

DOI: 10.13718/j.cnki.xdzk.2026.01.018

唐山林, 吴涛, 周启钊. MEC 中基于前景理论的混合任务风险卸载与资源分配 [J]. 西南大学学报(自然科学版), 2026, 48(1): 229-239.

MEC 中基于前景理论的 混合任务风险卸载与资源分配

唐山林, 吴涛, 周启钊

成都信息工程大学 计算机学院, 成都 610225

摘要: 移动边缘计算(Mobile Edge Computing, MEC)技术的发展有效缓解了移动设备计算和存储需求与系统资源受限之间的矛盾。通过将计算任务卸载至移动边缘计算服务器,可以显著提升移动设备的服务质量。然而,在实际网络环境中,任务卸载存在潜在的失败风险,导致任务仍然需要在本地计算,从而产生延迟等额外开销。因此,如何制定合理的卸载决策并优化资源分配,以适应不同任务类型,仍然是 MEC 系统面临的关键挑战。针对上述问题,基于异步优势动作—评价(Asynchronous Advantage Actor-Critic, A3C)框架,提出一种混合任务卸载与资源分配算法,该算法结合前景理论,以平衡任务卸载的风险与收益,确保更合理的决策。此外,根据任务的截止时间确定其优先级,并据此分配 MEC 计算资源,保障高优先级任务获得所需的计算资源。算法考虑了卸载失败的风险,并利用深度强化学习自适应优化卸载决策,以适应动态网络环境。仿真结果表明,与基线算法相比,所提算法在获得更优卸载方案的同时,有效降低了任务延迟和能耗。

关键词: 移动边缘计算; 任务卸载; 资源分配

中图分类号: TP393

文献标识码: A

文章编号: 1673-9868(2026)01-0229-11

开放科学(资源服务)标识码(OSID):



Prospect Theory-Based Risk Offloading and Resource Allocation for Hybrid Tasks in Mobile Edge Computing

Tang Shanlin, Wu Tao, Zhou Qizhao

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

Abstract: The development of mobile edge computing (MEC) technology has effectively alleviated the contradiction between the growing computational and storage demands of mobile devices and their limited system resources. By offloading computational tasks to MEC servers, the service quality of mobile devices can be significantly enhanced. However, in real-world network environments, task offloading is subject to potential failure risks, leading to local execution and additional latency overhead. Therefore, making

收稿日期: 2025-01-02

基金项目: 四川省科技研究开发项目(2024ZHCG0195, 2023jdzh0034); 成都信息科技大学人才引进研究启动项目(kytz202269)。

作者简介: 唐山林, 硕士, 主要从事边缘计算的研究。

通信作者: 周启钊, 博士, 副教授。

reasonable offloading decisions and optimizing resource allocation to accommodate various task types remain critical challenges in MEC systems. To address these challenges, this paper proposed a hybrid task offloading and resource allocation algorithm based on the asynchronous advantage actor-critic (A3C) framework. The algorithm integrated prospect theory to balance the risks and rewards associated with task offloading, ensuring more rational decision-making. Additionally, task priorities were determined based on their deadlines, and MEC computing resources were allocated accordingly to guarantee that high-priority tasks received the necessary computational resources. The proposed algorithm considered the risk of offloading failure and leveraged deep reinforcement learning to adaptively optimize offloading decisions in dynamic network environments. Simulation results demonstrated that, compared to baseline algorithms, the proposed approach achieved superior offloading strategies while effectively reducing task delay and energy consumption.

Key words: mobile edge computing; task offloading; resource allocation

5G 技术的快速发展推动了互联设备的爆炸式增长,如移动设备、传感器和物联网终端等,这些设备已广泛融入现代社会的各个领域^[1]。然而,移动设备面临着计算能力的固有限制^[2],需要将计算任务卸载到边缘网络以提升服务质量。这种对移动边缘计算(Mobile Edge Computing, MEC)的依赖加剧了移动网络的负担,特别是 MEC 计算资源的有限性难以满足用户不断增长且多样化的计算需求^[3-4]。此外,现实世界网络环境的复杂性、多样性以及任务间错综复杂的依赖关系进一步复杂化了卸载过程,凸显了对有效解决方案的迫切需求^[5]。

传统的集中式网络架构常面临高延迟及不可预测的服务质量下降问题,而 MEC 提供了一种可行的替代方案,有望缓解这些挑战^[1, 6-7]。相比云计算,MEC 具备多项优势,包括通过分布式架构降低延迟并增强对用户需求的响应能力。尽管 MEC 带来了显著的性能提升,它和云计算仍然面临类似于区块链挖掘过程中观察到的风险^[8-9],例如环境因素和网络波动可能导致的任务卸载失败。有效应对这些挑战对于优化 5G 及未来移动网络的性能和可靠性至关重要。

