

(南京信息工程大学电子与信息工程学院 南京 214442)

摘要: 为满足远郊和灾区物联网(IoT)设备的时延与能耗需求, 该文构建了由IoT终端、低地球轨道(LEO)卫星和云计算中心组成的新型动态卫星物联网模型。在时延、能耗等实际约束条件下, 将系统时延与能耗加权视为系统开销, 构造了最小化系统开销的任务卸载、功率和计算资源联合分配问题。针对动态任务到达场景, 提出一种模型辅助的自适应深度强化学习(MADRL)算法, 实现任务卸载决策、通信资源和计算资源的联合配置。该算法将问题分为两部分解决, 第1部分通过模型辅助、二分搜索算法和梯度下降法优化了通信资源与计算资源; 第2部分通过自适应深度强化学习算法训练出 Q 网络以适应随机任务的到达, 进行卸载决策优化。该算法实现了有效的资源分配和可靠及时的任务卸载决策, 且在降低系统开销方面表现出优异的效果。仿真结果表明, 引入卫星的移动性, 使得系统开销降低了41%。引入星间协作技术, 使系统开销降低了22.1%。此外, 该文所提算法收敛性能好。与基准算法相比, 该算法的系统开销降低了3%, 在不同环境下的性能表现都是最优。

关键词: 云边端协同计算; 卫星物联网; 深度强化学习; 任务卸载; 资源分配。

中图分类号: TN927

文献标识码: A

文章编号: 1009-5896(2025)02-0458-12

DOI: 10.11999/JEIT240659

1 引言

随着经济社会数字化转型和智能升级步伐加快, 物联网已经成为新型基础设施的重要组成部分^[1], 在智能家居、智慧城市、智慧交通、医疗健康等诸多领域发挥着重要的作用^[2]。现有的广域物联网网络主要建立在第5代移动通信(5th Generation mobile communication technology, 5G)地面网络的基础上^[2]。虽然地面网络已经随着科技的不断发展而逐渐完善, 但是受到环境和经济限制, 仍然存在许多挑战^[3,4]。地面网络基站部署面临挑战, 尤其是在远郊地区, 难以实现经济效益且易受自然灾害影响。与此相比, 卫星网络因其无缝覆盖、灵活性和可靠性, 正迅速成为重要通信基础设施, 推动全球互联互通的发展。卫星网络可作为地面网络的补充, 为远郊或灾区的物联网设备提供服务, 形成卫星辅助物联网(Satellite-assisted Internet of Things, SIoT)。SIoT被视为一种极具潜力的网络, 可为全球IoT设备提供无处不在和可靠的连接^[5,6]。

SIoT网络下的任务卸载与资源分配问题成为了一个新的研究热点。由于IoT设备通常功率有限, 可用于通信、计算和缓存的可用资源很少, 因此,

IoT设备产生的传感数据通常需要转发到云或边缘计算节点^[7]进行进一步处理。移动边缘计算(Mobile Edge Computing, MEC)将计算能力赋予网络的边缘, 通过内容缓存和在IoT设备附近部署边缘计算服务器, 可以有效降低处理时延。随着星载处理技术的发展, 带有星载处理单元的卫星也可以看作是一个边缘计算节点, 为地面用户提供计算服务^[8]。与卫星节点相比, 云平台具备更强的计算能力和能量供应, 但通常距离物联网设备较远, 导致较高的通信时延。随着5G技术发展, 物联网设备激增, 产生大量数据交互和处理需求。通过设计云-边-端协同的分布式计算架构, 可以挖掘分布式算力资源, 提升任务响应速度、减轻网络负担, 并支持高效的资源分配。

近年来, 支持MEC的SIoT得到了广泛的关注。文献^[9]研究了多轨道卫星网络中的动态卸载问题, 采用基于Stackelberg博弈的方法求解了最优卸载决策。文献^[10]设计了多卫星辅助网络中用户关联与计算卸载决策的联合优化问题, 优化目标为最小化任务计算能耗。文献^[11]提出了空天地一体化网络(Space-Air-Ground Integrated Network, SAGIN)中的计算卸载和缓存问题以减少任务完成时延。文献^[12]研究了节能SAGIN下的计算卸载问题, IoT设备可以根据自身通信条件和计算能力选择最合适的LEO卫星或无人机进行任务卸载。文献^[13]提出了一种联合优化通信和计算资源分配和计算卸载决策的随机计算卸载问题, 以最小化地面用户和低地球轨道(Low Earth Orbit, LEO)卫星的长期平

收稿日期: 2024-07-26; 改回日期: 2024-12-10; 网络出版: 2024-12-17

*通信作者: 施建锋 jianfeng.shi@nuist.edu.cn

基金项目: 国家自然科学基金(62201274, 62201275), 江苏省自然科学基金(BK20210641)

Foundation Items: The National Natural Science Foundation of China (62201274, 62201275), The Natural Science Foundation of Jiangsu Province (BK20210641)

均总功率成本。文献[14]提出了一种基于学习的天空地一体化网络任务卸载方法，但忽略了卫星处理任务的能力。这些研究主要集中在边缘计算或云计算单独应用的情况下。然而，随着设备激增和应用多样化，单一依赖卫星边缘或云计算难以满足需求。因此，本文提出基于终端-边缘(LEO卫星)-云的SIoT网络，以满足用户的多样化需求。

文献[15]研究了卫星辅助MEC系统中的任务划分和资源分配问题，旨在最大限度地减少工业IoT设备的任务完成时延。文献[16]提出了一种混合云和边缘计算的LEO卫星网络，可以为用户提供异构计算资源。文献[17]提出了一种基于软件定义网络和网络功能虚拟化的SAGIN-车联网(Internet of Vehicles, IoV)边缘云架构，优化业务时延、系统能耗、资源利用率和业务安全性。由于卫星边缘计算的计算卸载问题复杂，许多研究未考虑星间协同计算。然而，LEO卫星的计算资源受体积和重量限制，仅依靠单颗LEO卫星辅助所有用户计算任务有时无法满足所有用户对服务性能的期望^[18]。因此，本文考虑了多卫星协同计算对于提高服务质量和资源利用率至关重要。已有研究通过光通信或可见光通信系统实现星间链路(Inter-Satellite Link, ISL)，实现快速、可靠、高效的卫星网络^[19]。通过星间协作(Inter-Satellite Collaboration, ISC)利用ISL，既能扩大系统容量和覆盖范围，满足更多用户需求，又能减少单卫星资源消耗，提升卫星网络的整体生存能力。

文献[20]提出了一种由终端-卫星-云组成的3层边缘计算架构，通过ISC实现星上负载均衡，对卸载决策、通信和计算资源分配变量进行了联合优化。文献[21]研究了协同卫星边缘计算中计算卸载与资源分配的联合优化问题，提出了一种分层动态资源分配算法来解决该问题。上述研究大多基于静态场景，假设卫星网络拓扑和用户关联固定。然而，SIoT网络中的卫星高速运动，地面用户需频繁切换卫星，这对任务调度和资源分配提出更高要求。系统需实时调整卸载策略和资源分配参数，以适应卫星轨迹和可用性。因此，在LEO卫星高速运动和资源受限的情况下，如何确定任务的卸载策略是一个挑战性问题。

为了补充相关工作，本文研究基于终端-边缘-云的动态SIoT网络，利用ISC技术提升边缘性能，并通过将任务卸载到最合适的节点实现负载均衡，综合考虑时延与能耗需求，将卸载决策、通信和计算资源联合分配问题建模为混合整数非线性规划(Mixed Integer Non-Linear Programming,

MINLP)问题。结合传统优化与深度强化学习，提出模型辅助的自适应深度强化学习(Model-assisted Adaptive Deep Reinforcement Learning, MADRL)算法，优化任务卸载、CPU频率及传输功率，最终实现系统开销最小化。本文的主要贡献总结如下：

(1) 构建了基于终端-边缘-云的动态SIoT网络模型，以服务远郊与灾区的IoT设备。针对卫星边缘资源受限的问题，利用ISC技术来提高边缘的性能。同时，为了有效应对卫星高速移动性带来的挑战，通过对卫星移动性的建模，以及对卫星服务范围和调度周期的合理界定，动态确定用户的接入卫星。

(2) 构造了满足通信资源与计算资源约束下，系统开销最小化问题。通过设计高效的资源分配与任务卸载算法，确保任务在最合适的节点上顺利完成卸载与执行。

(3) 大量的仿真表明，本文所提出的算法具有良好的收敛性能。卫星移动性的引入使得系统开销降低了41%。ISC的引入使系统开销降低了22.1%。与(Deep Q Learning, DQN)等基准算法相比，MADRL可进一步降低系统开销3%。

