

基于复杂依赖关系分解的低轨卫星网络任务卸载策略

申运波¹, 何杏宇^{1,2}, 杨桂松^{1*}

¹上海理工大学光电信息与计算机工程学院, 上海

²上海理工大学出版学院, 上海

收稿日期: 2025年4月16日; 录用日期: 2025年5月9日; 发布日期: 2025年5月16日

摘要

地面设备受限于本地计算资源,需将计算任务卸载至低轨卫星网络以降低处理时延。在任务卸载过程中,为了突破单个低轨卫星的计算资源限制,则需要将计算任务拆解成多个子任务。然而任务内部复杂的依赖关系给任务分解带来极大挑战。为此,文章提出了一种基于复杂依赖关系分解的任务卸载策略。首先,利用区域控制器将网络划分为控制域,统筹协调域内外计算资源,并负责任务卸载。其次,将具有复杂依赖关系的任务建模为有向无环图(DAG),并采用图卷积网络(GCN)对复杂依赖关系进行特征提取与分解。然后,将任务卸载问题构建为马尔可夫决策过程(MDP),利用深度Q网络(DQN)算法求解最优卸载策略。最后,实验结果表明,该策略能够有效降低任务处理时延。

关键词

低轨卫星, 任务卸载, 深度强化学习, 依赖关系, 图卷积网络

Task Offloading Strategy for LEO Satellite Networks Based on Complex Dependency Decomposition

Yunbo Shen¹, Xingyu He^{1,2}, Guisong Yang^{1*}

¹School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

²College of Publishing, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 16th, 2025; accepted: May 9th, 2025; published: May 16th, 2025

*通讯作者。

文章引用: 申运波, 何杏宇, 杨桂松. 基于复杂依赖关系分解的低轨卫星网络任务卸载策略[J]. 建模与仿真, 2025, 14(5): 417-430. DOI: 10.12677/mos.2025.145404

Abstract

Due to the limitations of local computing resources on ground devices, computation tasks are offloaded to Low Earth Orbit (LEO) satellite networks to reduce processing delays. During the task offloading process, to overcome the computing resource constraints of single LEO satellites, a complex task needs to be decomposed into multiple subtasks. However, complex dependency relationships within the tasks pose significant challenges for task decomposition. To address this problem, this paper proposes a task-offloading strategy based on the decomposition of complex dependency relationships. Firstly, a regional controller is introduced to divide a network into control domains, which coordinates computing resources within and across domains in a unified manner and is responsible for task offloading. Secondly, tasks with complex dependencies are modeled as directed acyclic graphs (DAG), and graph convolutional networks (GCN) are utilized to model and extract features from these complex dependencies. Then, the task offloading problem is formulated as a Markov decision process (MDP), and the deep Q network (DQN) algorithm is used to search for the optimal offloading strategy. Finally, experimental results demonstrate that this strategy can effectively reduce the delay in task processing.

Keywords

Low Earth Orbit Satellite, Task Offloading, Deep Reinforcement Learning, Dependency, Graph Convolution Network

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着物联网与 5G 技术的推动,大量计算密集型任务出现[1][2],如对于智慧医疗、智慧交通和工业互联网而言,这些任务对计算资源要求较高,而自身设备因计算能力有限,难以高效处理。

为减轻地面设备计算负担,现有研究利用地面网络对任务进行卸载,例如文献[3]通过车载终端将计算任务卸载至地面边缘服务器,减少计算负担,提高系统性能;文献[4]将地面物联网终端产生的计算任务卸载至云中心、农业服务器和移动边缘计算 MEC 服务器,最小化系统的总长期成本。

为支持偏远地区无地面网络区域的设备进行任务卸载,现有研究将任务通过卫星中继传输到地面云计算中心,例如文献[5]提出一种用于物联网网络的卫星 MEC 模型,将地面设备产生的计算任务通过卫星网络中继到地面云计算中心;文献[6]针对卫星边缘计算网络场景,提出一种新的卫星边缘计算卸载策略,实现卫星与地面云计算中心之间的协同资源共享。

通过卫星中转至地面云计算中心会导致传输时延长等问题[7],为减少任务传输时延,将计算能力下沉到卫星网络内部[8]。现有研究将任务直接卸载至低轨卫星网络,例如文献[9]为降低总任务处理时延,利用星地协同边缘计算架构,将计算任务卸载至卫星网络。文献[10]提出一种基于深度强化学习的任务卸载决策方法,优化在低轨卫星网络中的任务处理时间。文献[11]提出一种基于 DDPG 的低延迟卸载策略,优化任务处理时延。文献[12]研究卫星边缘计算中的任务卸载问题,提出一种多智能体算法,通过集中训练并分布式执行,有效降低了任务处理时延。

相较于地面基站而言,LEO 卫星和地面用户之间的距离是处于变化状态的,而这种距离上的变化很

可能会引发更高的传输延迟情况出现。并且, 由于受到功耗、体积以及载重等多方面因素的限制, LEO 卫星所具备的计算资源较稀缺, 要想满足全部用户的任务卸载需求存在较大难度。因此, 高效地运用整体 LEO 卫星星座有限的资源, 对任务卸载策略进行优化, 已然成为一个全新的挑战。此外, 在实际应用中, 诸多任务存在依赖关系(例如卫星遥感数据的采集与预处理), 各子任务需要以前一子任务输出为依赖, 且仅当全部子任务完成时, 整个任务才算完结, 这无疑加剧了任务卸载策略制定的复杂性。再者, 卫星网络覆盖范围广, 加之任务本身具有复杂依赖关系, 致使在大规模且多约束的应用场景中, 求解相关问题的难度显著提升。因此, 在动态变化的低轨卫星网络环境中, 将具有复杂依赖关系任务的非线性 DAG 拓扑结构转换为一组线性且可并行处理的任务集合, 并实现全域网络卸载是个难题。

为了解决上述难题, 本研究旨在处理低轨卫星网络中具有复杂依赖关系任务卸载的问题, 设计适应网络动态变化和任务非线性变化的卸载策略, 充分利用卫星网络的全局计算能力, 降低任务处理时延。本文的主要贡献如下: 1) 基于有向无环图 DAG 对任务内部依赖关系进行建模, 为高效处理依赖任务的拓扑信息, 采用图神经网络 GCN 学习 DAG 特征, 将具有复杂依赖关系的任务进行分解, 以时延优化为目标构建任务卸载模型; 2) 为有效卸载各子任务, 利用深度 Q 网络算法, 将每个区域控制器视为独立智能体, 使其自主处理状态和动作, 以应对网络空间的复杂性。

2. 系统模型和问题描述

如图 1 所示, 本文将低轨卫星网络划分为 H 个控制域, 各控制域含 K 颗配备 MEC 服务器的卫星, 控制器为控制域中心卫星, 负责处理任务请求及卸载, 能有效处理请求并利用网络全局计算能力处理复杂任务。

本文聚焦地面网络无法覆盖的远程场景, 服务地面网关及卫星直连区域。地面终端设备 GU 产生具有复杂依赖关系任务后发送至最近低轨卫星, 卫星接收请求后转至所在控制域控制器, 控制器依网络状态与任务需求指定各子任务卸载目标区域, 再将结果返回卫星, 卫星把子任务传至对应卸载区域的控制器。控制器接收子任务卸载请求后, 依据网络资源、计算能力及负载情况, 决策子任务在控制域内的具体卸载目标, 保障任务高效执行与资源合理利用。

