

不可靠信道传输环境下的移动边缘计算卸载优化算法

邓渊崎, 周启钊

(成都信息工程大学 计算机学院, 四川 成都 610225)

摘要: 随着移动边缘计算应用环境的日趋复杂化, 计算卸载面临着诸多严峻挑战, 具体表现为数据传输信道可靠性难以保障、应用程序呈现多样化特征以及计算资源具有显著的异构性。这些挑战直接导致传统的卸载决策方法效率低下, 难以满足实际应用需求。为此, 提出一种基于深度强化学习的增强型部分卸载优化算法, 其能够根据任务需求和计算资源等为用户产生的时序任务制定最优卸载方案。为应对数据传输的不可靠性, 该方案融入任务信息预测模型, 当上传至智能体的任务信息发生丢失或错误时, 由预测模型对任务信息进行补充, 辅助算法作出准确的卸载决策。同时, 针对任务的多样性问题设计一个具有任务特性的动态卸载成本函数, 作为计算卸载问题的优化目标, 并利用改进的深度强化学习算法对该目标进行优化。仿真实验结果表明, 与3种现有方案相比, 所提方案的卸载成本分别降低了36.3%、21.8%和12.9%, 且具有更高的收敛效率, 能够适应不同偏好设置的应用和动态的网络环境。

关键词: 移动边缘计算; 计算卸载; 深度强化学习; 信息补充; 卸载优化

DOI: 10.11907/rjtk.251117

扫描二维码阅读全文:



中图分类号: TP301.6

文献标识码: A

文章编号: 1672-7800(2026)005-0143-08

Offloading Optimization Algorithm for Mobile Edge Computing in Unreliable Channel Transmission Environment

DENG Yuanqi, ZHOU Qizhao

(School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: With the increasingly complex application environment of mobile edge computing, computing offload is facing many serious challenges, which are manifested in the difficulty in ensuring the reliability of data transmission channels, the diversity of applications and the significant heterogeneity of computing resources. These challenges directly lead to the inefficiency of traditional unloading decision-making methods, which is difficult to meet the actual application requirements. Therefore, an enhanced partial unloading optimization algorithm based on deep reinforcement learning is proposed, which can formulate the optimal unloading scheme for user generated sequential tasks according to task requirements and computing resources. In order to deal with the unreliability of data transmission, the scheme integrates the task information prediction model. When the task information uploaded to the agent is lost or wrong, the prediction model will complement the task information and assist the algorithm to make accurate unloading decisions. At the same time, a dynamic unloading cost function with task characteristics is designed for the problem of task diversity as the optimization goal of computing the unloading problem, and an improved deep reinforcement learning algorithm is used to optimize this goal. The simulation results show that compared with the three existing schemes, the unloading cost of the proposed scheme is reduced by 36.3%, 21.8% and 12.9% respectively, and has higher convergence efficiency, which can adapt to applications with different preferences and dynamic network environment.

Key Words: mobile edge computing; computing offloading; deep reinforcement learning; information completion; offloading optimization

收稿日期: 2025-04-02

基金项目: 重载快捷大功率电力机车全国重点实验室开放项目(QZKFKT2025-06); 成都信息工程大学引进人才科研启动项目(KYTZ202269)

作者简介: 邓渊崎(1998-), 男, 成都信息工程大学计算机学院硕士研究生, 研究方向为边缘计算与深度强化学习; 周启钊(1991-), 男, 博士, 成都信息工程大学计算机学院副教授, 研究方向为软件定义网络、边缘计算、网络安全、机器学习等。

0 引言

随着移动互联网技术的发展和5G网络的普及,移动端流量呈现出爆炸式增长趋势^[1]。移动边缘计算(Mobile Edge Computing, MEC)作为一种新兴计算范式,允许移动设备(Mobile Device, MD)将计算任务卸载到附近的边缘服务器(Edge Server, ES)进行处理,从而有效减少任务处理延迟^[2]。为用户设计一种高效且可靠的卸载策略是MEC研究领域的核心问题之一^[3-4]。

人工智能方法因在动态环境中具有智能决策能力,已被大量研究者应用于解决计算卸载问题。该类方法基于智能体对环境状态的观测作出卸载决策。现有研究大多基于理想的网络环境,认为在观测过程中,MD均通过无线网络正确地上传任务信息。然而,随着MD数量呈现出爆发式增长态势,同时无线通信信道容量逐渐逼近香农极限,导致传输信道可靠性降低,显著影响MD与ES之间的数据传输稳定性^[5]。不稳定的数据传输将导致MD上传的任务信息发生丢失或错误,而错误的状态观测将导致智能体无法作出准确的卸载决策。因此,在传输网络不可靠的MEC环境中,确保卸载方案的有效性是一个重大挑战。

近年来,已有大量研究致力于解决具有场景化特征的计算卸载问题。为了在保证服务质量的同时提高能源效率,这些研究专门分析了场景需求和特点,制定了个性化计算卸载方案。例如,文献[6]提出一种基于块坐标下降与凸优化技术的优化算法,提高了边缘系统在人群拥挤场景下的卸载效率。文献[7]通过改进传统人工鱼群算法提升了网络性能,加快了算法的收敛速度,并有效降低了海上探测传感器的网络延迟。以上研究主要在静态或准静态网络环境中执行卸载策略,且算法通常需要进行大量迭代才能找到最优解,在实际应用中存在局限性,难以满足快速响应的卸载需求^[4]。