2 系统模型和问题描述

本节首先构建了LEO卫星移动性模型和带ISC的卫星辅助物联网(Satellite-assited Internet of Things, SIoT)网络模型，描述了本地计算、边缘计算、云计算3种模式下的端到端时延与系统能耗。最后，提出了系统开销最小化的任务卸载与资源分配联合优化问题。表1列出了基本符号及其含义。

表1 基本符号及其含义

符号	含义
$\mathcal{M}$	设备集合
$\mathcal{D}$	灾区设备集合
$\mathcal{R}$	远郊设备集合
$\mathcal{S}$	LEO卫星集合
d_m^n	时隙 n 设备 m 生成任务的大小
c_m^n	时隙 n 设备 m 的工作负载
w_m^n	时隙 n 设备 m 处理任务所需CPU周期数
$T_m^{n,\max}$	时隙 n 设备 m 处理任务的最大容忍时延
x_m^n	时隙 n 设备 m 的任务卸载决策
f_m^n	时隙 n 设备 m 的CPU工作频率
p_m^n	时隙 n 设备 m 的传输功率
t_m^n	时隙 n 设备 m 的系统时延
e_m^n	时隙 n 设备 m 的系统能耗
c_m^n	时隙 n 设备 m 的系统开销

2.1 LEO卫星移动性模型

LEO卫星 s 与设备 m 之间的几何关系如图1所示。考虑到LEO卫星的移动性^[22], 假设LEO卫星在离地面高度 H 的轨道上以 V_s 匀速飞行, $\alpha_{m,s}[n]$ 为卫星 s 在时隙 n 时与水平正方向的夹角, $\gamma_{m,s}[n]$ 为卫星 s 在时隙 n 时到卫星用户 m 的剩余覆盖弧长对应的几何夹角, R 为地球半径, H 为LEO卫星的高度。忽略其他因素的影响, 当 $0^\circ < \alpha_{m,s} < 180^\circ$ 时, LEO卫星可以与设备 m 建立通信链路。

据几何关系, 当 $0^\circ < \alpha_{m,s} < 90^\circ$ 时, $\gamma_{m,s}[n]$ 表示为

$$D_{m,s}[n] = \begin{cases} \left\{ R + H \sin \left(\arccos \frac{R}{R+H} - \gamma_{m,s}[n] \right) \right\} / \cos \alpha_{m,s}[n], & 0^\circ < \alpha_{m,s}[n] < 90^\circ \\ H, & \alpha_{m,s}[n] < 90^\circ \\ \left\{ R + H \sin \left(\gamma_{m,s}[n] - \arccos \frac{R}{R+H} \right) \right\} / \cos \alpha_{m,s}[n], & 90^\circ < \alpha_{m,s}[n] < 180^\circ \end{cases} \quad (3)$$

2.2 网络模型

本文考虑动态场景, SIoT网络模型随时间变化。系统中包含 M 个IoT设备, S 个LEO卫星和1个云计算中心。其中 R_d 个设备在远郊, D 个设备在灾区。假设远郊、灾区和云中心在地面上从左到右依次分布。灾区因突发事件导致人口密集和通信需求紧急, 系统需快速响应大量任务请求。为此, 接入卫星与周围四颗卫星建立ISL, 同一轨道上相邻卫星以及相邻轨道上的卫星通过ISL相连^[23], 形成边缘云, 提升通信能力并确保快速响应。而远郊地区人口稀疏, 通信需求和任务生成频率较低, 系统可专注于长期稳定性和资源利用率, 单颗接入卫星即可满足需求, 降低系统复杂性并提高资源效率。

为不失一般性, 本文将卫星的服务范围设定为一个圆形区域。虽然LEO卫星的高速移动会导致覆盖的地理区域不断变化, 但是固定且相对较大的服务范围为任务卸载和切换提供了缓冲时间, 使算法能够在动态变化中平稳地进行卸载和资源分配, 确保优化逻辑稳定。

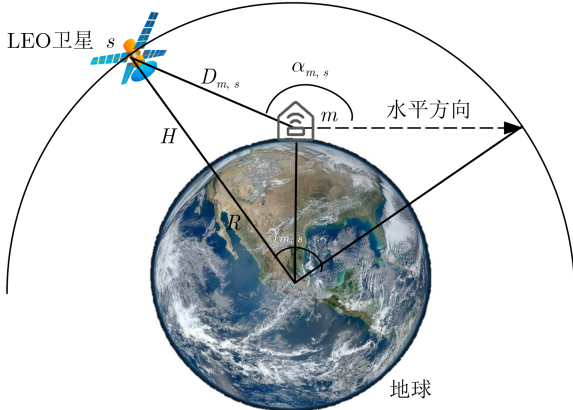


图1 LEO卫星 s 与设备 m 之间的几何关系

$$\gamma_{m,s}[n] = \arccos \left(\frac{R}{R+H} \right) - \arccos \left(\frac{R \cos \alpha_{m,s}[n]}{R+H} \right) + \alpha_{m,s}[n] \quad (1)$$

当 $90^\circ < \alpha_{m,s} < 180^\circ$ 时, $\gamma_{m,s}[n]$ 表示为

$$\gamma_{m,s}[n] = \arccos \left(\frac{R}{R+H} - \cos \alpha_{m,s}[n] \right) + \arccos \left(\frac{R}{R+H} \right) - \pi - \alpha_{m,s}[n] \quad (2)$$

卫星 s 在时隙 n 与设备 m 的距离表示为式(3)

LEO卫星表示为集合 $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, 其中 $\{S_1, S_2\}$ 为接入卫星。将卫星 S_1 的服务范围刚覆盖到云计算中心的时刻视为时刻0, 此时, 灾区的接入卫星为 S_1 , 协作卫星为 S_2, S_3, S_4, S_5 , 远郊的接入卫星为 S_2 。随着时间的推移, 灾区的接入卫星切换为 S_2 , 协作卫星为 S_1, S_3, S_4, S_5 , 远郊的接入卫星仍为 S_2 。直到卫星 S_1 的服务范围脱离云计算中心, 将卫星 S_2 的服务范围刚好覆盖到云计算中心的时刻视为下一个周期的时刻0, 将这个过程视为一个调度周期, 重点研究一个周期内任务卸载与资源分配问题。

设一个时隙的长度为 τ , 一个周期分为 N 个时隙, 假设在一个时隙内信道状态不变。将设备 m 在时隙 n 的任务建模为 $k_m^n = \{d_m^n, c_m^n, T_m^{n,\max}\}$, 表示任务 k_m^n 包含 d_m^n bit数据, 需要在时间 $T_m^{n,\max}$ 内完成, 其工作负载为 c_m^n kcycle/bit, 处理任务需要的CPU周期数为 $w_m^n = d_m^n \times c_m^n$, 用 x_m^n 表示任务 k_m^n 的卸载决策。

对于卫星通信, 信道衰落与地面通信完全不同。这里, 本文考虑一个更贴近真实场景的卫星信道模型, 包括自由空间损耗(Free-Space Loss, FSL)、雨和云衰减。FSL表示为

$$\phi_{\text{fs}} = \left(\frac{4\pi d_F}{\lambda} \right)^2 = \left(\frac{4\pi d_F f_c}{c} \right)^2 \quad (4)$$

其中, d_F (km)为通信距离, λ 为波长, f_c (GHz)为载波频率, c 为光速。当载波频率在10 GHz以上时, 降雨是卫星通信信号衰减的主要原因之一。降雨衰减由式(5)给出

$$\phi_{\text{rain}} = a R_{0.01}^b d_{\text{eff}} \quad (5)$$

其中, $R_{0.01}^b$ 为每年超过0.01%的降雨率, d_{eff} 为有

效路径, a 和 b 为雨滴大小分布、温度和频率相关的回归系数。云的衰减由式(6)给出

$$\phi_{\text{cloud}} = \frac{Lk_c}{\sin \theta} = \frac{L}{\sin \theta} \frac{0.819f_c}{\varepsilon''} \left/ \left[1 + \left(\frac{2 + \varepsilon'}{\varepsilon''} \right)^2 \right] \right. \quad (6)$$