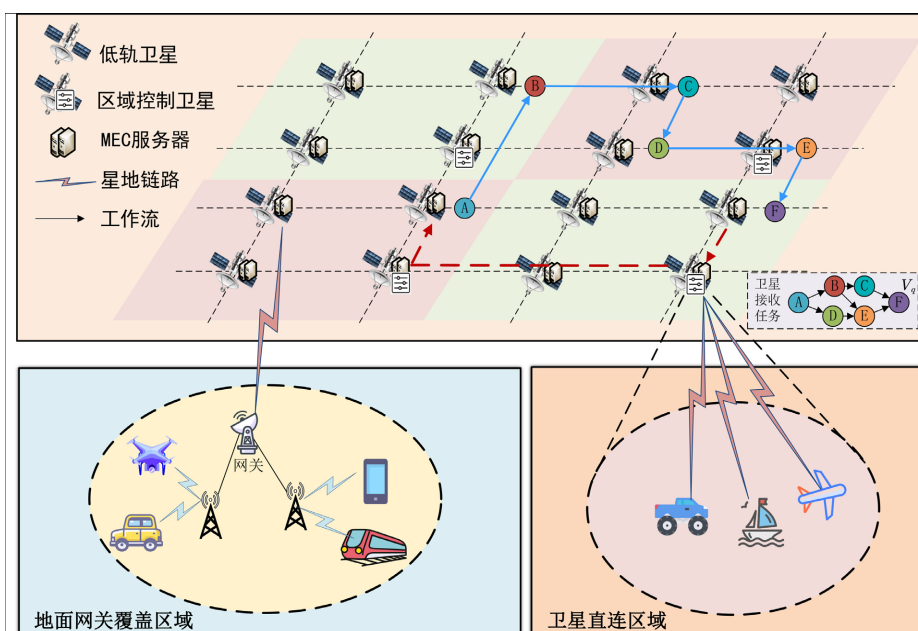


Figure 1. Computing architecture of LEO satellite network

图 1. 低轨卫星网络计算架构

为了便于表达, 本研究定义控制域集合为 $SN = \{SN_1, SN_2 \cdots SN_h \cdots SN_H\}$, 其中域内卫星集合 $SN_h = \{n_{h,1} \cdots n_{h,k} \cdots n_{h,K}\}$, 使 $SN_h \in N$ 。 $GU = \{gu_1, \cdots, gu_r, \cdots, gu_R\}$ 表示地面终端设备集合。 $T = \{t_1 \cdots t_i \cdots t_L\}$ 表示时间集合, $VS = \{V_1, V_2 \cdots V_q \cdots V_Q\}$ 是任务集合。此外, $(n_{h,j}, n_{h,k})$ 是从卫星 $n_{h,j}$ 到 $n_{h,k}$ 的星间链路, 链路 $(gu_r, n_{h,k})$ 是从地面终端设备 gu_r 到卫星 $n_{h,k}$ 的星地链路。

2.1. 链路通信模型

在时隙 t_i , 计算星间链路 ISL 从卫星 $n_{h,j}$ 到卫星 $n_{h,k}$ 的传输速率表示为:

$$C_i^{(h,j),(h,k)} = \frac{P_{t_i}^{h,j} \cdot G_{tr}^{h,j} \cdot G_{re}^{h,k} \cdot L_t \cdot Lp_i^{(h,j),(h,k)}}{K \cdot T \cdot \left(\frac{Eb}{No}\right)_{req} \cdot LM} \quad (1)$$

$$Lp_i^{(h,j),(h,k)} = \left(\frac{c}{4\pi \cdot sl_i^{(h,j),(h,k)} \cdot F} \right)^2 \quad (2)$$

其中 $P_{t_i}^{h,j}$ 为在 t_i 时刻卫星 $n_{h,j}$ 的传输功率, $G_{tr}^{h,j}$ 和 $G_{re}^{h,k}$ 表示卫星 $n_{h,j}$ 和卫星 $n_{h,k}$ 的传输天线增益和接收天线增益, L_t 是其他链路损耗, $Lp_i^{(h,j),(h,k)}$ 为在 t_i 时刻卫星 $n_{h,j}$ 到卫星 $n_{h,k}$ 星间的自由空间损耗, c 为光速, $sl_i^{(h,j),(h,k)}$ 为 t_i 时刻卫星 $n_{h,j}$ 到卫星 $n_{h,k}$ 的链路长度, F 为通信频率, K 为玻尔兹曼常数, T 为噪声温度, $(Eb/No)_{req}$ 为所需的每比特接收能量与一定噪声密度的比值, LM 为链路余量。

将任务传输时延表述为星地链路传输时延和星间链路传输时延, 其中星间链路传输时延表示为

$$td_{q,n} = \sum_{h=1}^H \sum_{j=1}^K \sum_{k=1}^K \left[M_{(h,j),(h,k)} \cdot \left(\frac{sl_i^{(h,j),(h,k)}}{c} \right) + \frac{\sum_{n=1}^U \varphi_{q,n}^{(h,j),(h,k)} \cdot v_{q,n}}{C_i^{(r,k)}} \right] \quad (3)$$

其中 $M_{(h,j),(h,k)}$ 表示星间链路 $sl_i^{(h,j),(h,k)}$ 是否传输任务, $M_{(h,j),(h,k)} \in \{0,1\}$ 。 $\varphi_{q,n}^{(h,j),(h,k)}$ 表示任务 $v_{q,n}$ 是否占用链路 $sl_i^{(h,j),(h,k)}$, $\varphi_{q,n}^{(h,j),(h,k)} \in \{0,1\}$ 。星地链路传输时延表示为:

$$td_q^{(r),(h,k)} = \frac{\sum_{n=1}^U v_{q,n}}{C_i^{(r),(h,k)}} \quad (4)$$

其中 $C_i^{(r),(h,k)}$ 为第 t_i 时隙地面设备 gu_r 到卫星 $n_{h,k}$ 的速率。

2.2. 卫星任务队列模型

每个 LEO 卫星具有任务队列来存储卸载的任务。在时隙 t_i 开始时, $at_{h,k}(i)$ 表示由卫星 $n_{h,k}$ 执行卸载的任务:

$$at_{h,k}(i) = \sum_{n=1}^U (\omega_{h,k}^{q,n} \cdot v_{q,n}) \quad (5)$$

其中 $\omega_{h,k}^{q,n} \in \{0,1\}$ 表示任务 $v_{q,n}$ 是否卸载到卫星 $n_{h,k}$ 的决策。 $ct_{h,k}(i)$ 表示在时隙 t_i 卫星 $n_{h,k}$ 已经计算完成的数据量。在 t_{i+1} 时隙卫星 $n_{h,k}$ 的任务队列表示为:

$$Q_{h,k}^F(i+1) = \max\{Q_{h,k}^F(i) - ct_{h,k}(i), 0\} + at_{h,k}(i) \quad (6)$$

其中 $Q_{h,k}^F(0) = 0$ 。在时隙 t_{i+1} , 卫星 $n_{h,k}$ 的剩余能源表示为:

$$be_{i+1}^{h,k} = \min\{be_i^{h,k} - ce_n^{h,k} + e_i^{h,k}, be_{sa}^{max}\} \quad (7)$$