深度强化学习(Deep Reinforcement Learning, DRL)因能在动态环境中根据不同的系统状态选择合适的动作,已成为解决场景化计算卸载问题的重要路径。例如,文献[8]在具有多个服务器的时变MEC系统中提出一种端到端的DRL方法,以最大化任务完成数量并最小化能耗。为有效应对6G紧急场景下计算密集型任务的挑战,文献[9]结合决斗网络和优先级经验重放(Prioritized Experience Replay, PER),提出一种支持人工智能的空中计算网络方法,旨在最大限度地降低系统总延迟。为提高灾后应急响应能力,文献[10]提出采用无人机作为MEC服务器的方案,并利用双深度Q网络(Double Deep Q-Network, DDQN)优化卸载过程。上述研究均理想地认为无线网络环境是可靠的。然而在实际应用中,无线通信具有不可靠性,智能体观测到的任务信息可能不完整或发生数据错误,这种不确定性将导致计算卸载性能显著降低,并严重影响模型在训练过程中的收敛速度,增加优化过程的复杂性^[11]。因此,设计一个能有效应对无线通信不确定性以及动态网络

环境的计算卸载策略,对于提升MEC系统的性能至关重要。

目前,针对不可靠传输网络的MEC卸载场景研究相对较少,但其对用户体验的影响不容忽视。此外,相关文献中对卸载成本的定义缺乏特异性,通常解释为对时延和能耗采用固定的权重分配,忽略了不同应用程序在计算卸载过程中可能存在的个性化卸载需求。本文针对不可靠传输网络场景设计了具有个性化偏好应用程序的边缘计算卸载策略,并提出一种部分卸载优化算法POOA-UN(Partial Offloading Optimization Algorithm for Unreliable transmission Network,),主要贡献点为:①增强了算法的卸载性能。针对MEC在无线链路不稳定场景下的实际应用挑战,提出一种基于预测模型的卸载方法。该方法利用历史任务数据对缺失信息进行补全,提高了卸载策略的可靠性。结合预测模型,算法在训练过程中的学习效率和稳定性得到明显提升;②定义了个性化用户成本。根据应用程序对时延和能耗的特定需求,将其分为时延敏感型、能耗敏感型和均衡型。根据应用程序的特定需求设计了一种个性化卸载成本函数,并以此作为优化目标;③验证了所提方法的可行性与有效性。实验结果表明,与其他基准方法相比,POOA-UN不仅能够快速、准确地作出卸载决策,且具有更好的收敛性和更低的用户成本。此外,该方法在处理具有不同个性化偏好设置的应用程序时表现出更强的鲁棒性。

1 系统模型

如图1所示,本文提出的MEC系统由一组ES和一个MD构成。其中, m 个ESs记为集合 $\mathcal{M} = \{1, 2, \dots, M\}$ 。MD在MEC的覆盖范围内自由移动,并随机产生计算密集型任务。该系统采用时长为 T 的离散时间模型,时间集合表示为 $\mathcal{T} = \{1, 2, \dots, T\}$ ^[12]。网络环境会随着时间产生变化,但是为了保证在每个时隙 t 开始时作出的卸载决策是有效的,本文假设MD的位置和系统状态在时隙内不变。

1.1 任务模型

假设在每个时隙 t 开始时只有一个随机任务到达MD,任务信息可以表示为:

$$\mathcal{U}(t) = \{D(t), C(t), \xi(t)\}, \forall t \in \mathcal{T} \quad (1)$$

式中: $D(t)$ 表示任务数据大小; $C(t)$ 表示完成任务所需的计算资源; $\xi(t)$ 表示任务类型。本文将任务分为时延敏感型、能耗敏感型和均衡型3种代表性类型,不同类型的任务采用不同的成本计算方式,以便更好地满足用户的多样化需求。例如,智能驾驶应用的响应速度直接关系到驾驶安全,在这种场景下,降低响应延迟是计算卸载的首要任务。而对于高能耗的应用,用户则更关注设备的续航能力,因此降低能耗成为这些任务的首要目标。基于任务的独立性假设,本文理想地认为这些任务可以被划分为独立的两部分,并分别进行处理^[13-14]。因此,卸载决策可由