在评估 MEC 相关风险时,前景理论提供了重要的理论框架。该理论强调个体在不确定环境下的决策行为,特别是如何在风险与回报之间权衡选择^[10]。前景理论指出,个体在面临潜在损失时往往更倾向于冒险,而在面临潜在收益时则更倾向于风险规避。这一特性与 MEC 中的任务卸载决策密切相关,因为移动用户和网络运营商需评估任务卸载的潜在风险和回报,包括延迟、服务质量和计算资源的可用性^[11]。因此,深入理解不同情境下的用户决策行为,有助于制定更高效的任务卸载策略,降低潜在的服务中断风险,从而优化网络整体性能^[12-13]。此外,将前景理论应用于风险评估不仅能更准确地预测用户对 MEC 业务的接受程度,还能促进 MEC 在 5G 环境中的广泛采用。

为此,本文提出了一种基于前景理论的 MEC 混合任务风险卸载和资源分配(Prospect Theory-Based Risk Offloading and Resource Allocation, PTRORA)算法。该算法采用前景理论衡量风险与收益的平衡关系,并基于任务的最大可容忍时延确定其优先级。随后,通过异步优势动作—评价(Asynchronous Advantage Actor-Critic, A3C)算法优化计算资源分配,以提高任务执行效率。本文的主要贡献如下:

1) 针对现实环境中任务卸载失败带来的挑战,本文提出了一种基于前景理论的卸载决策方法,以平衡卸载风险与收益。该方法能够更细致评估风险,改进不确定性下的决策,并通过 A3C 算法实现动态任务卸载和资源分配优化。

2) 针对任务服务质量问题,提出了一种基于任务最大可容忍时延的资源分配策略,构建一个优先级驱动的高效资源分配框架。该策略通过任务分类提高资源分配的精度,在保证系统性能的同时有效降低时延。

3) 仿真结果表明, PTRORA 算法在卸载决策过程中充分考虑了卸载失败风险,表现出优越的性能。实验结果进一步验证了所提算法在降低任务时延和能耗方面的有效性,相较于基线方法,实现了更优的用户体验与系统可靠性。

1 相关工作

随着云服务向边缘的扩展, MEC 在降低任务执行延迟、提升服务质量方面展现出显著优势。这一转变使得卸载优化成为 MEC 中的一个关键问题^[12]。近年来, 研究人员对卸载决策的优化展开了广泛研究, 重点关注如何在计算资源有限的情况下, 实现高效的任务调度与分配^[14]。相关研究主要致力于最小化任务延迟, 提高能效并优化整体服务质量^[15]。

国内外在 MEC 任务卸载方面取得了诸多研究进展, 但 MEC 中的动态因素, 如网络带宽波动和用户移动性, 依然可能导致卸载失败从而降低业务性能^[16]。为应对这些问题, 近年来机器学习算法被引入到 MEC 任务卸载领域, 以增强卸载决策在动态环境下的适应性^[17-18]。文献[19]提出了一种面向任务依赖关系的优化策略, 旨在提高动态 MEC 环境下卸载决策的效率和可靠性。该方法动态调整资源分配并优化跨边缘设备的任务执行, 采用 A3C 算法有效降低延迟并提高系统性能。文献[9]探讨了 MEC 中的区块链挖矿任务卸载问题, 基于前景理论构建博弈模型, 以优化矿工与 MEC 服务器的效用, 为计算卸载提供了一种新的卸载思路。文献[20]研究了一种最小化长期成本的任务卸载模型, 针对不可分割的、延迟敏感的任务以及边缘负载动态性问题进行优化。其方法允许设备自主做出卸载决策, 以减少任务丢失率和平均延迟, 而无需获取其他设备的任务信息或卸载选择。文献[5]提出了一种面向多类型任务的均衡卸载算法, 该算法在 MEC 内对卸载任务进行优先级划分, 并根据优先级和负载情况执行不同的卸载决策, 在降低系统时延的同时, 提高任务完成率。

此外, 博弈论与强化学习也被广泛应用于解决 MEC 中的资源竞争与协同优化问题。例如, 文献[21]提出一种基于 Stackelberg 博弈的资源分配策略, 通过粒子群算法求解最优定价与资源分配, 以激励用户共享闲置资源。在强化学习方面, 文献[22]采用深度确定性策略梯度 (DDPG) 算法对车联网中的任务卸载与资源分配问题进行建模与优化, 有效降低了任务处理时延。