其中, L 为云中液态水总柱状含量, k_c 为云层的特定衰减系数, ε' 和 ε'' 分别为水介电常数的实部和虚部。因此, 卫星通信过程中的总信道衰落可表示为

$$h = \phi_{\text{fs}} \phi_{\text{rain}} \phi_{\text{cloud}} \quad (7)$$

2.2.1 本地计算模型

设备 m 在时隙 n 的CPU工作频率为 f_m^n , 设备 m 在时隙 n 的任务处理时延表示为

$$t_m^{n,l} = \frac{w_m^n}{f_m^n} \quad (8)$$

设备 m 在时隙 n 的任务处理能耗表示为

$$e_m^{n,l} = \varepsilon w_m^n (f_m^n)^2 \quad (9)$$

设备 m 在时隙 n 的开销表示为式(10)

$$c_m^{n,l} = \lambda_l t_m^{n,l} + (1 - \lambda_l) e_m^{n,l} \quad (10)$$

其中, λ_l 为本地计算模式下的时延敏感系数。

2.2.2 边缘计算模型

通过对卫星运动过程的研究, 可以将1个周期

分为两个阶段, 设中间时刻 $t = t_{\text{mid}}$, 当 $t < t_{\text{mid}}$ 时, 灾区与云计算中心在卫星 S_1 的服务范围内, 远郊在卫星 S_2 的服务范围内。当 $t > t_{\text{mid}}$ 时, 云计算中心在卫星 S_1 的服务范围内, 灾区与远郊在卫星 S_2 的服务范围内。设卫星 S_1, S_2 在时隙 n 的CPU工作频率为 f_1^n, f_2^n , 传输功率为 p_1^n, p_2^n 。信号传输速率由香农公式计算得出。假设物联网设备具有足够数量的正交信道, 可以忽略多个设备之间的信道分配。这一假设是基于多频时分多址[24]技术的应用, 该技术可以根据当前需求和交通状况对这些正交信道进行动态分配。

设备 m 在时隙 n 的时延表示为式(11), 其中, $c_{m,1}$ 表示设备 m 将任务卸载到卫星 S_1 的信号传输速率, $c_{m,2}$ 表示设备 m 将任务卸载到卫星 S_2 的信号传输速率, c_s 表示接入卫星与协作卫星之间的信号传输速率, $D_{m,1}$ 表示设备 m 与卫星 S_1 之间的距离, $D_{m,2}$ 表示设备 m 与卫星 S_2 之间的距离, D_S 表示接入卫星与协作卫星之间的距离。 $t < t_{\text{mid}}$ 时, 灾区用户使用边缘计算时将任务卸载给接入卫星 S_1 , $t > t_{\text{mid}}$ 时, 灾区用户将任务卸载给接入卫星 S_2 。总时延包括传输时延[25]、任务处理时延和传播时延[26]。

$$t_m^{n,e} = \begin{cases} \frac{d_m^n}{c_{m,1}} + \frac{w_m^n}{f_1^n} + \frac{D_{m,1}}{c}, & t < t_{\text{mid}}(\text{No ISL}), m \in \mathcal{D} \\ \frac{d_m^n}{c_{m,2}} + \frac{w_m^n}{f_2^n} + \frac{D_{m,2}}{c}, & t > t_{\text{mid}}(\text{No ISL}), m \in \mathcal{D} \\ \frac{d_m^n}{c_{m,1}} + \frac{w_m^n}{f_1^n} + \frac{D_{m,1}}{c} + \frac{d_m^n}{c_s} + \frac{D_S}{c}, & t < t_{\text{mid}}(\text{ISL}), m \in \mathcal{D} \\ \frac{d_m^n}{c_{m,2}} + \frac{w_m^n}{f_2^n} + \frac{D_{m,2}}{c} + \frac{d_m^n}{c_s} + \frac{D_S}{c}, & t > t_{\text{mid}}(\text{ISL}), m \in \mathcal{D} \\ \frac{d_m^n}{c_{m,2}} + \frac{w_m^n}{f_2^n} + \frac{D_{m,2}}{c}, & m \in \mathcal{R} \end{cases} \quad (11)$$

$$e_m^{n,e} = \begin{cases} p_m^n \frac{d_m^n}{c_{m,1}} + \varepsilon w_m^n (f_1^n)^2, & t < t_{\text{mid}}(\text{No ISL}), m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,2}} + \varepsilon w_m^n (f_2^n)^2, & t > t_{\text{mid}}(\text{No ISL}), m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,1}} + p_m^n \frac{d_m^n}{c_s} + \varepsilon w_m^n (f_1^n)^2, & t < t_{\text{mid}}(\text{ISL}), m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,2}} + p_m^n \frac{d_m^n}{c_s} + \varepsilon w_m^n (f_2^n)^2, & t > t_{\text{mid}}(\text{ISL}), m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,2}} + \varepsilon w_m^n (f_2^n)^2, & m \in \mathcal{R} \end{cases} \quad (12)$$

设备 m 在时隙 n 的能耗表示为式(12), 其中, p_m^n 表示设备 m 在时隙 n 的传输功率。

边缘计算模式下设备 m 在时隙 n 的开销为

$$c_m^{n,e} = \lambda_e t_m^{n,e} + (1 - \lambda_e) e_m^{n,e} \quad (13)$$

其中, λ_e 为边缘计算模式下的时延敏感系数。

式(11)与式(12)中, 判断是否需要ISL的规则是: 任务卸载到接入卫星时, 无需通过ISL进行任务传输。任务卸载到协作卫星时, 则需要通过ISL将任务从接入卫星传输至协作卫星。

2.2.3 云计算模型

设云计算中心单核CPU工作频率为 f_c ，核心数为 N_c 。设备 m 在时隙 n 的时延表示为式(14)，其中 $c_{m,1}$ 表示设备 m 将任务卸载到卫星 S_1 的信号传输速率， $c_{m,2}$ 表示设备 m 将任务卸载到卫星 S_2 的信号传输速率， $c_{2,1}$ 表示卫星 S_2 将任务卸载

到卫星 S_1 的信号传输速率， $c_{1,c}$ 表示卫星 S_1 将任务卸载到云计算中心 c 的信号传输速率， $D_{m,1}$ 表示设备 m 与卫星 S_1 之间的距离， $D_{m,2}$ 表示设备 m 与卫星 S_2 之间的距离， $D_{2,1}$ 表示卫星 S_1 与卫星 S_2 之间的距离， $D_{1,c}$ 表示卫星 S_1 与云计算中心之间的距离。

$$t_m^{n,c} = \begin{cases} \frac{d_m^n}{c_{m,1}} + \frac{d_m^n}{c_{1,c}} + \frac{w_m^n}{f_c \times N_c} + \frac{(D_{m+1} + D_{1,c})}{c}, & t < t_{\text{mid}}, m \in \mathcal{D}, n \in N \\ \frac{d_m^n}{c_{m,2}} + \frac{d_m^n}{c_{2,c}} + \frac{d_m^n}{c_{1,c}} + \frac{w_m^n}{f_c \times N_c} + \frac{(D_{m+2} + D_{2,1} + D_{1,c})}{c}, & t > t_{\text{mid}}, m \in \mathcal{D}, n \in N \\ \frac{d_m^n}{c_{m,2}} + \frac{d_m^n}{c_{2,c}} + \frac{d_m^n}{c_{1,c}} + \frac{w_m^n}{f_c \times N_c} + \frac{(D_{m+2} + D_{2,1} + D_{1,c})}{c}, & m \in \mathcal{R}, n \in N \end{cases} \quad (14)$$

由于云的能量供应充足^[27]，本文参考文献^[28]中的问题表述，忽略了此时的计算能耗。设备 m 在时隙 n 的能耗表示为式(15)。

$$e_m^{n,c} = \begin{cases} p_m^n \frac{d_m^n}{c_{m,1}} + p_1^n \frac{d_m^n}{c_{1,c}}, & t < t_{\text{mid}}, m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,2}} + p_2^n \frac{d_m^n}{c_{2,1}} + p_1^n \frac{d_m^n}{c_{1,c}}, & t > t_{\text{mid}}, m \in \mathcal{D} \\ p_m^n \frac{d_m^n}{c_{m,2}} + p_2^n \frac{d_m^n}{c_{2,1}} + p_1^n \frac{d_m^n}{c_{1,c}}, & m \in \mathcal{R} \end{cases} \quad (15)$$