其中 $e_i^{h,k}$ 为卫星 $n_{h,k}$ 在时隙 t_i 收获的能量, be_{sa}^{max} 表示卫星最大能量存储值。

2.3. 卫星计算模型

子任务 $v_{q,n}$ 在卫星 $n_{h,k}$ 的计算延迟表示为:

$$cd_{q,n}^{h,k} = \frac{c_{q,n} \cdot sd_{q,n}}{f_{q,n}^{h,k}} \quad (8)$$

其中 $f_{q,n}^{h,k}$ 是卫星 $n_{h,k}$ 为 $v_{q,n}$ 任务分配的计算资源。将子任务 $v_{q,n}$ 排队延迟表示为:

$$qd_{q,n}^{h,k} = t_{q,n}^s - t_{q,n}^a \quad (9)$$

其中 $t_{q,n}^a$ 为子任务 $v_{q,n}$ 到达的时间, $t_{q,n}^s$ 为子任务 $v_{q,n}$ 开始执行的时间。子任务 $v_{q,n}$ 在卫星 $n_{h,k}$ 的计算能耗表示为:

$$ce_{q,n}^{h,k} = \sum_{h=1}^H \sum_{k=1}^K \left[(1 - \varphi_{h,k}^{q,n}) \mu_{mob}^{h,k} \cdot (f_{q,n}^{h,k})^3 \cdot cd_{q,n}^{h,k} + \varphi_{h,k}^{q,n} P t_i^{h,k} \right] \quad (10)$$

其中 $\mu_{mob}^{h,k}$ 表示卫星 $n_{h,k}$ 的有效能量系数, $f_{q,n}^{h,k}$ 是 $n_{h,k}$ 分配给子任务 $v_{q,n}$ 的计算能力。 $\varphi_{h,k}^{q,n} \in \{0,1\}$ 是传输标志, 当其等于 0 时表示在卫星 $n_{h,k}$ 上为子任务 $v_{q,n}$ 做计算, 为 1 表示子任务 $v_{q,n}$ 途径卫星 $n_{h,k}$ 做传输, $P t_i^{h,k}$ 为第 t_i 时隙子任务经由卫星 $n_{h,k}$ 时 $n_{h,k}$ 的发送能量消耗, 其中:

$$\sum_{h=1}^H \sum_{k=1}^K \varphi_{h,k}^{q,n} = 1$$

3. 基于复杂依赖关系分解的任务卸载策略

如图 2 所示, 本文利用 DQN 算法设计复杂依赖关系分解与任务卸载模型。该模型能够将具有复杂依赖关系的任务分解为多个子任务, 并针对每个子任务进行卸载决策。具体由 Env 模块、Training 模块和 DQN Network 模块三部分构成。

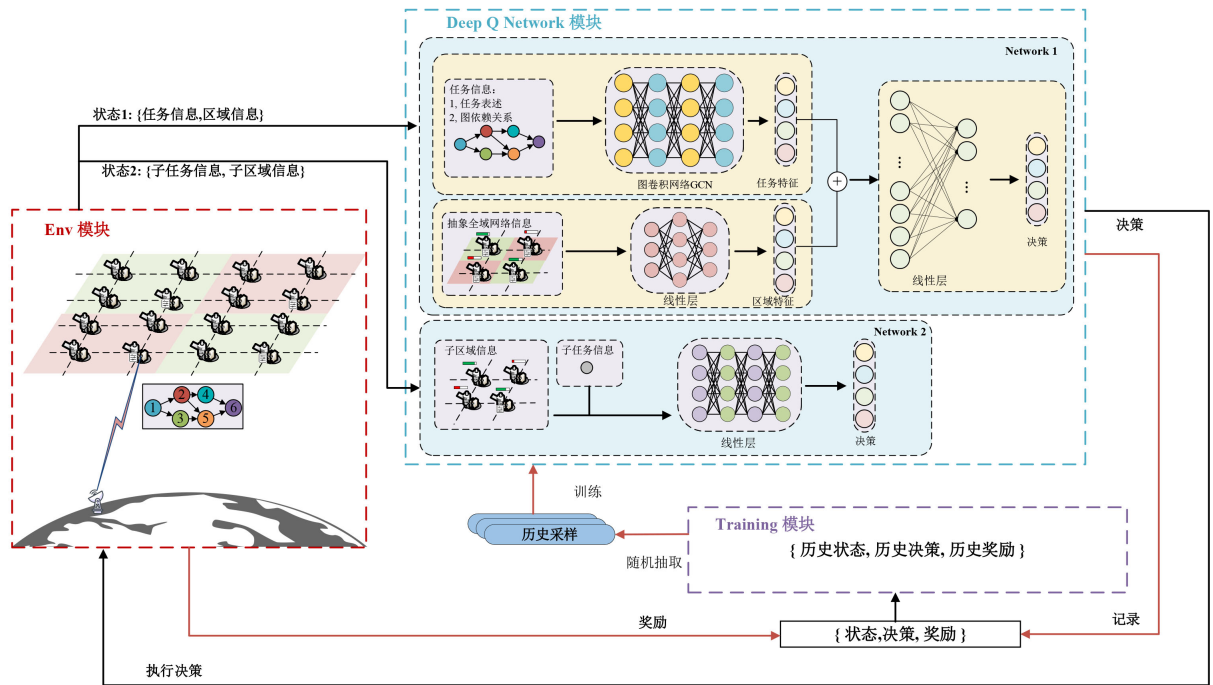


Figure 2. Structure diagram of the DQN algorithm integrated with GCN

图 2. 融合图卷积网络的深度 Q 网络算法结构图

在 Env 模块中, 任务依赖关系被建模为有向图, 节点表示任务, 边表示任务间的数据依赖。每个任务节点具有任务数据大小、计算需求、容忍时延等属性, 而每颗卫星具有计算能力、通信带宽和能耗等约束条件。

在 Training 模块中, 系统实时记录每次的历史状态、决策及奖励。通过随机抽样方式反复训练, 持续优化卸载决策。

在 DQN Network 模块中, 将 GCN 嵌入 DQN Network 模块, 处理任务依赖并优化卸载决策。该模块由两个子网络组成, 分别负责子任务的卸载区域选择及区域内最终的卫星决策和资源分配。首先, 任务依赖关系建模为图结构, 其中节点表示任务, 边表示依赖关系。任务信息(任务描述及依赖关系)经 GCN 提取高阶特征, 捕获上下文及全局依赖。区域状态信息(如卫星状态、通信链路)通过神经网络提取特征并生成区域表示。任务特征与区域特征融合后, 输入全连接网络, 生成子任务的卸载目标区域。任务接收卫星依据决策将子任务发送至目标区域控制器。控制器收到子任务请求后, 通过第二子网络综合子任务需求和区域状态, 生成最终决策, 包括目标卫星选择及计算资源分配。

当 DQN Network 模块输出卸载决策后, Env 模块根据策略开始执行任务, 计算任务的执行时间、传输时间等, 并判断任务是否成功完成, 最后生成即时奖励。