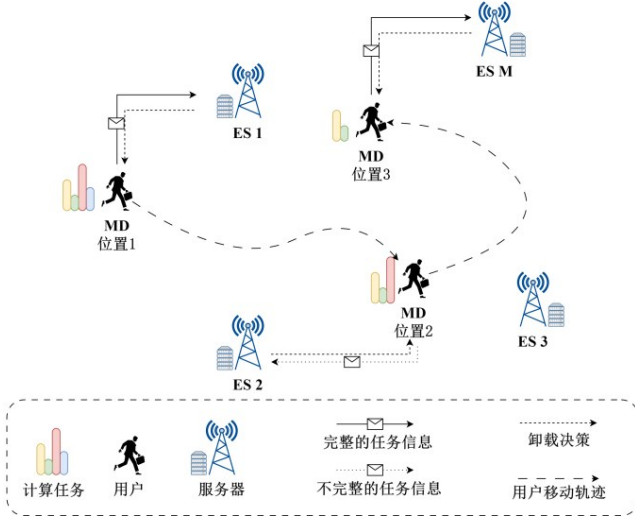


Fig. 1 MEC system model with unreliable transmission network

图1 传输网络不可靠的MEC系统模型

卸载比例 $\lambda(t)$ 和卸载节点 $\varphi_m(t)$ 两部分组成。

1.2 计算模型

由于MD和ES均可以提供计算服务,可以根据任务的执行位置将计算模式划分为本地计算和边缘计算两种。

1.2.1 本地计算模式

本地计算模式的延迟和能耗主要由设备的CPU导致^[15]。该模式的时延和能耗分别定义为:

$$T_m^{local}(t) = \frac{C(t)(1 - \lambda(t))}{f_i(t)} \quad (2)$$

$$E_m^{local}(t) = kC(t)(1 - \lambda(t))f_i(t)^2 \quad (3)$$

式中: $f_i(t)$ 表示MD的CPU频率; k 表示MD的有效电容系数。

1.2.2 边缘计算模式

当MD的计算资源不足以处理密集型任务时,可以将部分任务卸载到ES进行处理,从而提高计算效率。本文遵循以下两个假设:①忽略MD下载计算结果的延迟和能耗;②忽略任务处理的排队延迟^[4]。将MD与第 m 个ES之间的数据传输速率定义为:

$$r_m(t) = B_m \log_2 \left(1 + \frac{h(t)p}{p l_m(t) B_m N_0} \right) \quad (4)$$

式中: B_m 表示系统带宽; $h(t)$ 表示信道增益; p 表示MD的上传功率; $p l_m(t)$ 表示路径损耗; N_0 表示噪声功率谱密度; $h(t)/p l_m(t) = g_0(Dis_0/Dis_m(t))^\sigma$, g_0 表示路径损耗常量, Dis_0 表示参考距离, $Dis_m(t)$ 表示MD与第 m 个ES之间的实际距离, σ 表示路径损耗指数。因此,MD产生的任务卸载到第 m 个ES的传输时间定义为:

$$T_m^{trans}(t) = \frac{\lambda(t)D(t)}{r_m(t)} \quad (5)$$

计算任务在第 m 个ES上的处理延迟定义为:

$$T_m^{comp}(t) = \frac{\lambda(t)C(t)}{f_m(t)} \quad (6)$$

式中: $f_m(t)$ 表示MD可以从第 m 个ES获得的计算资源。因此,边缘计算模式的时延和能耗分别为:

$$T_m^{edge}(t) = T_m^{trans}(t) + T_m^{comp}(t) \quad (7)$$

$$E_m^{edge}(t) = p * T_m^{trans}(t) \quad (8)$$

1.3 问题建模

鉴于应用程序的多样性,固定权重的成本计算方式难以满足各类任务的特定需求。为此,本文提出一种个性化成本函数。首先,设置一组时延和能耗的阈值,并根据阈值对任务进行分类;其次,根据任务类型分配时延和能耗在总成本中的权重比例。该种方式能更精确地适应不同任务的需求,实现更加灵活和高效的成本计算。卸载成本定义为:

$$C(t) = \alpha_\xi(t) \max(T_m^{local}(t), T_m^{edge}(t)) + \beta_\xi(t) (E_m^{local}(t) + E_m^{edge}(t)) \quad (9)$$

$$\begin{cases} \alpha_\xi(t), \beta_\xi(t) \in (0, 1), \forall t \in T \\ \alpha_\xi(t) + \beta_\xi(t) = 1, \forall t \in T \end{cases} \quad (10)$$

式中: $\alpha_\xi(t)$ 和 $\beta_\xi(t)$ 分别表示 ξ 型任务的时延和能耗权重,这两个权重因子受限于式(10)。该问题的目标为最小化用户成本,目标函数可表示为:

$$\begin{cases} \min_{\lambda(t), \varphi_m(t)} \sum_{t=1}^T C(t) \\ s.t. C1: \lambda(t) \in (0, 1), \forall t \in T \\ C2: \varphi_m(t) \in \{0, 1\}, \sum_{m=1}^M \varphi_m(t) = 1, \forall t \in T \end{cases} \quad (11)$$

式中: C1 表示对卸载比例的约束,即每个计算任务可以完全本地处理、部分卸载或完全卸载; C2 表示每个任务每次只能卸载到一个ES进行处理。

2 POOA-UN算法

POOA-UN算法是一种针对传输网络不稳定场景的增强型深度Q网络(Deep Q-Network, DQN)计算卸载方案,具有任务信息预测和PER两大显著特征。以下首先将问题建模为马尔可夫决策过程(Markov Decision Process, MDP),其次从增强机制的角度详细介绍算法框架。