云计算模式下设备 m 在时隙 n 的总开销为

$$c_m^{n,c} = \lambda_c t_m^{n,c} + (1 - \lambda_c) e_m^{n,c} \quad (16)$$

其中， λ_c 为云计算模式下的时延敏感系数。

2.3 问题描述

根据动态SIoT网络中不同的任务处理方式，设备 m 在时隙 n 不同处理方式下生成的任务的端时延 t_m^n 和能耗 e_m^n 分别为

$$t_m^n = \begin{cases} t_m^{n,1}, x_m^n = 0 \\ t_m^{n,e}, x_m^n = 1, 2, 3, 4, 5 \\ t_m^{n,c}, x_m^n = 6 \end{cases} \quad (17)$$

$$e_m^n = \begin{cases} e_m^{n,1}, x_m^n = 0 \\ e_m^{n,e}, x_m^n = 1, 2, 3, 4, 5 \\ e_m^{n,c}, x_m^n = 6 \end{cases} \quad (18)$$

x_m^n 表示任务处理方式，当 x_m^n 等于0时，任务在本地处理，当 x_m^n 等于1时，任务卸载到接入卫星 S_1 中处理，当 x_m^n 等于2,3,4,5时，任务卸载到协作卫星 S_2, S_3, S_4, S_5 中处理，当 x_m^n 等于6时，任务卸载到云计算中心处理。卸载到边缘的任务循环分配给 S_1, S_2, S_3, S_4, S_5 5颗卫星。简单分析显示，本地处理能耗低但计算能力有限，可能无法满足时延要求；边缘处理减少计算时延但增加卫星能耗；云中心处理降低计算时延但传播时延增加。因此，不同

处理方式下的任务卸载决策与资源分配对时延和能耗有显著影响。本文研究动态网络场景中，在时延与能耗约束下，最小化系统开销的优化问题。优化目标是最小化一个周期内系统开销，优化变量包括任务卸载决策、IoT设备与LEO卫星的传输功率及CPU工作频率。式(19)给出了该优化问题的数学模型。针对时延敏感系数 λ ，本地节点因设备电池寿命，对能耗更敏感，系数应较低；边缘节点因卫星计算能力有限，对时延和能耗均敏感；云节点计算能力强，但传播时延高，对时延更敏感，因此系数应设置较高。

$$\begin{aligned} \text{P1: } & \min_{\{x_m^n, p_m^n, p_s^n, f_m^n, f_s^n\}} \sum_{n=1}^N \sum_{m=1}^M \lambda t_m^n + (1 - \lambda) e_m^n \\ \text{s.t. } & \left. \begin{aligned} \text{C1: } x_m^n &\in \{0, 1, \dots, 6\} \\ \text{C2: } t_m^n &\leq T_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ \text{C3: } e_m^n &\leq E_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ \text{C4: } e_s^n &\leq E_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ \text{C5: } 0 < f_m^n &\leq F_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ \text{C6: } 0 < p_m^n &\leq P_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ \text{C7: } 0 < f_s^n &\leq F_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ \text{C8: } 0 < p_s^n &\leq P_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ \text{C9: } |f_s^{n+1} - f_s^n| &< \varepsilon_f, \forall s \in \mathcal{S}, \forall n \in N \\ \text{C10: } |p_s^{n+1} - p_s^n| &< \varepsilon_p, \forall s \in \mathcal{S}, \forall n \in N \end{aligned} \right\} \quad (19) \end{aligned}$$

约束条件如下：C1定义任务处理方式为本、边缘或云计算；C2限制任务的端到端时延需小于任务需求；C3和C4确保设备和卫星的能耗不超过最大值；C5和C7约束设备与卫星的CPU频率小于最大工作频率；C6和C8限制设备与卫星的传输功率不超过最大功率；C9和C10为卫星CPU频率和传输功率的连续性约束。连续性约束确保相邻时隙资源调整的渐进性，避免因卫星移动引发频繁资源分配调整。如果没有连续性约束会导致两大问题：

(1)系统开销增加,每次调整需重新规划资源,浪费带宽与计算资源;(2)性能不稳定,频繁变动引发任务执行波动,难以及时响应需求。引入连续性约束可平滑调整资源分配,提升系统性能稳定性。

3 基于深度强化学习的任务卸载与资源分配算法

在问题式(19)中,CPU工作频率和传输功率为连续变量,卸载决策为离散变量。目标函数相对于这些变量是非线性的。因此,问题式(19)是一个MINLP问题。本文提出一种MADRL算法,第1层通过模型辅助并使用二分搜索算法和梯度下降法优化CPU工作频率和传输功率。第2层利用自适应DRL算法适应动态网络场景,通过自学习生成Q网络。

3.1 模型辅助资源分配

对于问题式(19),本文首先通过模型辅助对CPU工作频率和传输功率进行优化。

3.1.1 本地计算

当任务采用本地计算时,优化问题可以转化为

$$\left. \begin{aligned} P1: \min_{\{f_m\}} & \sum_{n=1}^N \sum_{m=1}^M \lambda_l t_m^{n,l} + (1 - \lambda_l) e_m^{n,l} \\ \text{s.t. } C1: & t_m^{n,l} \leq T_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C2: & e_m^{n,l} \leq E_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C3: & 0 < f_m^n \leq F_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \end{aligned} \right\} \quad (20)$$

本地计算模式下的开销函数可以表示为

$$F1 = \lambda_l W_m^n / f_m^n + (1 - \lambda_l) \varepsilon W_m^n (f_m^n)^2 \quad (21)$$

通过简单的计算,可以发现函数F1是一个先增后减的函数,极值点为 $f_m^{n*} = \sqrt[3]{[\lambda_l/(1 - \lambda_l)]/2\varepsilon}$, 约束C1, C2可以简化为 $f_m^n \geq W_m^n / T_m^{n,\max}$, $f_m^n \leq \sqrt{E_m^{n,\max}/\varepsilon W_m^n}$ F1可行解范围的下界和上界为

$$\left. \begin{aligned} f_1 &= \max \{0, W_m^n / T_m^{n,\max}\} \\ f_2 &= \min \left\{ \sqrt{E_m^{n,\max}/\varepsilon W_m^n}, F_m^{n,\max} \right\} \end{aligned} \right\} \quad (22)$$

本地计算的最优开销表示为

$$F1 = \begin{cases} F1(f_1), & f_m^{n*} < f_1 \\ F1(f_m^*), & f_1 \leq f_m^{n*} \leq f_2 \\ F1(f_2), & f_m^{n*} > f_2 \end{cases} \quad (23)$$

在本地计算的场景中,目标函数仅涉及1个未知数,即本地计算工作频率。由于这是一个单变量问题,因此本文可以直接使用二分法^[29]来高效地求解该问题。

3.1.2 边缘计算

当任务采用边缘计算时,优化问题可以转化为

$$\left. \begin{aligned} P1: \min_{\{p_m, f_s\}} & \sum_{n=1}^N \sum_{m=1}^M \lambda_e t_m^{n,e} + (1 - \lambda_e) e_m^{n,e} \\ \text{s.t. } C1: & t_m^n \leq T_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C2: & e_s^n \leq E_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ C3: & 0 < p_m^n \leq P_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C4: & 0 < f_s^n \leq F_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ C5: & |f_s^{n+1} - f_s^n| < \varepsilon_f, \forall s \in \mathcal{S}, \forall n \in N \end{aligned} \right\} \quad (24)$$

3.1.3 云计算

当任务采用云计算时,优化问题可以转换为

$$\left. \begin{aligned} P1: \min_{\{p_m, p_s\}} & \sum_{n=1}^N \sum_{m=1}^M \lambda_c t_m^{n,c} + (1 - \lambda_c) e_m^{n,c} \\ \text{s.t. } C1: & t_m^n \leq T_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C2: & 0 < p_m^n \leq P_m^{n,\max}, \forall m \in \mathcal{M}, \forall n \in N \\ C3: & 0 < p_s^n \leq P_s^{n,\max}, \forall s \in \mathcal{S}, \forall n \in N \\ C4: & |p_s^{n+1} - p_s^n| < \varepsilon_p, \forall s \in \mathcal{S}, \forall n \in N \end{aligned} \right\} \quad (25)$$

由于边缘计算模式下参数 p_m^n 与 f_s^n 相互约束,云计算模式下 p_m^n 与 p_s^n 相互约束,不能直接推导出最优开销的表达式。本文使用梯度下降法^[30]找出在特定约束条件下每个时隙中,使目标函数值最小的参数。

3.2 学习优化卸载决策

任务卸载决策由深度Q网络^[31] (Deep Q-Network, DQN)和双重深度Q网络^[32] (Double Deep Q-Network, DDQN)确定。DQN因目标网络选择动作而存在Q值过估计问题,DDQN通过在线网络选择动作加以改进。两者主要区别在于学习过程中 y 值的计算方式。然而,在数据量较小时,DDQN因网络参数训练不足,性能不如DQN;而数据量大时,DDQN表现更优。基于此,灾区设备多、数据量大,适合使用DDQN;远郊设备少、数据量小,DQN更具优势。为此,本文提出自适应DRL算法,针对不同场景动态训练Q网络实现任务卸载决策,其伪代码如算法1所示。