3.1. 基于复杂依赖关系分析的任务分解

在 Env 模块中, 本文采用有向无环图 $G_q = (V_q, E_q)$ 表示任务之间的依赖关系, 其中 V_q 表示任务集, E_q 表示边集。为了便于描述, 为具有复杂依赖关系的任务构建 DAG 时增加 2 个虚拟任务 $v_{q,s}$ 和 $v_{q,e}$ 。 $v_{q,s}$ 表示开始任务, 没有前驱任务; $v_{q,e}$ 表示结束任务, 没有后继任务。这 2 个任务无需被卸载, 即在本地执行时所需时间为 0, 那么包含 U 个子任务的 DAG 任务集为 $V_q = (v_{q,s}, v_{q,1}, v_{q,2} \cdots v_{q,U}, v_{q,e})$ 。有向边 (v, v') 表示子任务 v 和 v' 之间存在依赖关系, 即任务 v 的输出是任务 v' 的输入。将任务 $v_{q,n}$ 建模为元组:

$$v_{q,n} = \{c_{q,n}, sd_{q,n}, \rho_{q,n}\} \quad (11)$$

其中 $sd_{q,n}$ 是任务 $v_{q,n}$ 的数据大小, $sd_{q,n}$ 遵循独立且同分布于 $[sd^{\min}, sd^{\max}]$, $c_{q,n}$ 表示计算任务 $v_{q,n}$ 每 bit 所需的 CPU 周期数, 遵循独立且同分布于 $[c^{\min}, c^{\max}]$, $\rho_{q,n}$ 是表示任务的最大容忍时延。

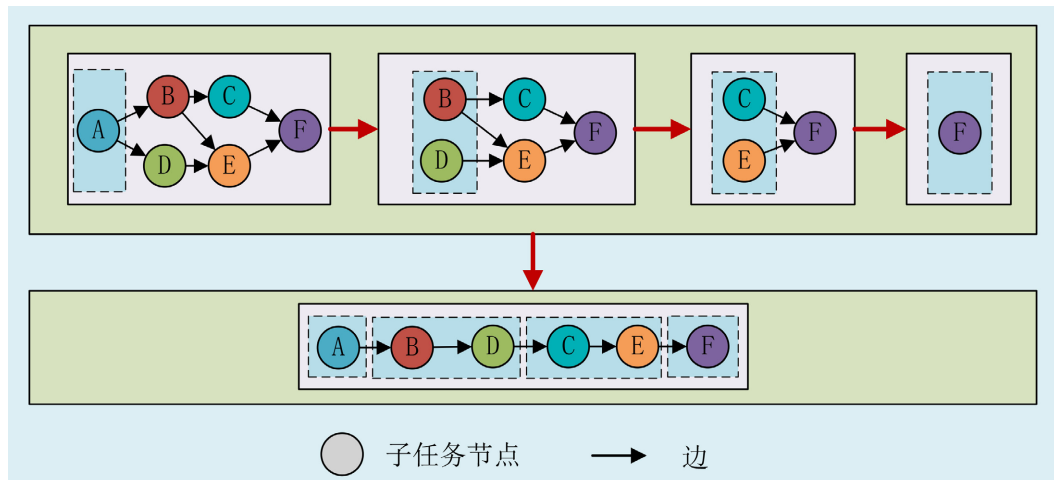


Figure 3. Example diagram of parallelizable computational decomposition for dependent tasks

图 3. 具有复杂依赖关系的任务可并行化计算分解示例图

卫星网络中, 由于单个卫星在同一时刻只能处理一个任务, 为提高任务执行效率, 通过任务分解与

并行化, 将可并行的任务卸载到多个卫星节点进行处理。如图 3 所示, 一个具有复杂依赖关系的任务通过构建 DAG 被分解为多个子任务(节点表示子任务, 边表示依赖关系)。基于任务依赖关系, 将任务分为 4 组, 第一组为子任务 A, 第二组为子任务 B 和 D; 第三组为子任务 C 和 E; 第四组为子任务 F。各组内任务被并行分配到不同的卫星节点中执行, 能够充分利用卫星的计算资源减少任务处理时延。

将具有复杂依赖关系的任务 V_q 分解为多个可并行计算组, 表示如下:

$$\begin{aligned} \mathcal{G}_q &= \{G_{q,1}, G_{q,2}, \dots, G_{q,z}, \dots, G_{q,Z}\}, \\ G_{q,z} &\in V_q \end{aligned} \quad (12)$$

其中 G_z 表示第 z 个可并行计算组, 并且 $Z \leq U$ 。

$$td_{q,z} = \max_{v_{q,n} \in G_{q,z}} (td_{q,n}) \quad (13)$$

$$gd_{q,z} = \max_{v_{q,n} \in G_{q,z}} (cd_{q,n}^{h,k} + qd_{q,n}^{h,k}) \quad (14)$$

其中 $gd_{q,z}$ 为具有复杂依赖关系的任务 V_q 中第 z 个并行计算组的计算时延, $td_{q,z}$ 为跨并行计算组之间的传输时延。

3.2. 任务卸载优化模型

在 DQN Network 模块中, 将 $Pred(G_{q,z})$ 表示为并行计算组 $G_{q,z}$ 的直接前驱集合, 并行计算组 $G_{q,z}$ 可能由多个前驱组成, 那么 $G_{q,z}$ 接收到的所有前驱任务后的前驱完成时延为:

$$pd_{q,z} = \max_{G_{q,z} \in Pred(G_{q,z})} (td_{q,z} + gd_{q,z}) \quad (15)$$

并行计算组 $G_{q,z}$ 的开始时间为:

$$ts_{q,z} = \max(td_{q,z}, pd_{q,z}) \quad (16)$$

任务 $v_{q,n}$ 的开始时间为:

$$ts_{q,n} = ts_{q,z} + qd_{q,n}^{h,k} \quad (17)$$

此外, 定义 $Succ(v_{q,n})$ 为子任务 $v_{q,n}$ 的直接后继集合。通过以上分析, 一个具有复杂依赖关系任务的总完成时间可表示为:

$$T(V_q) = td_q^{(r),(h,k)} + pd_{q,z} + gd_{q,z} \quad (18)$$

基于上述系统模型制定的优化问题, 旨在针对具有复杂依赖关系的任务优化其任务执行时延。针对该计算架构联合优化其任务卸载策略, 优化问题表述为:

$$(P1): \min_{x(q), f(q)} \sum_{q=1}^Q T(V_q) \quad (19)$$

$$0 \leq f_{q,n}^{h,k} \leq f_{h,k}^{\max}, \quad \forall q, n, h, k \quad (19a)$$

$$td_{q,z} \leq \max_{v_{q,n} \in G_{q,z}} (\rho_{q,n}), \quad \forall q, z, w, n \quad (19b)$$

$$\omega_{h,k}^{q,n} \in \{0, 1\} \quad (19c)$$

$$\sum_{h=1}^H \sum_{k=1}^K \omega_{h,k}^{q,n} = 1 \quad (19d)$$

其中 $\mathbf{x}(q) = [x_1(q), x_2(q) \cdots x_U(q)]$, 为任务 V_q 中子任务产生的卸载决策, $\mathbf{f}(q) = [f_1(q), f_2(q) \cdots f_U(q)]$, 为任务 V_q 中子任务分配的计算能力。式(19a)确保计算资源不会超过最大计算能力, 式(19b)确保并行计算任务组的最大容忍时延不超过组内任务最大容忍时延, 式(19c)确保任务被卸载至低轨卫星上, 式(19d)确保每个子任务只能被卸载至一颗配备 MEC 服务器的低轨卫星上。