2.1 MDP构建

实时状态包括MD产生的任务信息、MD与ESs之间的位置信息、MD的可用计算资源、ESs的可用计算资源。因此,状态空间可表示为:

$$S(t) = \{D(t), C(t), \xi(t), d(t), f(t), f_i(t)\} \quad (12)$$

式中: $d(t)$ 表示 t 时刻MD与ESs之间的距离,记作集合 $d(t) = \{d_1(t), \dots, d_M(t)\}$; $f(t)$ 表示 t 时刻ESs的可用计算资源,记作集合 $f(t) = \{f_1(t), \dots, f_M(t)\}$ 。

卸载决策包括卸载比例和卸载节点的选择。 t 时刻的动作空间可表示为:

$$A(t) = \{\lambda(t), \varphi_m(t)\} \quad (13)$$

强化学习的目标为最大化长期总回报,而本文的核心

验数据的重要性和优先级,从而进一步优化经验采样过程。在模型的探索过程中,根据时间差分(Temporal Difference, TD)误差计算被采集经验的优先级,然后根据优先级对其进行重放学习。TD误差的差值 δ 定义为:

$$\delta = y - Q(s, a) \quad (18)$$

式中: y 为目标Q值; $Q(s, a)$ 为当前网络的预测Q值。差值越大,表示其精度的提升空间越大,优先级越高。

经验 j 的优先级定义为:

$$p_j = |r_j(y_j - Q(s, a))| \quad (19)$$

式中: r_j 为动作的即时奖励。然而,优先级采样会带来多样性损失和偏差问题,导致经验池中的一些经验很少被采样,甚至被忽略。为解决优先级抽样可能导致的样本多样性降低问题,根据样本的优先级将抽样概率定义为:

$$\psi_j = \frac{p_j^\alpha}{\sum_k p_k^\alpha} \quad (20)$$

式中: α 为优先级控制指标,当 $\alpha = 0$ 时,抽样方式退化为均匀采样。在这种抽样方式下,即使是优先级最低的经验数据,其采样概率也不会为零,从而确保了样本的多样性。针对优先级抽样带来的偏差问题,采用重要性抽样(Importance-Sampling, IS)权重进行纠正。定义为:

$$\omega_j = \frac{(|D|\psi_j)^{-\beta}}{\max_{i \in D} \omega_i} \quad (21)$$

式中: β 用于控制校正过程,在进行更新参数时,使用 $\omega_j \delta_j$ 而不是 δ_j ,从而保证训练过程更加接近真实环境。因此,改进后的损失函数定义为:

$$Loss(\theta) = \omega_j (y_j - Q(s_j, a_j; \theta))^2 \quad (22)$$

式中:目标值函数 $y(t)$ 定义为:

$$y_j = r_j + \gamma Q(s_{j+1}, \arg\max_a Q(s_{j+1}, a_j; \theta); \theta') \quad (23)$$

式中: γ 为折扣因子。

综上所述,POOA-UN优化算法的伪代码为:

输入:任务信息 $D(t)$, $C(t)$, $\xi(t)$, 距离信息 $d(t)$, 计算资源 $f(t)$, $f_i(t)$

输出:卸载比例 $\lambda(t)$ 和卸载节点 $\varphi_m(t)$

1. 初始化评估网络和目标网络参数 θ, θ'

2. 初始化经验池 Buffer

3. FOR $episode = 1$ TO M DO

4. 随机初始化状态 $s(1)$

5. FOR $t = 1$ TO T DO

6. 根据改进后的贪婪策略选择动作

7. 执行动作 $a(t)$, 得到即时奖励 $r(t)$ 和下一个状态 $s(t+1)$

8. 根据式(20)计算经验 (s_t, a_t, r_t, s_{t+1}) 的优先级并存入经验池 Buffer

9. 按照优先级经验重放采样 batch size 个样本进行训练

10. 根据式(21)计算 mini batch 的 IS 权重

11. 根据式(22)计算损失函数 $Loss(\theta)$

12. 通过梯度下降反向传播更新评估网络权重 θ

13. 每 C 步更新目标网络权重 $\theta' = \theta$

14. END FOR

15. END FOR

3 实验方法与结果分析

3.1 实验环境与参数设置

使用 Python 语言在 Anaconda 平台上运行算法,利用开源机器学习框架 Tensorflow 构建和训练神经网络。在搭建仿真环境时,本文设计了一种随机修改状态空间的机制,以模拟 MEC 场景中传输网络的不可靠性。在计算卸载过程中,该机制会依据预设概率值对模型接收到的状态空间中的任务信息进行随机修改,使仿真环境更贴近 MEC 实际应用场景的真实性和复杂性。此外,场景中包含多个均匀分布在系统覆盖范围内的 ES 以及一个自由移动的 MD, MD 的位置及其运动轨迹在系统中随机生成。对于 MD 产生的随机任务,假设其数据量在 $[0.5, 10]$ MB 范围内均匀分布,完成任务所需的 CPU 周期均匀分布在 $[0.4, 8]$ GHz 范围内。为简化模型,假设 ES 的可用计算资源在 $[5, 20]$ GHz/s 内均匀分布,MD 的计算能力为 $f_i = 1.26\text{GHz/s}^{[21]}$ 。其余仿真参数设置如表 1 所示。通过大量实验确定强化学习的超参数,初始化网络学习率为 0.005,经验缓冲区大小为 10 000,批学习大小为 16,奖励折扣因子为 0.95。