3.3 MADRL算法

为了实现全面而有效的解决方案,综合考虑前两节的优化策略,将模型辅助资源分配与学习优化卸载决策结合,提出MADRL算法,MADRL算法流程图如图2所示。

Q网络中的各个元素描述如下:

状态空间:在每个时隙中,系统观察当前状态,获取环境信息。本文分别用每个用户的任务状态来表示环境勘探信息 s ,包括任务大小 $d_m (m \in \mathcal{M})$,工作负载 $c_m (m \in \mathcal{M})$,最大容忍时延 $T_m^{n,\max} (m \in \mathcal{M})$ 。在时隙 n 时刻的状态空间定义为 $s_n = \{d_1^n, \dots, d_M^n, c_1^n, \dots, c_M^n, T_1^{n,\max}, \dots, T_M^{n,\max}\}$ 。

算法1 自适应DRL算法

输入: 开销矩阵

- (1) 初始化在线网络Q和目标网络Q_hat
- (2) 初始化训练参数
- (3) for episode = 1 to n_ep do
- (4) 初始化状态 s
- (5) for $n=1$ to N do
- (6) 根据 ε 贪婪策略选择动作 a
- (7) 更新状态 s'
- (8) end for
- (9) end for
- (10) if $\mathcal{D}$ 的大小 $\geq n_b$:
- (11) 从 $\mathcal{D}$ 中随机抽取最小批量转移元组
- (12) 根据任务状态选择DQN或DDQN计算 y 值
- (13) end if
- (14) 计算损失函数Loss(θ)
- (15) 更新在线网络Q
- (16) 每隔 X 步, 更新目标网络: Q_hat=Q
- (17) 更新状态 $s \leftarrow s'$
- (18) 返回Q网络

动作空间: 在线网络得到状态空间 s_n 后, 将产生相应的离散卸载决策 a_n , $a_n = \{x_1^n, x_2^n, \dots, x_M^n\}$, $x_m^n \in \{0, 1, 2, 3, 4, 5, 6\}$, ($m \in \mathcal{M}$).

奖励函数: 在状态 s 下采取行动 a 后, 环境进入下一个状态 s' 并返回奖励 r , 奖励值 r 定义为系统开销的倒数。

3.4 算法复杂度分析

在计算环境中, 算法的效率与复杂度是衡量其可行性和应用价值的重要指标。本节将分析组成MADRL算法的3种子算法的复杂度, 并进行总体复杂度分析。

(1) 二分搜索算法:

搜索范围为 $(0, F_m^{\max}]$, 每次搜索完成搜索范围减小 $1/2$, 假设搜索进行了 k 次, 搜索范围缩小到1, 即 $F_m^{\max}/2^k = 1$, 通过取对数, 可以解出 $k = \log_2 F_m^{\max}$, 因此, 二分搜索需要进行 $O(\log_2 F_m^{\max})$ 次比较, 才能找到目标元素。算法复杂度为 $O(\log_2 F_m^{\max})$ 。

(2) 梯度下降法:

梯度下降法的关键步骤包括对参数求偏导, 求解偏导数仅涉及常数时间的基本运算, 其计算复杂度为 $O(1)$ 。更新参数的过程同样只需加法和乘法运算, 因此其复杂度为 $O(1)$ 。每次迭代的总计算复杂度为 $O(2)$, 大 O 记法忽略常数项, 即 $O(1)$ 。对于凸函数, 梯度下降法的收敛速度通常是 $O(1/\epsilon)$, 其中 ϵ 为收敛精度, 因此, 计算复杂度为 $O(1) \times O(1/\epsilon) = O(1/\epsilon)$ 。

(3) 自适应深度强化学习算法:

DQN算法的计算复杂度主要来源于以下几个步骤(a)前向传播: 在每一步训练中, DQN需要通过深度神经网络预测 Q 值。对于一个有 L 层, 每层包含 q 个神经元的神经网络, 每层的计算复杂度为 $O(q^2)$, 前向传播的总复杂度为 $O(L \cdot q^2)$ 。(b)经验

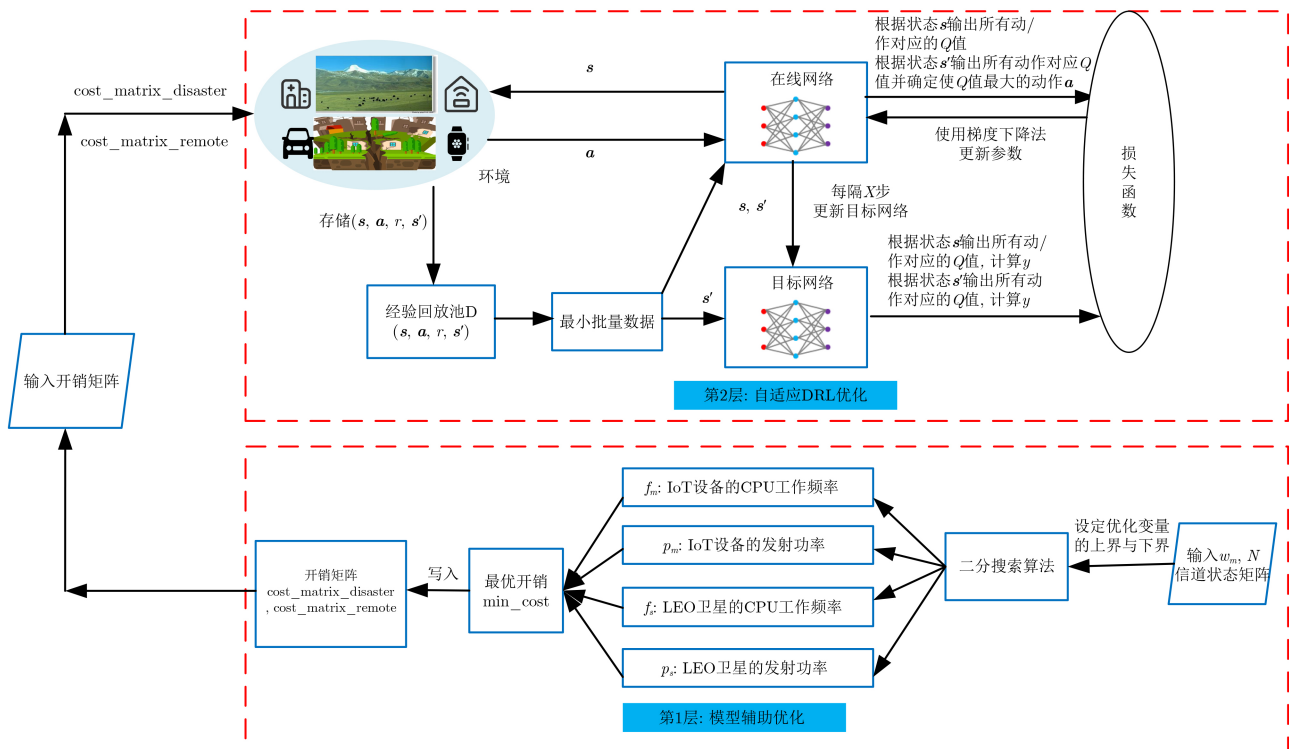


图2 MADRL 算法流程图

回放：在训练过程中，每个样本都进行一次前向传播和反向传播，从经验回放池中提取一个批量大小为 n_b 的样本，提取样本的复杂度为 $O(n_b)$ 。(c)反向传播：反向传播的复杂度与前向传播相同。因此，处理一个批次的复杂度为 $O(2 \cdot n_b \cdot L \cdot q^2)$ 。忽略常数项，因此每个回合DQN算法的复杂度为 $O(n_b \cdot L \cdot q^2)$ 。算法共执行 n_ep 个回合，算法的总复杂度为 $O(n_ep \cdot n_b \cdot L \cdot q^2)$ 。DDQN算法较DQN多出了一次目标 Q 网络的前向传播，但不涉及额外的反向传播，因此DDQN算法的总复杂度也为 $O(n_ep \cdot n_b \cdot L \cdot q^2)$ 。自适应深度强化学习算法能够根据不同地区的需求灵活选择DQN算法或DDQN算法，算法的总体复杂度也为 $O(n_ep \cdot n_b \cdot L \cdot q^2)$ ，与前述两种算法一致。

综上所述，MADRL算法总复杂度为 $O(\log_2 F_m^{\max}) + O(1/\epsilon) + O(n_ep \cdot n_b \cdot L \cdot q^2)$ 。