P1 是一个非线性优化问题, 传统方法难以求解。为此, 本文将卫星网络中的任务卸载建模为最优马尔可夫决策过程, 以最小化系统平均任务处理时延为目标。针对任务的复杂依赖关系, 提出基于 DQN 算法的任务卸载策略, 优化多卫星网络中的任务处理时延。在资源受限条件下, 该方法有效提升计算资源利用效率, 减少任务处理时延, 并适应动态环境变化。

3.3. 基于强化学习的任务卸载策略求解

在本文求解模型中, 构建了一个基于马尔可夫决策过程(Markov Decision Process, MDP)的具有复杂依赖关系的任务卸载策略, MDP 定义如下:

1) 状态空间: 针对具有复杂依赖关系的任务卸载请求, 定义状态空间为 $S_i = \{G_q, Area_i\}$, 其中 $G_q = \{V_q, E_q\}$, V_q 表示任务集, $v_{q,n} \in V_q$ 表示子任务信息, E_q 以邻接矩阵的方式表示各子任务之间的依赖关系, $Area_i = \left\{ \text{Avg}(Q_h^F(i+1), \text{Avg}(be_i^h)) \mid h \in \{1, 2 \cdots H\} \right\}$, 其中 $\text{Avg}()$ 表示对某一区域所有卫星状态信息取平均。此外, 针对子任务域内的卸载请求, 本研究还定义状态空间 $SubS_{i,h} = \{v_{q,n}, SubArea_i\}$, 其中 $SubArea_{i,h} = \{Q_{h,k}^F(i), be_i^{h,k} \mid k \in \{1, 2 \cdots K\}\}$ 。

2) 动作空间: 针对具有复杂依赖关系的任务卸载请求, 定义动作空间为 $A_i = \{\omega_h^{q,n} \mid h \in \{1, 2 \cdots H\}\}$, 其中 $\omega_h^{q,n}$ 为子任务卸载目标区域 SN_h 。针对子任务卸载请求, 定义动作空间为 $SubA_{i,h} = \{\mathbf{x}(q), \mathbf{f}(q)\}$, 其中 $\mathbf{x}(q)$ 为域内卸载决策 $\omega_{h,k}^{q,n}$, $\mathbf{f}(q)$ 为 V_q 分配的计算资源。

3) 奖励函数: 奖励函数用于量化每个动作对目标优化的贡献。在本问题中, 奖励函数为延迟惩罚。定义奖励函数为:

$$R_i = -Del_q \quad (20)$$

其中 Del_q 表示处理任务 V_q 的计算时延。

本文算法目标是通过最小化时间差分误差来更新 Q 网络的参数, 使其能够更准确地预测不同状态 - 动作对的期望回报。通过引入目标网络, 用于计算 Q 值的目标。目标值的计算公式为:

$$\gamma = R + \gamma \max_{a'} Q_{target}(S', A'; \theta^-)$$

其中 R 是即时奖励, γ 是折扣因子, $Q_{target}(S', A'; \theta^-)$ 是目标网络预测下一状态的最大 Q 值。DQN 通过最小化均方误差来优化主网络参数:

$$L(\theta) = E_{(S,A,R,S') \sim D} [\gamma - Q(S', A'; \theta)^2]$$

其中 γ 是上述目标值, $Q(S', A'; \theta)$ 是主网络输出的 Q 值。通过反向传播算法计算梯度, 并使用优化器(如 Adam 或 RMSprop)更新主网络的参数:

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} L(\theta)$$

其中 $\alpha \in (0, 1)$ 为更新系数。此外, 使用 ϵ -贪婪策略(ϵ -greedy policy)来平衡探索与利用。

3.4. 算法复杂度分析

本文所提基于 DQN 算法的卸载策略时间复杂度涵盖多层面: 环境交互中, 单任务执行与奖励生成复杂度均为 $O(1)$, 若单次训练包含 N 个任务交互, 总复杂度为 $O(N)$; DQN Network 模块中, GCN 特征提

取的图构建复杂度为 $O(V+E)$ ，其中 V 为节点数， E 为边数， L 层 GCN 总复杂度为 $O(L \cdot E \cdot F_{in} \cdot F_{out})$ ， F_{in}/F_{out} 为输入和输出特征维度；第一子网络(卸载区域选择)全连接计算复杂度为 $O(D \cdot H \cdot M)$ ，其中特征融合后维度为 D ，全连接层隐藏层神经元数为 H ，候选区域数为 M ；第二子网络(卫星决策与资源分配)为 $O(S \cdot H' \cdot A)$ ，其中输入特征维度为 S ，隐藏层神经元数为 H' ，动作空间维度为 A ；Training 模块中，经验池单次抽样复杂度为 $O(B)$ ，若训练迭代 T 次、每次抽样 B 样本，总复杂度为 $O(T \cdot B)$ 。算法最坏时间复杂度为 $O(N + L \cdot E \cdot F_{in} \cdot F_{out} + D \cdot H \cdot M + S \cdot H' \cdot A + T \cdot B)$ ，其通过 GCN 分解任务依赖关系、线性层网络决策有效控制复杂度，在保证卸载决策精度的同时，满足任务卸载的实时性需求。

4. 实验模拟与分析

4.1. 参数设置

在本文中，使用仿真实验来评估所提出算法的性能，实验数据来自 STK 软件。在仿真中，LEO 卫星星座由 36 颗卫星组成，轨道倾角为 55° ，高度为 550 公里，每个区域卫星规模为 2×2 。任务 V_q 的节点数为 10，依赖边数为 10，星地链路带宽为 80 Mhz， $sd_{q,n}$ 任务大小随机分布在 $[sd^{\min}, sd^{\max}]$ ， $c_{q,n}$ 独立且同分布于 $[c^{\min}, c^{\max}]$ ，详细参数如表 1 所示。

Table 1. Experimental parameters

表 1. 实验参数

名称	值
任务 V_q 的节点数量	10
依赖边数量	10
$C_i^{(r)(h,k)}$ 速率	[200, 300] Mbit/s
$Pt_i^{h,j}$ 发射功率	0.1 W
通信频率 F	0.915 Ghz
LEO 卫星天线增益	43 dBi
$sd^{\min}, sd^{\max}$	[5, 20] Mbit/s
$c^{\min}, c^{\max}$	[20, 40] cycles/bit
$\mu_{mob}^{h,k}$	10^{-26}
子任务容忍时延	[0.8, 2] seconds

4.2. 仿真结果

本文分两部分来说明所提出策略的仿真效果。首先，评估所提方法的收敛性。其次，将所提出的策略与卸载策略性能进行比较，依次比较不同条件下的平均任务处理时延，展示所提策略的优越性。

1) 收敛性能

图 4 展示两个子网络的训练损失随回合数的变化，其中(a)为第一个子网络，(b)为第二个子网络。整体来看，两个子网络的损失初始较高，随后迅速下降并趋于稳定，表明模型实现有效策略学习。第一个子网络初始损失较大，前几千回合波动明显，但下降较快，约 2000 回合后趋于稳定，说明其能快速适应任务与区域特征的依赖关系，优化子任务卸载区域决策。第二个子网络同样快速收敛，波动更小，约 2000 回合后稳定，表明其能有效完成资源分配及最终卸载目标选择。