Table 1 Simulation environment parameters setting

表 1 仿真环境参数设置

参数	数值
MD 的传输功率 p/mW	500
信道带宽 B/MHz	5
噪声功率谱密度 $N_0/\text{dBm/Hz}$	-174
路径损耗常量 g_0	-40
路径损耗指数 σ	4
参考距离 Dis_0/m	100
有效电容系数 k	10^{-28}

3.2 性能分析

为验证 PPOA-UN 算法的性能,将其与以下 3 种基准方法进行比较:①随机卸载(Random Offload, RO)。该方法通过随机分配任务卸载比例和卸载节点进行卸载;②DDQN。该方法是一种基于值的 DRL 算法,通过在 DQN 的基础上引入两个独立网络来解决动作值高估的问题;③基于优先级双深度 Q 网络的卸载策略(PER-Double DQN, PDDQN)。该方法使用 PER 机制,能从经验缓冲区中优先考虑期望学习值较高的经验,从而提高算法的学习效率。

实验在具有相同设置的仿真环境下进行,使用相同的任务序列和移动轨迹来训练 DDQN、PDDQN 和 PPOA-UN,收敛性能比较如图 4 所示。可以看出,在训练初始阶段,由于缺乏对环境的先验知识,各方案的用户成本均处于较高水平。随着 3 种模型持续与环境进行交互,其卸载成本

均呈现下降趋势,表明其均成功学习到有效的卸载策略。然而,当训练进行到150回合后,POOA-UN的卸载成本开始显著低于其他两种强化学习算法,且下降速度更快,表明即使在传输网络不可靠的情况下,其仍能实现高效卸载。这一优势源于POOA-UN的预测模块在持续的训练中逐渐成熟,能够较为准确地预测任务信息并选择最优卸载节点。结合预测结果,POOA-UN可以在任务信息丢失的情况下作出正确的卸载决策,并将学习经验存储至经验缓冲区,有效避免了因任务数据丢失而产生不合理的学习经验,最终提升了经验缓冲区的可靠性。结合PER机制对关键经验的重视,POOA-UN在训练过程中的收敛速度进一步加快。综上所述,与对照算法相比,POOA-UN在传输网络不可靠的环境下表现出更优越的卸载性能和更快的收敛速度。

为更具体地比较各方法的卸载性能,采用系统成本缩减比率(System Total Cost Reduction Ratio, STCRR)作为评价指标^[22]。计算公式为:

$$STCRR = \frac{Cost_{RO} - Cost_{other}}{Cost_{RO}} \quad (24)$$

式中: $Cost_{RO}$ 为随机卸载的卸载成本; $Cost_{other}$ 为除随机卸载以外的其他方案的卸载成本。

不同算法的STCRR结果比较如表2所示。可以看出,与DDQN和PDDQN相比,POOA-UN的STCRR分别提升了18%和10%,而PDDQN仅比DDQN提升了8.5%。这是由于不可靠的传输网络可能会导致任务数据丢失,进而在经验缓冲区中积累大量无效经验。在这种情况下,即使采用PER的采样方式,现有算法依然难以有效学习到更优的卸载策略。

3.3 任务信息预测

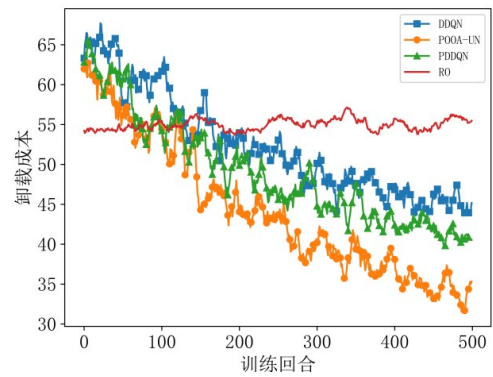
在任务信息预测研究中,通常使用前 n 次产生的任务特征来预测第 $n+1$ 次任务的特征^[16]。本文设计多组实验探究不同历史序列长度对LSTM预测模型的影响,并将预测结果的准确率作为评价模型性能的主要指标。该指标定义为:

$$准确率 = \left(1 - \frac{|\text{预测值} - \text{真实值}|}{|\text{真实值}|} \right) * 100\% \quad (25)$$

分别测试 $n = 10, 20, 30$ 时的预测精度,结果见图5。可以看出,整体精度呈现出先升后降的趋势,分别为78.45%、82.14%和78.12%。实验结果表明,过大的历史窗口可能导致模型过拟合,从而降低预测精度。为了更好地拟合实际任务数据,本文选择 $n = 20$ 的窗口大小。

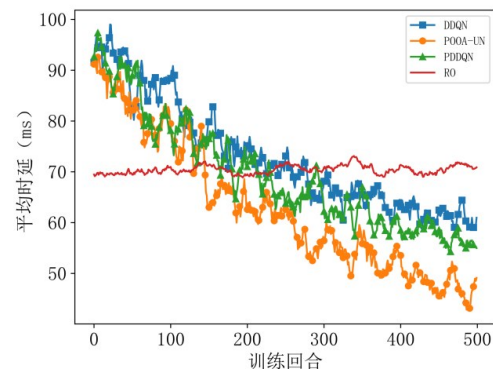
3.4 多类型任务卸载性能测试

在MEC的应用场景中,不同的计算任务对延迟和能耗有不同的要求。例如,智能驾驶技术需要处理高精度的地图数据,对实时性要求很高;而无人机视频监控因受到电池容量的限制,通常对能耗要求较高。为评估各卸载方案在多类型应用程序中的适应性,本文选取3种典型应用场