4 仿真与性能分析

本节通过仿真分析来评估所提算法的性能。首先设置仿真参数。其次，通过实验说明参数对学习过程的影响并进行分析。最后，将所提算法与基准算法的性能进行了比较。

4.1 仿真设置

表2提供了仿真中使用的主要参数设置。基于激光链路部署，ISL的通信容量设置为10 Gbit/(s·Hz)^[9]。

本文将提出的MADRL算法与3种基准算法进行比较：全边缘计算(Full Edge Computing, FEC)：所有任务都卸载到边缘进行计算；全云计算(Full Cloud Computing, FCC)：所有任务都卸载到云端进行计算；DQN算法：通过DQN算法优化任务卸载策略；基于分布式深度学习的动态优化^[33](Distributed Deep Learning-based Dynamic Offloading, DDLDO)算法。

4.2 参数分析

本节通过仿真实验比较了学习率对算法收敛性能的影响。奖励值的稳定表明算法开始收敛，本文将奖励值定义为开销的负数。

图3显示，不同学习率下奖励值随迭代次数的变化情况。当学习率为0.1或0.01时，奖励值波动剧烈，训练效果较差；当学习率降至0.001或0.000 1时，收敛性能显著提高，奖励值波动较小。

图4显示了不同学习率下算法累计开销随时间的变化。当学习率为0.001时，系统开销最低，说明此学习率在网络更新速度和结果精度间达到了平衡。学习率过大会导致结果迅速收敛于较差值，过小则更新缓慢，需更多迭代，因此学习率应适当选择。

图5展示了不同探索衰减因子下算法累计开销

的变化。当衰减因子为0.95或0.999 5时，系统开销高于衰减因子为0.995。本文选择衰减因子为0.995的系统开销，以平衡探索与利用，既保证训练初期的充分探索，又能在后期高效利用已学策略，快速收敛至较优奖励值。

4.3 性能对比

接下来测试了不同卸载算法在不同指标下的性能，以验证本文所提MADRL算法的有效性。

图6展示了1个调度周期内，考虑与不考虑卫星

表2 主要参数设置

参数	值
灾区设备数 D	300
远郊设备数 R	5
卫星服务范围半径 r	1 400 km
任务大小 d_m^n	[1e2,1e3,1e4,1e5,1e6] bit
任务负载 c_m^n	[1,1.5] kcycle/bit
最大容忍时延 $T_m^{n,\max}$	[0.05,0.1] s
电气系数 ϵ	10^{-28}
信道带宽 B	10 MHz
天线增益 G	20 dBi
噪声温度 T	290 K
IoT设备 m 的最大能耗 $E_m^{\max}$	5 W
LEO卫星 s 的最大能耗 $E_s^{\max}$	2 000 W
云计算中心单核CPU工作频率 f_c	1.45 GHz
云计算中心核心数 N_c	256

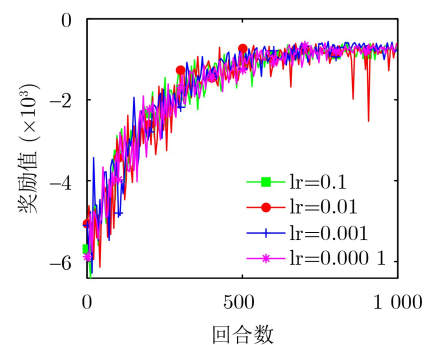


图3 不同学习率下的收敛性能

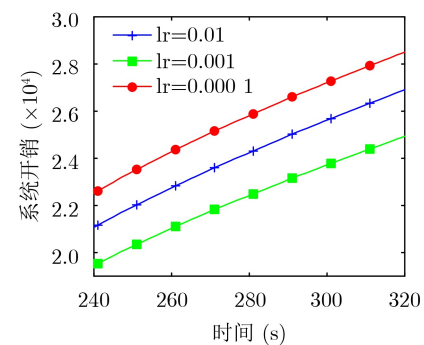


图4 不同学习率下的系统累积开销

移动性情况下各算法的系统累计开销变化。 M 为移动性(Mobility)的缩写,不考虑卫星移动性时,固定的资源分配导致资源浪费和系统开销增加。相比之下,考虑卫星移动性可使系统开销降低41%,因其动态调整资源分配,提高效率。本文提出的MADRL算法系统开销始终低于DQN和DDLDO算法,能实时监测用户需求并根据卫星位置灵活优化资源分配与任务卸载,展现出在动态环境中的显著优越性。

图7展示了1个调度周期内,考虑与不考虑ISC情况下各算法的系统累计开销变化。不考虑ISC时,任务卸载到接入卫星处理,导致时延和能耗增加。采用ISC技术可降低系统开销22.1%,因其提升了边缘计算性能。此外,本文提出的MADRL算法系统开销始终低于DQN和DDLDO算法,进一步体现了其优越性。

图8展示了系统累计开销随时间变化的过程。从时间 $t=240$ s开始,所有算法的系统开销随时间逐渐升高,本文提出的MADRL算法始终保持最低开销,与DDLDO相比降低3%,与FEC和FCC相比分别降低1.3%。这表明MADRL算法能够更高效地进行任务卸载决策。

图9展示了系统开销与LEO卫星移动速率的关系。随着卫星高度增加,移动速率降低,通信距离和信号传播时延增加,导致系统开销上升。尤其在云计

算环境中,较高的卫星高度显著增加传输时延,影响系统性能。当卫星速率为7.255 8 km/s、高度达1 200 km时,FEC因距离用户更近,传输时延显著减少,开销低于FCC,而随着卫星高度降低,卫星速率增加,FCC的性能优于FEC。图中显示,随着卫星移动速率增加,本文提出的MADRL算法在系统开销上始终优于其他算法,体现了其自适应能力。MADRL通过动态调整参数优化性能,在不同操作环境下均能保持较低的系统开销。

5 结束语

本研究提出了一种创新的动态卫星物联网模型,该模型融合了物联网终端、LEO卫星和云计算中心,以解决远郊及灾区物联网设备的时延和能耗问题。通过引入ISC技术提升卫星边缘性能,并综合考虑

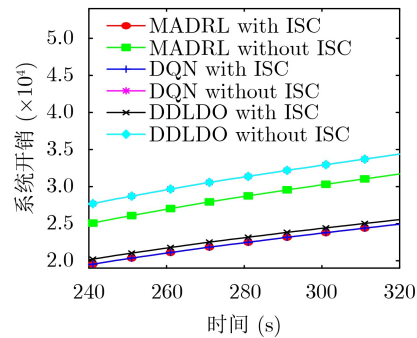


图7 ISC对DQN, MADRL算法性能的影响

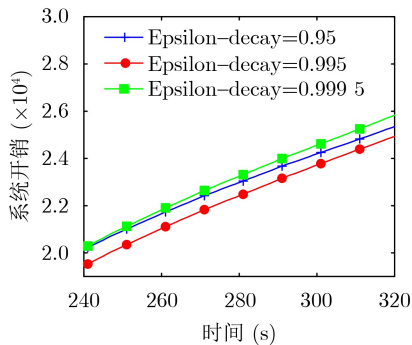


图5 不同衰减因子下的系统累积开销

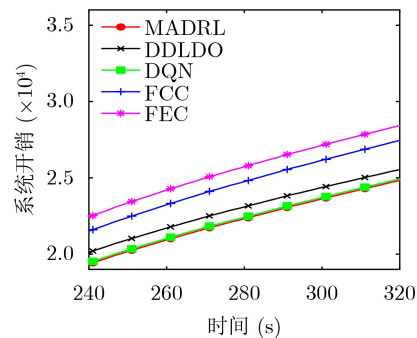


图8 不同算法下的累积系统开销

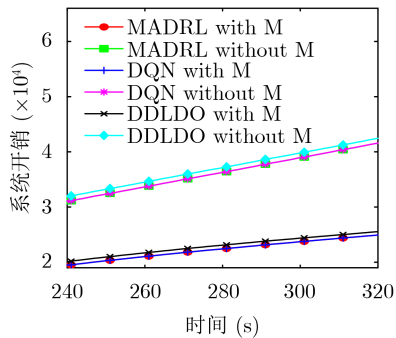


图6 卫星移动性对DQN, MADRL算法性能的影响

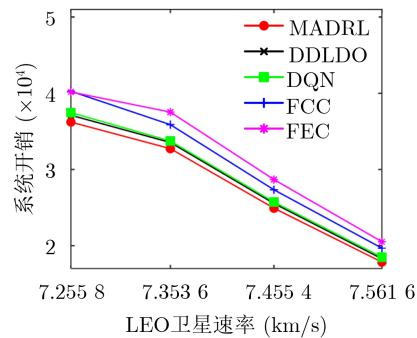


图9 不同算法下系统开销与LEO卫星速率的关系

卫星移动性，构造了最小化系统开销的任务卸载和资源分配问题。通过引入一种MADRL算法，本文有效地实现了任务卸载决策与资源配置的最优化。仿真结果显示，该模型和算法有效降低了系统开销，其中引入卫星移动性和ISC技术分别减少41%和22.1%的开销。与基准算法相比，MADRL在多种测试环境下均表现出更优性能，展现显著的应用优势。未来可进一步扩展状态空间复杂性，探索边缘缓存与边缘学习技术的结合，以更贴近真实SIoT场景。