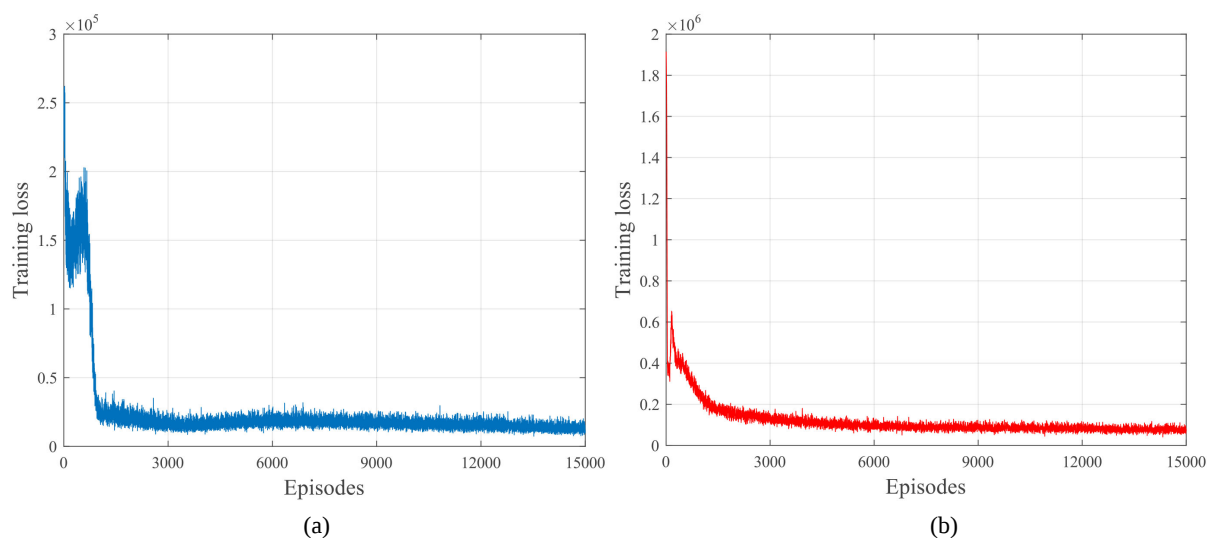


Figure 4. Algorithm training loss diagram: (a) Training loss of network 1; (b) Training loss of network 2
图 4. 算法训练损失图: (a) 网络 1 训练损失图; (b) 网络 2 训练损失图

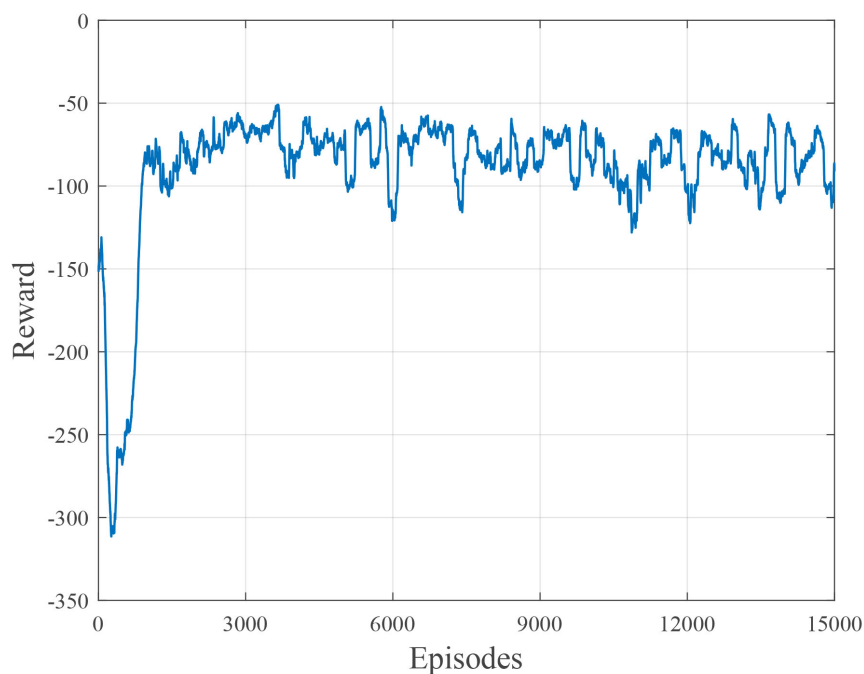


Figure 5. Reward variation with training
图 5. 随训练变化的奖励趋势图

图 5 展示奖励值随训练回合数的变化趋势。可以看出初期奖励值较低且波动较大，因为模型尚未建立有效决策机制。约 2000 回合后，奖励值显著提升，表明子网络逐步学习到有效策略。随后，尽管奖励值仍有波动，但总体保持较高水平，反映模型已基本收敛，能稳定生成卸载决策。

图 6 展示任务完成率的变化趋势。任务完成率基于任务的实际执行情况统计，当子任务失败时，整个任务记录为失败并停止计算。可以看出初期任务完成率较低，表明系统资源分配未优化，策略尚未收敛。随着训练的推进，策略逐渐优化子任务卸载策略，任务完成率快速提升并保持高水平，表明系统能高效完成任务，并具有较强的适应能力和鲁棒性。

图 7 展示任务执行时延的变化趋势。可以看出初期任务执行时延较低，因子任务计算失败未进入完整执行流程。随着任务完成率提升，更多任务被成功执行，时延上升。当任务完成率稳定后，时延在 1.95 秒至 2.1 秒之间波动并稳定。这表明尽管网络环境和计算负载动态变化，该策略能够适应并维持较低处理延迟。