上述研究为 MEC 中的任务卸载决策和资源分配提供了有效的基础思路。然而, 现有研究方案忽视了 MEC 任务卸载失败造成的影响, 这将可能直接导致系统性能显著下降, 特别是在延迟敏感的应用程序中^[23]。为此, 本文提出了一种基于前景理论的卸载框架以最小化卸载失败对系统性能的影响。该方法根据任务优先级定义, 采用 A3C 算法分配 MEC 计算资源, 确保不同任务的时延需求得到满足, 解决了任务卸载失败对系统性能的负面影响, 提升了系统的整体稳定性和服务质量。

2 系统模型和问题描述

本文所提系统模型如图 1 所示, 其中包含多个移动设备 (Mobile Devices, MDs) 和一个移动边缘计算 MEC 基站。MDs 生成两种类型的任务: 独立任务和依赖任务。任务可通过无线网络转移到 MEC 基站进行计算, 也可以完全在本地处理。

2.1 混合任务模型

MDs 产生多个任务, 每个任务具有不同的依赖关系。本文将任务集分为独立任务和具有线性依赖关系的任务, 后者采用有向无环图 (Directed Acyclic Graph, DAG) 表示。针对 DAG 任务的卸载和资源分配策略, DAG 记作 $G=(V, L)$, 其中 $V=\{V_1, \dots, V_M\}$ 表示包含 M 个任务的集合, L 代表任务之间的依赖关系。 $e_{a, a+1} \in L$ 表示任务 V_{a+1} 必须在任务 V_a 计算完成后才能执行计算^[19], $a=1, 2, \dots, M-1$ 。任务 V_a 输入数据大小为 S_a (单位为 bit), C_a 是完成 1 bit 任务所需的 CPU 周期数, 则任务 V_a 所需的总 CPU 周期数为 $I_a=S_a \times C_a$ 。计算中间结果记为 ΔV_a , 任务卸载决策集合为 $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_M\}$, 其取值为 1 或者 0, 其中

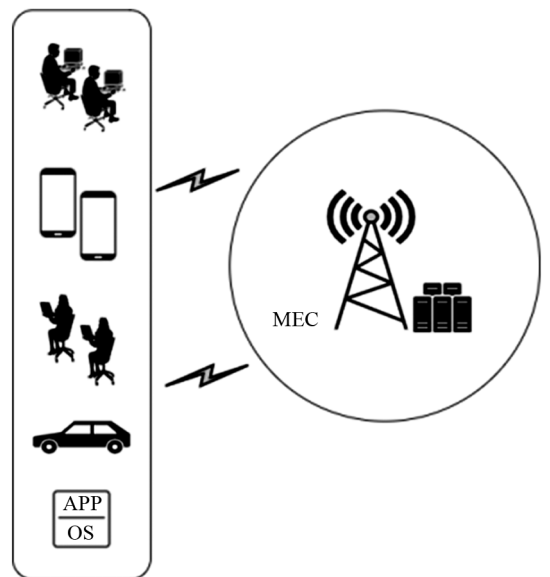


图 1 系统模型

0 表示任务在本地执行计算, 1 表示任务卸载至 MEC 服务器计算。例如: 决策集合 $\{0, 0, 1, 0, 0\}$ 表示第三个任务卸载到 MEC 上执行, 其余任务均在本地执行。对于独立任务, 其执行不受依赖关系的约束。表 1 为本文所用变量列表。

表 1 变量符号及其数值

变量名称及符号	变量值	变量名称及符号	变量值
总任务数 M	16	移动设备的发射功率 P	0.1 W 或 0.2 W
卸载决策 α	0 或 1	任务成功卸载的概率 ρ	0.5
信道噪声功率密度 σ^2	10^{-10}	时延分类常量 ρ	50 ms
无线链路上行带宽 B_{up}	1.5 MHz	任务卸载失败的惩罚因子 D	0.01
无线链路下行带宽 B_{down}	7 MHz	整数常量 η	4
任务最大容忍时延 T_{max}	50 ms~200 ms	异步线程数 N	3
本地计算能力 f	0.5 GHz	任务优先级 κ	$\{1, 2, 3, 4\}$
任务本地计算时的功率 γ	0.5 W	理性参数 β	3