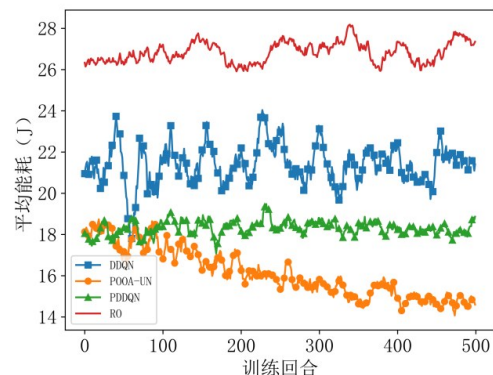
(a) Comparison of offloading cost

(a) 卸载成本比较



(b) Comparison of average delay

(b) 平均时延比较



(c) Comparison of average energy consumption

(c) 平均能耗比较

Fig. 4 Convergence performance of different algorithms

图4 各算法收敛性能比较

Table 2 Comparison of STCRR results of different algorithms

表2 不同算法STCRR结果比较

算法	STCRR/%
DDQN	18.53
PDDQN	26.91
POOA-UN	36.33

景作为代表,分别为增强现实(Augmented Reality, AR)、信息娱乐应用(Infotainment Application, IA)、视频监控应用(Video Surveillance Application, VSA)^[22]。为保证用户的沉浸感和交互体验,AR对实时性要求极高。因此,在卸载

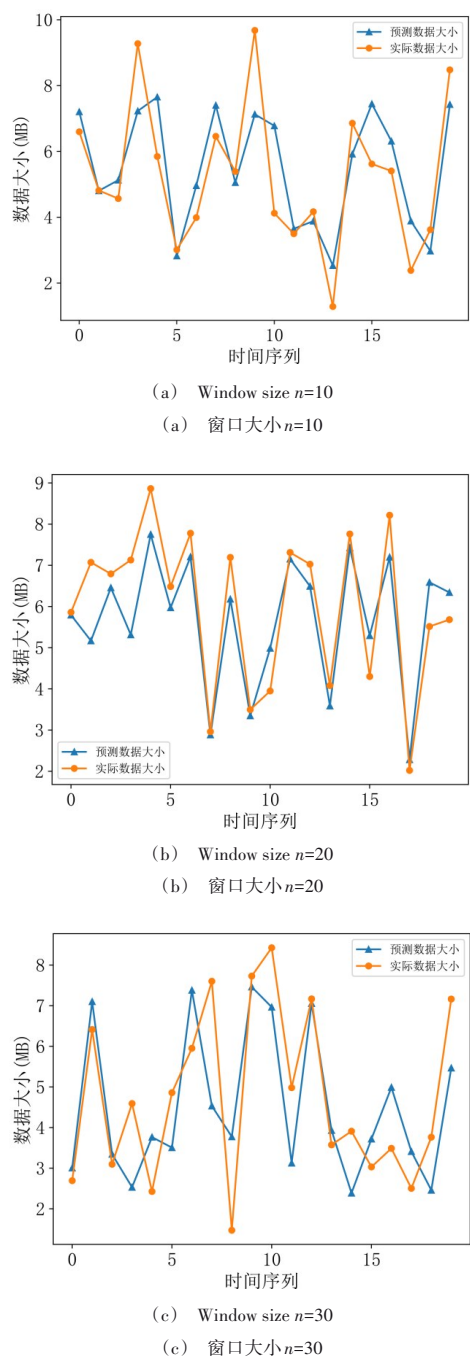


Fig. 5 Effect of different window sizes on the prediction result

图5 窗口大小对预测结果的影响

成本中,时延的权重占比最大;IA 则需要在用户体验与能量消耗之间取得平衡;VSA 由于涉及大量视频分析,且这一过程耗能较大,对能耗更为敏感。不同的性能需求通过不同的权重占比来体现,各应用的权重偏好设置如表 3 所示。

图 6 展示了不同算法在不同应用场景下的卸载成本、平均时延和平均能耗。如图 6(a) 所示,POOA-UN 算法在处理具有不同偏好设置的应用时,其卸载表现优于其他对照算法,表明其能够更好地平衡各种应用的卸载需求,展现出更强的鲁棒性。由图 6(b)可以看出,POOA-UN 的任