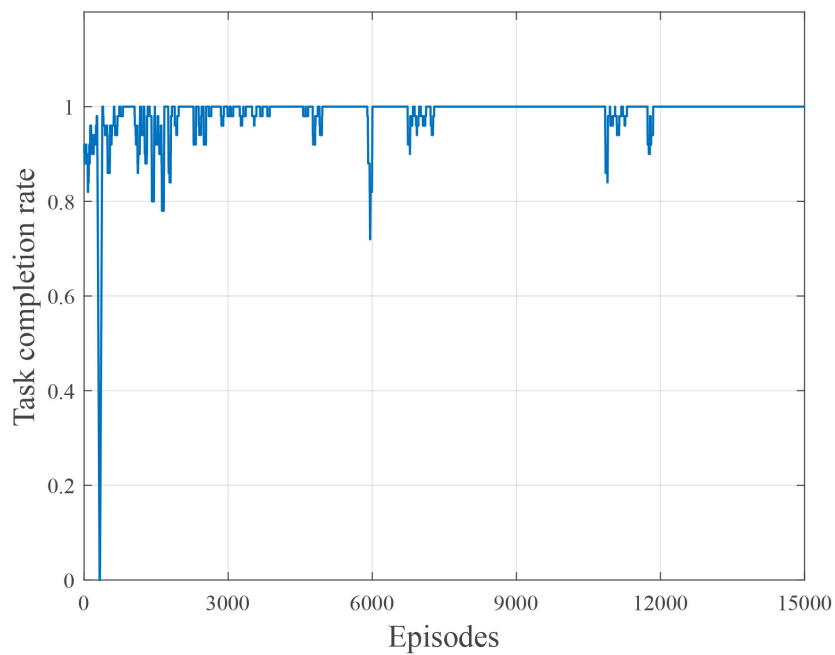


Figure 6. Task completion rate variation

图 6. 任务完成率变化趋势

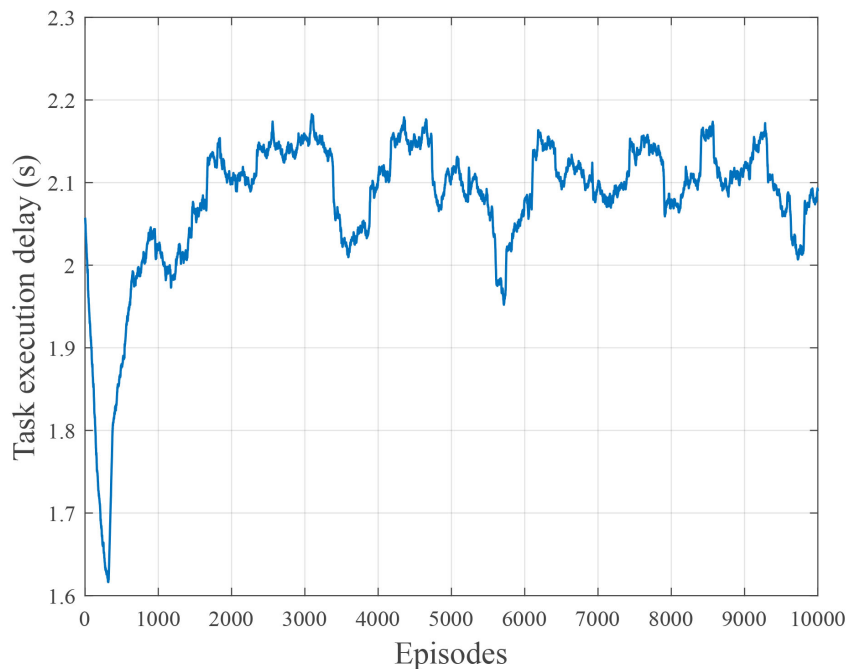


Figure 7. Task processing latency variation

图 7. 任务处理时延变化趋势图

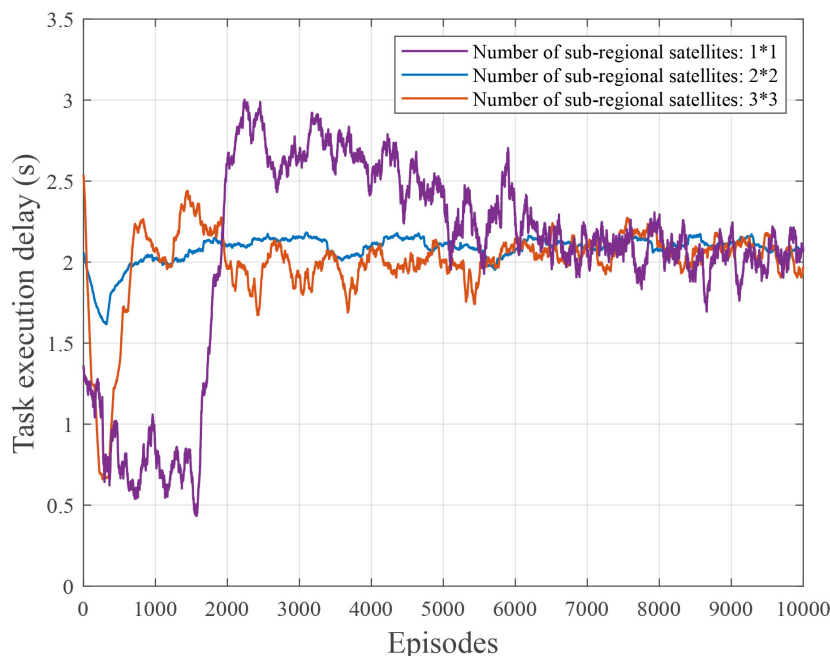


Figure 8. The influence of satellite area scale variation on task processing delay
图 8. 卫星区域规模变动对任务处理时延的影响

图 8 展示了卫星区域规模对任务处理时延的影响。初始时各曲线波动大，而后处理时延逐渐稳定，表明策略可适配不同网络规模。具体来说，小规模区域计算资源有限，初期任务卸载困难，导致时延波动剧烈，经后续学习训练得以稳定。大规模区域资源丰富，却因拓扑复杂、卸载决策多导致初期卸载任务策略低效，而后凭借不断学习实现稳定收敛。由此可见，区域规模的设置需要根据实际应用综合考虑以达最优任务处理效果。

2) 策略性能对比

为了验证所提策略的优越性和可靠性，本文将其与以下任务卸载策略进行比较：

- **DQN-SCCO 策略：**该策略[13]利用深度 Q 网络来实现最优卸载策略，为低轨卫星网络中的任务卸载提供有效解法。
- **DDRATO 策略：**该策略[14]通过融合策略网络(Actor)和价值网络(Critic)实现对任务卸载策略的优化。
- **Rand 随机策略：**在此策略中，卸载决策以随机的概率制定动作输出。

图 9 图展示了在依赖边数量固定为 10 的情况下，四种策略在不同节点数量(即子任务数量)下的任务执行时延对比。对于具有复杂依赖关系的任务，节点数量的增加意味着任务被分解成了更多的子任务需要处理。然而，由于依赖边数量的固定，任务之间的依赖关系并未增加，这允许更多的子任务能够被并行计算。因此，尽管节点数量增加，但是总任务执行时延的增长并不显著。这表明在这种特定条件下，任务的并行性被有效利用，从而在一定程度上抵消了节点数量增加带来的潜在时延增加。

图 10 对比了不同依赖边数量下各策略的任务处理时延。结果显示，随着依赖边数量增加，所有策略的时延均上升，但性能差异明显。本文所提策略表现最佳，时延最低且增长幅度最小，表明融合 GCN 的 DQN 算法能有效捕获任务依赖，提升任务卸载效率。DQN-SCCO 并未融合 GCN，所表现出时延显著增加。DDRATO 性能略逊于 DQN-SCCO，可能因其在复杂环境下对任务特性和资源分配的适应性不足。Random 策略时延最高，且随依赖边数量增加增长最快，难以适应任务依赖的复杂性。

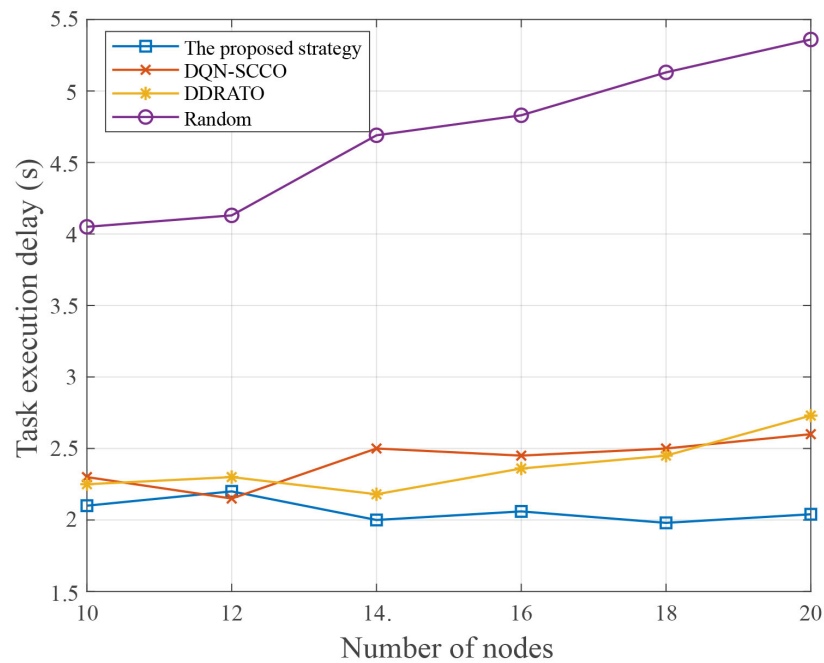


Figure 9. Comparison of task processing delays of different strategies under different numbers of sub-tasks when the number of dependency edges is fixed at 10