参 考 文 献

- [1] 工业和信息化部. 物联网新型基础设施建设三年行动计划(2021-2023年)[R]. 2021.
Ministry of Industry and Information Technology of the People's Republic of China. A three-year action plan for the construction of new IoT infrastructure[R]. 2021.
- [2] CUI Gaofeng, DUAN Pengfei, XU Lexi, *et al.* Latency optimization for hybrid GEO-LEO satellite-assisted IoT networks[J]. *IEEE Internet of Things Journal*, 2023, 10(7): 6286–6297. doi: [10.1109/JIOT.2022.3222831](https://doi.org/10.1109/JIOT.2022.3222831).
- [3] DE COLA T and BISIO I. QoS optimisation of eMBB services in converged 5G-satellite networks[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(10): 12098–12110. doi: [10.1109/TVT.2020.3011963](https://doi.org/10.1109/TVT.2020.3011963).
- [4] KANEKO K, NISHIYAMA H, KATO N, *et al.* Construction of a flexibility analysis model for flexible high-throughput satellite communication systems with a digital channelizer[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(3): 2097–2107. doi: [10.1109/TVT.2017.2736010](https://doi.org/10.1109/TVT.2017.2736010).
- [5] BOERO L, BRUSCHI R, DAVOLI F, *et al.* Satellite networking integration in the 5G ecosystem: Research trends and open challenges[J]. *IEEE Network*, 2018, 32(5): 9–15. doi: [10.1109/MNET.2018.1800052](https://doi.org/10.1109/MNET.2018.1800052).
- [6] CHIEN W C, LAI C F, HOSSAIN M S, *et al.* Heterogeneous space and terrestrial integrated networks for IoT: Architecture and challenges[J]. *IEEE Network*, 2019, 33(1): 15–21. doi: [10.1109/MNET.2018.1800182](https://doi.org/10.1109/MNET.2018.1800182).
- [7] RANAWEERA P, JURCUT A D, and LIYANAGE M. Survey on multi-access edge computing security and privacy[J]. *IEEE Communications Surveys & Tutorials*, 2021, 23(2): 1078–1124. doi: [10.1109/COMST.2021.3062546](https://doi.org/10.1109/COMST.2021.3062546).
- [8] LI Chengcheng, ZHANG Yasheng, XIE Renchao, *et al.* Integrating edge computing into low earth orbit satellite networks: Architecture and prototype[J]. *IEEE Access*, 2021, 9: 39126–39137. doi: [10.1109/ACCESS.2021.3064397](https://doi.org/10.1109/ACCESS.2021.3064397).
- [9] WANG Dezhi, WANG Wei, KANG Yuhan, *et al.* Distributed data offloading in ultra-dense LEO satellite networks: A stackelberg mean-field game approach[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2023, 17(1): 112–127. doi: [10.1109/JSTSP.2022.3226400](https://doi.org/10.1109/JSTSP.2022.3226400).
- [10] TANG Zhixuan, YU Kai, YANG Guannan, *et al.* New bridge to cloud: An ultra-dense LEO assisted green computation offloading approach[J]. *IEEE Transactions on Green Communications and Networking*, 2023, 7(2): 552–564. doi: [10.1109/TGCN.2022.3208819](https://doi.org/10.1109/TGCN.2022.3208819).
- [11] YU Shuai, GONG Xiaowen, SHI Qian, *et al.* EC-SAGINs: Edge-computing-enhanced space-air-ground-integrated networks for internet of vehicles[J]. *IEEE Internet of Things Journal*, 2022, 9(8): 5742–5754. doi: [10.1109/JIOT.2021.3052542](https://doi.org/10.1109/JIOT.2021.3052542).
- [12] LIU Yi, JIANG Li, QI Qi, *et al.* Energy-efficient space-air-ground integrated edge computing for internet of remote things: A federated DRL approach[J]. *IEEE Internet of Things Journal*, 2023, 10(6): 4845–4856. doi: [10.1109/JIOT.2022.3220677](https://doi.org/10.1109/JIOT.2022.3220677).
- [13] TANG Qingqing, FEI Zesong, LI Bin, *et al.* Stochastic computation offloading for LEO satellite edge computing networks: A learning-based approach[J]. *IEEE Internet of Things Journal*, 2024, 11(4): 5638–5652. doi: [10.1109/JIOT.2023.3307707](https://doi.org/10.1109/JIOT.2023.3307707).
- [14] ZHU Xiangming and JIANG Chunxiao. Delay optimization for cooperative multi-tier computing in integrated satellite-terrestrial networks[J]. *IEEE Journal on Selected Areas in Communications*, 2023, 41(2): 366–380. doi: [10.1109/JSAC.2022.3227083](https://doi.org/10.1109/JSAC.2022.3227083).
- [15] ZHANG Shanghong, CUI Gaofeng, LONG Yating, *et al.* Joint computing and communication resource allocation for satellite communication networks with edge computing[J]. *China Communications*, 2021, 18(7): 236–252. doi: [10.23919/JCC.2021.07.019](https://doi.org/10.23919/JCC.2021.07.019).
- [16] TANG Qingqing, FEI Zesong, LI Bin, *et al.* Computation offloading in LEO satellite networks with hybrid cloud and edge computing[J]. *IEEE Internet of Things Journal*, 2021, 8(11): 9164–9176. doi: [10.1109/JIOT.2021.3056569](https://doi.org/10.1109/JIOT.2021.3056569).
- [17] CAO Bin, ZHANG Jintong, LIU Xin, *et al.* Edge-cloud resource scheduling in space-air-ground-integrated networks for internet of vehicles[J]. *IEEE Internet of Things Journal*, 2022, 9(8): 5765–5772. doi: [10.1109/JIOT.2021.3065583](https://doi.org/10.1109/JIOT.2021.3065583).
- [18] LI Zhipeng, LI Meng, and WANG Qian. Predator-prey model based asymmetry resource allocation in satellite-terrestrial network[J]. *Symmetry*, 2021, 13(11): 2113. doi: [10.3390/sym13112113](https://doi.org/10.3390/sym13112113).
- [19] LEE Y and CHOI J P. Connectivity analysis of mega-constellation satellite networks with optical intersatellite links[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2021, 57(6): 4213–4226. doi: [10.1109/TAES.2021.3090914](https://doi.org/10.1109/TAES.2021.3090914).
- [20] ZHANG Hangyu, LIU Rongke, KAUSHIK A, *et al.* Satellite