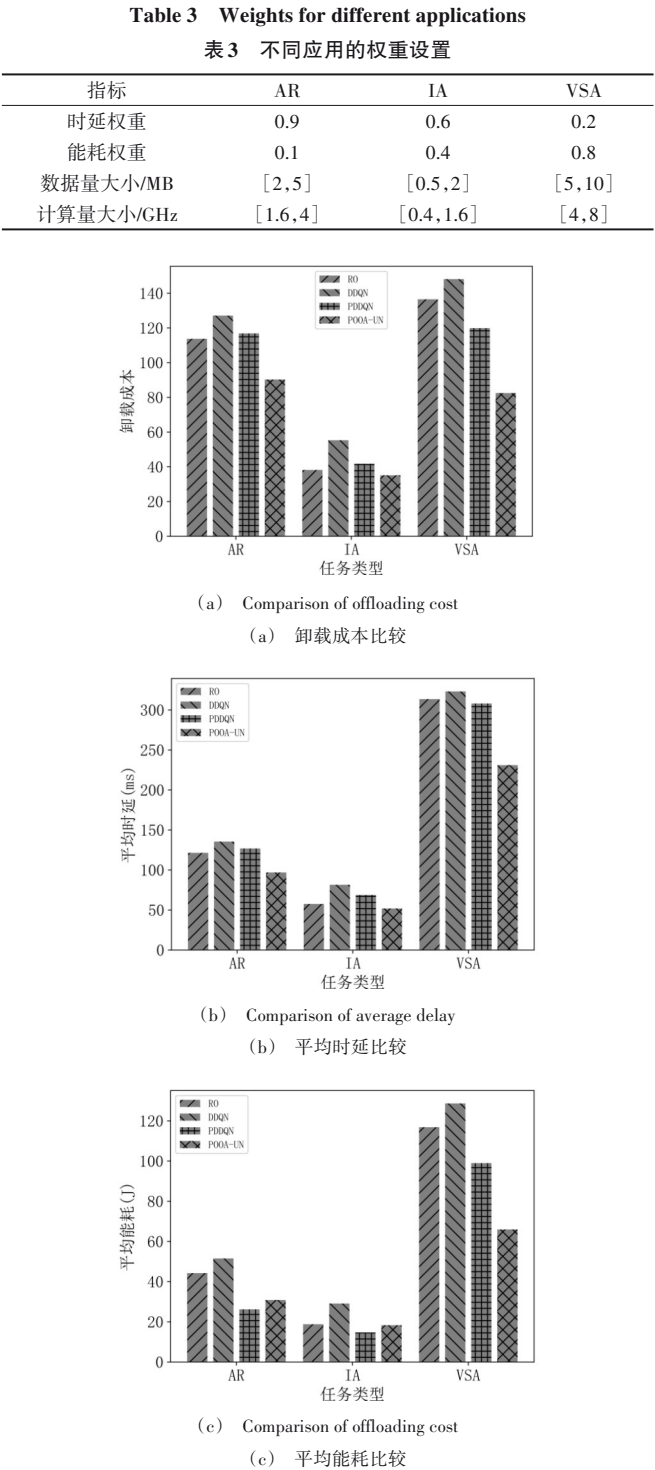


Fig. 6 Comparison of unloading cost, average delay and average energy consumption of different algorithms in different application scenarios

图 6 不同算法在不同应用场景下的卸载成本、平均时延和平均能耗比较

务处理时延总是低于其他算法。在某些应用中,RO 表现出比其他算法更低的时延,这主要是由于 MD 的计算资源十分有限,而 ES 则拥有更加充足的计算资源。在不考虑能耗或卸载成本的情况下,将更多的任务卸载到 ES 进行处理可获得更低的时延。然而,在传输网络复杂且不可靠

的 MEC 环境下,RO 方法可能导致 MEC 系统负载不均衡和无线通信压力。在网络复杂且不稳定的环境下,大规模的数据传输还将增加卸载失败的风险,从而影响任务的正确执行。因此,这种方案在实际场景中不具备适用性。图 6(c)展示了各算法对不同类型应用程序的卸载能耗。可以看出,由于 AR 和 IA 应用具有低延迟的需求,POOA-UN 在这两类应用程序的卸载能耗上并未达到最低指标。这是由于这两类应用程序对能耗的敏感度较低,若为了节省能量而牺牲时延性能,将严重影响用户满意度。而对于 VSA,POOA-UN 算法表现出最低的卸载能耗,这反映出 POOA-UN 对该类应用的能耗需求更为重视,并在同等情况下优先降低该类任务的能耗。

4 结语

本文针对 MEC 传输信道不可靠导致现有卸载方案性能不足的问题,提出融合 DRL 与任务信息缺失补全模型的计算卸载优化算法 POOA-UN。该算法设计了一个基于 LSTM 的信息缺失补全模型,通过对未来时序性任务信息进行预测,有效解决了不可靠场景下边缘计算任务信息缺失的问题;同时结合改进的 ϵ -贪婪策略和 PER,进一步提高了模型的收敛速度和稳定性,降低了任务执行的时延和能耗。仿真实验结果表明,与 RO、DDQN 和 PDDQN 方法相比,POOA-UN 在传输信道不可靠的 MEC 场景下的卸载成本分别降低了 36.3%、21.8% 和 12.9%,并表现出更快更稳定的收敛性能和更强的鲁棒性。

在多用户 MEC 环境中,集中决策方案会降低系统的卸载效率,并导致无线传输信道拥堵。未来工作将使用多智能体 DRL 算法解决多用户、多服务器场景下的信道传输不可靠问题。将智能体部署在 MD 上的分布式卸载方案不仅能提升决策效率,而且对复杂的 MEC 环境具有更强的适应性和可拓展性,可实现网络性能的最优化。