图 9. 依赖边数量固定为 10 的情况下, 不同策略在不同子任务数量下的任务处理时延对比

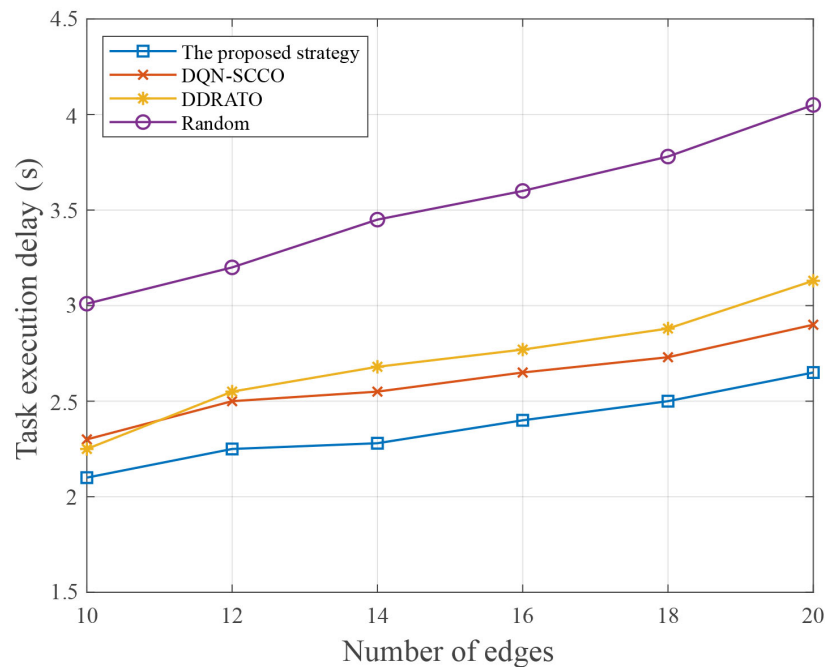


Figure 10. Comparison of task processing delays of different strategies under different numbers of sub-tasks with different numbers of dependency edges

图 10. 不同依赖边数量下, 不同策略在不同子任务数量下的任务处理时延对比

5. 总结

本文提出一种基于低轨卫星网络的具有复杂依赖关系的任务卸载策略, 设计基于 DQN 的深度强化

学习算法, 帮助控制器节点在动态条件下实时优化具有复杂依赖关系的任务卸载决策。仿真实验分析表明所提策略能显著降低任务处理时延, 展现优越的适应性与鲁棒性。

基金项目

国家自然科学基金资助项目(61602305, 61802257); 上海市自然科学基金资助项目(18ZR1426000, 19ZR1477600)。

参考文献

- [1] Sahni, Y., Cao, J., Yang, L. and Ji, Y. (2021) Multihop Offloading of Multiple DAG Tasks in Collaborative Edge Computing. *IEEE Internet of Things Journal*, **8**, 4893-4905. <https://doi.org/10.1109/jiot.2020.3030926>
- [2] Zhao, G., Xu, H., Zhao, Y., Qiao, C. and Huang, L. (2021) Offloading Tasks with Dependency and Service Caching in Mobile Edge Computing. *IEEE Transactions on Parallel and Distributed Systems*, **32**, 2777-2792. <https://doi.org/10.1109/tpds.2021.3076687>
- [3] Li, Y. and Li, L. (2024) Computation Offloading and Resource Allocation in Mec-Enabled Vehicular Networks: Partial Offloading versus Binary Offloading. 2024 7th World Conference on Computing and Communication Technologies (WCCCT), Chengdu, 12-14 April 2024, 260-266. <https://doi.org/10.1109/wccct60665.2024.10541511>
- [4] Wang, S., Xin, N., Luo, Z. and Lin, T. (2022) An Efficient Computation Offloading Strategy Based on Cloud-Edge Collaboration in Vehicular Edge Computing. 2022 International Conference on Computing, Communication, Perception and Quantum Technology (CCPQT), Xiamen, 5-7 August 2022, 193-197. <https://doi.org/10.1109/ccpqt56151.2022.00041>
- [5] Yang, X. and Hong, B. (2022) Cost-Efficient Task Offloading for Satellite Edge Computing Systems. 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, 19-22 July 2022, 1-5. <https://doi.org/10.1109/isncc55209.2022.9851711>
- [6] Zhai, Z., Yu, S., Zhang, F. and Chen, X. (2022) An On-Orbit Computation Offloading Framework for Satellite Edge Computing. 2022 IEEE/CIC International Conference on Communications in China (ICCC), Sanshui, 11-13 August 2022, 1062-1067. <https://doi.org/10.1109/iccc55456.2022.9880821>
- [7] Zhang, Z., Zhang, W. and Tseng, F. (2019) Satellite Mobile Edge Computing: Improving QOS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques. *IEEE Network*, **33**, 70-76. <https://doi.org/10.1109/mnet.2018.1800172>
- [8] Zhou, H., Li, L.J. and Dong, Z.S. (2023) Multi-Tasks Resource Joint Optimization Based on Asynchronous Reward Deep Deterministic Policy Gradient in Edge Computing. *Application Research of Computers*, **40**, 1491-1496.
- [9] Zhu, D.L., et al. (2021) Deep Reinforcement Learning-Based Task Offloading in Satellite-Terrestrial Edge Computing Networks. 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, 29 March-1 April 2021, 1-7.
- [10] Wang, J., Chen, P., Yin, Q., Yu, M., Zhang, M. and Zhang, X. (2023) A DRL-Based Algorithm for Dependent Task Offloading in Multi-Access Edge Computing. 2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, 8-10 December 2023, 38-43. <https://doi.org/10.1109/iaecst60924.2023.10503133>
- [11] Deng, Y. and Peng, A. (2024) A DRL-Based Intelligent Task Offloading and Caching Strategy for IIoT in MEC. 2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE), Guangzhou, 10-12 May 2024, 1347-1351. <https://doi.org/10.1109/cisce62493.2024.10653040>
- [12] Qin, Z., Yao, H., Mai, T., Wu, D., Zhang, N. and Guo, S. (2022) Multi-Agent Reinforcement Learning Aided Computation Offloading in Aerial Computing for the Internet-of-Things. *IEEE Transactions on Services Computing*, **16**, 1976-1986. <https://doi.org/10.1109/tsc.2022.3190562>
- [13] Cao, H., Peng, Y., Wang, H., Jia, H., Kong, L. and Cao, S. (2024) Multi-Satellite Cooperative Computing Task Offloading Strategy Based on Deep Reinforcement Learning. 2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI), Xi'an, 24-26 May 2024, 464-471. <https://doi.org/10.1109/ccai61966.2024.10603385>
- [14] Wang, C. and Luo, Z. (2024) A DRL-Based Dynamic Resource Allocation and Task Offloading Algorithm for LEO Satellite Network. 2024 International Conference on Satellite Internet (SAT-NET), Xi'an, 25-27 October 2024, 59-65. <https://doi.org/10.1109/sat-net62854.2024.00018>