- edge computing with collaborative computation offloading: An intelligent deep deterministic policy gradient approach[J]. *IEEE Internet of Things Journal*, 2023, 10(10): 9092–9107. doi: [10.1109/JIOT.2022.3233383](https://doi.org/10.1109/JIOT.2022.3233383).
- [21] GAO Xiangqiang, HU Yingmeng, SHAO Yingzhao, *et al.* Hierarchical dynamic resource allocation for computation offloading in LEO satellite networks[J]. *IEEE Internet of Things Journal*, 2024, 11(11): 19470–19484. doi: [10.1109/JIOT.2024.3367937](https://doi.org/10.1109/JIOT.2024.3367937).
- [22] 易必杰. 面向空天地一体化网络的计算卸载策略研究[D]. [博士/硕士论文], 西安电子科技大学, 2023. doi: [10.27389/d.cnki.gxadu.2023.002544](https://doi.org/10.27389/d.cnki.gxadu.2023.002544).
- YI Bijie. Research on computing offloading strategy for space-air-ground integrated network[D]. Xidian University, 2023. doi: [10.27389/d.cnki.gxadu.2023.002544](https://doi.org/10.27389/d.cnki.gxadu.2023.002544).
- [23] FANG Hai, JIA Yangyang, WANG Yuanle, *et al.* Matching game based task offloading and resource allocation algorithm for satellite edge computing networks[C]. Proceedings of 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, 2022: 1–5, doi: [10.1109/ISNCC55209.2022.9851813](https://doi.org/10.1109/ISNCC55209.2022.9851813).
- [24] 郭子桢, 梁俊, 肖楠, 等. 软件定义卫星网络多控制器可靠部署算法[J]. 西安交通大学学报, 2021, 55(2): 158–165. doi: [10.7652/xjtub202102019](https://doi.org/10.7652/xjtub202102019).
- GUO Zizhen, LIANG Jun, XIAO Nan, *et al.* Multi-controller reliable deployment algorithm for software defined satellite network[J]. *Journal of Xi'an Jiaotong University*, 2021, 55(2): 158–165. doi: [10.7652/xjtub202102019](https://doi.org/10.7652/xjtub202102019).
- [25] KUROSE J and ROSS K. Computer Networking: A Top-Down Approach[M]. 6th ed. Boston: Pearson, 2012.
- [26] 谢希仁. 计算机网络[M]. 4版. 大连: 大连理工大学出版社, 2003.
- XIE Xiren. Computer Networking[M]. 4th ed. Dalian: Dalian University of Technology Press, 2003.
- [27] QI Xiaoxin, ZHANG Bing, QIU Zhiliang, *et al.* Using inter-mesh links to reduce end-to-end delay in walker delta constellations[J]. *IEEE Communications Letters*, 2021, 25(9): 3070–3074. doi: [10.1109/LCOMM.2021.3095227](https://doi.org/10.1109/LCOMM.2021.3095227).
- [28] YOU Changsheng, HUANG Kaibin, and CHAE H. Energy efficient mobile cloud computing powered by wireless energy transfer[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(5): 1757–1771. doi: [10.1109/JSAC.2016.2545382](https://doi.org/10.1109/JSAC.2016.2545382).
- [29] NOWAK R. Generalized binary search[C]. 2008 46th Annual Allerton Conference on Communication, Control, and Computing, Monticello, USA, 2008: 568–574. doi: [10.1109/ALLERTON.2008.4797609](https://doi.org/10.1109/ALLERTON.2008.4797609).
- [30] AVRIEL M. Nonlinear Programming: Analysis and Methods[M]. Englewood Cliffs: Prentice-Hall, 1976.
- [31] MNIH V, KAVUKCUOGLU K, SILVER D, *et al.* Human-level control through deep reinforcement learning[J]. *Nature*, 2015, 518(7540): 529–533. doi: [10.1038/nature14236](https://doi.org/10.1038/nature14236).
- [32] VAN HASSELT H, GUEZ A, and SILVER D. Deep reinforcement learning with double Q-learning[C]. The Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 2016: 2094–2100.
- [33] SHUAI Jiaqi, CUI Haixia, HE Yejun, *et al.* Dynamic satellite edge computing offloading algorithm based on distributed deep learning[J]. *IEEE Internet of Things Journal*, 2024, 11(16): 27790–27802. doi: [10.1109/JIOT.2024.3404830](https://doi.org/10.1109/JIOT.2024.3404830).
- 施建锋: 男, 副教授, 研究方向为空地一体化网络、用户为中心网络、B5G和6G网络。
- 陈忻阳: 男, 硕士生, 研究方向为物联网、卫星边缘计算、资源分配。
- 李宝龙: 男, 副教授, 研究方向为无线光通信、数据管理、物联网。

责任编辑: 余 蓉

Research on Task Offloading and Resource Allocation Algorithms in Cloud-edge-end Collaborative Computing for the Internet of Things

SHI Jianfeng CHEN Xinyang LI Baolong

(School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 214442, China)

Abstract:

Objective With the rapid pace of digital transformation and the smart upgrading of the economy and society, the Internet of Things (IoT) has become a critical element of new infrastructure. Current wide-area IoT networks primarily rely on 5G terrestrial infrastructure. While these networks continue to evolve, challenges persist, particularly in remote or disaster-affected areas. The high cost and vulnerability of base stations hinder deployment and maintenance in these locations. Satellite networks provide seamless coverage, flexibility, and

reliability, making them compelling alternatives to terrestrial networks for achieving global connectivity. Satellite-assisted Internet of Things (SIoT) can deliver ubiquitous and reliable connectivity for IoT devices. Typically, IoT devices offload tasks to edge servers or cloud platforms due to their limited power, computing, and caching resources. Mobile Edge Computing (MEC) helps reduce latency by caching content and placing edge servers closer to IoT devices. Low Earth Orbit (LEO) satellites with integrated processing units can also serve as edge computing nodes. Although cloud platforms offer abundant computing resources and a reliable power supply, the long distance between IoT devices and the cloud results in higher communication latency. With the explosive growth of IoT devices and the diversification of application requirements driven by 5G, it is essential to design a collaborative architecture that integrates cloud, edge, and end devices. Recent research has extensively explored MEC-enhanced SIoT systems. However, many studies focus solely on edge or cloud computing, with little emphasis on their integration, satellite mobility, or resource constraints. Furthermore, LEO satellites providing edge services face challenges due to their limited onboard resources and the high mobility of the satellite constellation, complicating resource allocation and task offloading. Single-satellite solutions may not satisfy performance expectations during peak demand. Inter-Satellite Collaboration (ISC) technology, which utilizes visible light communications, can significantly increase system capacity, extend coverage, reduce individual satellite resource consumption, and prolong network operational life. Although some studies address three-tier architectures involving IoT devices, satellites, and clouds, proposing load balancing mechanisms through ISC for optimizing offloading and resource allocation, many rely on static assumptions about network topologies and user associations. In practice, LEO satellites require frequent switching and dynamic adjustments in offloading strategies to maintain service quality due to their high-speed mobility. Therefore, there is a need for a method of task offloading and resource allocation in a dynamic environment that considers satellite mobility and limited resources. To address these research gaps, this paper proposes a dynamic ISC-enhanced cloud-edge-end SIoT network model. By formulating the joint optimization problem of offloading decisions and resource allocation as a Mixed Integer Non-Linear Programming (MINLP) problem, a Model-assisted Adaptive Deep Reinforcement Learning (MADRL) algorithm is developed to achieve minimum system cost in a changing environment.

Methods The LEO satellite mobility model and the SIoT network model with ISC are constructed to analyze end-to-end latency and system energy consumption. This evaluation considers three modes: local computing, edge computing, and cloud computing. A joint optimization MINLP problem is formulated, focusing on task offloading and resource allocation to minimize system costs. A MADRL algorithm is introduced, integrating traditional optimization techniques with deep reinforcement learning. The algorithm operates in two parts. The first part optimizes communication and computational resource allocation using a model-assisted binary search algorithm and gradient descent method. The second part trains a Q-network to adapt offloading decisions based on stochastic task arrivals through an adaptive deep reinforcement learning approach.

Results and Discussions Simulation experiments were conducted under various dynamic scenarios. The MADRL algorithm exhibits strong convergence properties, as demonstrated in the analysis. Comparisons of different learning rates and exploration decay factors reveal optimal parameter values. Incorporating satellite mobility reduces system costs by 41% compared to static scenarios, enabling dynamic resource allocation and improved efficiency. Integrating ISC reduces system costs by 22.1%. This demonstrates the effectiveness of inter-satellite load balancing in improving resource utilization. Additionally, the MADRL algorithm achieves a 3% reduction in system costs compared to the Deep Q Learning (DQN) algorithm, highlighting its adaptability and efficiency in dynamic environments. System costs decrease as satellite speed increases, with the MADRL algorithm consistently outperforming other methods.

Conclusions This paper presents an innovative dynamic SIoT model that integrates IoT devices, LEO satellites, and a cloud computing center. The model addresses the latency and energy consumption issues faced by IoT devices in remote and disaster-stricken areas. The task offloading and resource allocation problem that minimizes system cost is constructed by incorporating ISC techniques to enhance satellite edge performance and by taking satellite mobility into account. A MADRL algorithm that combines traditional optimization with deep reinforcement learning is proposed. This approach effectively optimizes task offloading decisions and resource allocation. Simulation results demonstrate that our model and algorithm significantly reduce system costs. Specifically, the incorporation of satellite mobility and ISC technology leads to cost reductions of 41% and 22.1%, respectively. Compared to benchmark algorithms, the MADRL shows superior performance across various test environments, highlighting its significant application advantages.

Key words: Cloud-edge-end collaborative computing; Satellite-assisted Internet of Things (SIoT); Deep Reinforcement Learning (DRL); Task offloading; Resource allocation.