参考文献:

- [1] CHEN W, CHEN Y H, WU J X, et al. A multi-user service migration scheme based on deep reinforcement learning and SDN in mobile edge computing[J]. *Physical Communication*, 2021, 47: 101397.
- [2] TANG M, WONG V. Deep reinforcement learning for task offloading in mobile edge computing systems[J]. *IEEE Transactions on Mobile Computing*, 2022, 21(6): 1985–1997.
- [3] ZHANG J N, ZHAO X Y. An overview of user-oriented computation offloading in mobile edge computing[C]// Beijing: 2020 IEEE World Congress on Services (SERVICES), 2020.
- [4] CHEN J, XING H L, XIAO Z W, et al. A DRL agent for jointly optimizing computation offloading and resource allocation in MEC[J]. *IEEE Internet of Things Journal*, 2021, 8(24): 17508–17524.
- [5] IMTIAZ H H, TANG S H. Multi-task partial offloading with relay and adaptive bandwidth allocation for the MEC-assisted IoT [J]. *Sensors*, 2022, 23(1): 190.
- [6] TANG Q, CHANG L, YANG K, et al. Task number maximization offloading strategy seamlessly adapted to UAV scenario[J]. *Computer Communications*, 2020, 151(c): 19–30.
- [7] WANG Z Y, SU X, XIN Y X. Multi-access edge computing offloading method oriented to offshore scenarios[C]//Xiamen: 2021 IEEE/CIC International Conference on Communications in China (ICCC), 2021.
- [8] ALE H, ZHANG N, FANG X J, et al. Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning[J]. *IEEE Transactions on Cognitive Communications and Networking*, 2021, 7(3): 881–892.
- [9] CHENG M Q, SONG X Q, LEI L. AI-enabled offloading decision-making and resource allocation optimization under emergency scenarios[C]// Kuala Lumpur: 2023 IEEE Globecom Workshops, 2023.
- [10] RAIVI A M, MOH S. JDACO: joint data aggregation and computation offloading in UAV-enabled Internet of Things for post-disaster scenarios [J]. *IEEE Internet of Things Journal*, 2024, 11(9): 16529–16544.
- [11] DU T Y, LI C L, LUO Y L. Latency-aware computation offloading and DQN-based resource allocation approaches in SDN-enabled MEC [J]. *Ad Hoc Networks*, 2022, 135(c): 102950.
- [12] LI G Y, XUE X, LI Y, et al. Improved TD3 edge computing offloading strategy for software defined networking Internet of Vehicles[J]. *Computer Integrated Manufacturing Systems*, 2023, 29(5): 1627–1634.
李国燕, 薛翔, 刘毅, 等. 改进 TD3 的 SDN 车联网边缘计算卸载策略[J]. *计算机集成制造系统*, 2023, 29(5): 1627–1634.
- [13] BI J, YUAN H T, ZHANG K Y, et al. Energy-minimized partial computation offloading for delay-sensitive applications in heterogeneous edge networks[J]. *IEEE Transactions on Emerging Topics in Computing*, 2022, 10(4): 1941–1954.
- [14] NING Z L, DONG P R, WANG X J, et al. Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks [J]. *IEEE Transactions on Mobile Computing*, 2022, 21(4): 1319–1333.
- [15] ZHAO S X, YUAN L, ZHANG Z P. Multi-agent edge computing task offloading [J]. *Computer Engineering and Applications*, 2022, 58(6): 177–182.
赵庶旭, 元琳, 张占平. 多智能体边缘计算任务卸载[J]. *计算机工程与应用*, 2022, 58(6): 177–182.
- [16] MIAO Y M, WU G X, LI M, et al. Intelligent task prediction and computation offloading based on mobile-edge cloud computing[J]. *Future Generation Computer Systems*, 2020, 102(c): 925–931.
- [17] WEI Y F, WANG Z Y, GUO D, et al. Deep Q-learning based computation offloading strategy for mobile edge computing[J]. *Computers Materials & Continua*, 2019, 59(1): 89–104.
- [18] AI-HAMMADI I, LI M C, ISLAM S. Independent tasks scheduling of collaborative computation offloading for SDN-powered MEC on 6G networks [J]. *Soft Computing*, 2023, 27(14): 9593–9617.
- [19] LI Z J, YU K, ZHOU H, et al. DQN-based collaborative computation offloading for edge load balancing [C]//Beijing: 2023 8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), 2023.
- [20] SHI H G, TIAN Y X, LI H J, et al. Task offloading and trajectory scheduling for UAV-enabled MEC networks: an MADRL algorithm with prioritized experience replay[J]. *Ad Hoc Networks*, 2024, 154: 103371.
- [21] WANG L Y, ZHANG G L. Deep reinforcement learning based joint partial computation offloading and resource allocation in mobility-aware MEC system[J]. *China Communications*, 2022, 19(8): 85–99.
- [22] CHEN S Y, CHEN H P, RUAN J T, et al. Context-aware online offloading strategy with mobility prediction for mobile edge computing[C]//Athens: 2021 International Conference on Computer Communications and Networks (ICCCN), 2021.

(责任编辑:尹晨茹)