2.2 本地计算模型

如果任务在本地执行, 则卸载决策用 0 表示。为简化计算, 第一个依赖任务被视为独立任务, 因为它只影响后续任务。当前驱节点和后继节点都在本地计算时, 中间结果传输时间可以忽略不计, 则独立任务或依赖任务第一项的本地计算时延可表示为:

$$T_{V_a, local} = \frac{I_a}{f} \quad (1)$$

其中: I_a 为任务所需要的总 CPU 周期数; f 表示本地计算能力。

本地计算的能耗可表示为:

$$E_{V_a, local} = T_{V_a, local} \times \gamma \quad (2)$$

其中 γ 表示任务 V_a 本地执行时的计算功率。

2.3 卸载模型

当任务卸载到 MEC 进行计算时, MDs 通过无线链路传输卸载后的任务, 则数据传输速率为^[19]:

$$R = B \cdot \ln\left(1 + \frac{P \cdot h}{\sigma^2}\right) \quad (3)$$

其中: B 为 MDs 和 MEC 之间无线链路的带宽; σ 为信道噪声功率密度; P 为 MDs 的发射功率; h 为链路的信道增益。假设上传和下载的信道增益相同。任务卸载到 MEC 执行由传输时延和计算时延两部分组成, 其中上传时延和能耗表示为:

$$T_{V_a, up} = \frac{S_a}{R_{up}} \quad (4)$$

$$E_{V_a, up} = T_{V_a, up} \cdot P \quad (5)$$

其中 R_{up} 为上传速率。MEC 的计算时延如下表示:

$$T_{V_a, MEC} = \frac{I_a}{F} \quad (6)$$

其中 F 表示 MEC 根据任务优先级分配的计算能力。

依赖任务的中间数据会产生一定的下载延迟和能耗, 可分别表示为:

$$T_{\Delta V_a, down} = \frac{\Delta V_a}{R_{down}} \quad (7)$$

$$E_{\Delta V_a, down} = T_{\Delta V_a, down} \cdot P \quad (8)$$

其中 R_{down} 为下载速率。因此, 任务 V_a 在 MEC 的总时延表示为:

$$T_{MEC, total} = T_{V_a, up} + T_{V_a, MEC} + T_{\Delta V_a, up} \quad (9)$$

根据式(5)、(8), 任务 V_a 执行的能耗可由下式得到:

$$E_{\text{MEC}, \text{total}} = E_{V_a, \text{up}} + E_{\Delta V_a, \text{down}} \quad (10)$$

MDs 的总时延 T 和总能耗 E_n 可以分别表示为:

$$T = \sum_a^{n+1} \sum_j^M \{ \alpha_a \cdot T_{\text{MEC}, \text{total}} + (1 - \alpha_a) \cdot T_{V_a, \text{local}} \} \quad (11)$$

$$E_n = \sum_a^{n+1} \sum_j^M \{ \alpha_a \cdot E_{\text{MEC}, \text{total}} + (1 - \alpha_a) \cdot E_{V_a, \text{local}} \} \quad (12)$$

3 基于前景理论的卸载框架

3.1 前景理论卸载框架

前景理论用于描述决策在不确定环境下的行为特点, 即决策不仅关注预期结果, 还关注预期结果与实际结果之间的差距。当决策处于盈利状态时, 决策者倾向于规避风险; 而处于亏损状态时, 则更倾向于承担风险。本文选用 Tversky 函数作为 A3C 模型的奖励函数^[24]。参考文献[8], 将参考点设为 0 增益值, 任务的效用值表示如下:

$$u(t) = \begin{cases} g_n(t)^{\mu_n} & g_n(t) \geq 0 \\ -\lambda(-g_n(t))^{\xi_n} & g_n(t) < 0 \end{cases} \quad (13)$$

其中: $g_n(t)$ 是适应度函数, 即不考虑前景理论的原始价值函数; μ 、 λ 和 ζ 为前景参数, 且 $0 < \mu$, $\zeta < 1$, $\lambda > 1$ 。

本文选择 Prelec 函数作为概率权重函数, 其原因与选择 Tversky 函数相同, 两者都用于捕获风险下的决策, 同时在风险感知中纳入了不对称性^[24]。Prelec 函数表示如下:

$$w(p) = \frac{p^\beta}{(p^\beta + (1-p)^\beta)^{\frac{1}{\beta}}} \quad (14)$$

其中: β 为理性参数, 其值越小, 表示离绝对理性下的理想结果越远; p 是任务卸载成功的概率, $p \in [0, 1]$, 如果卸载失败将导致惩罚, 之后任务将在本地执行。当卸载成功时, 适应度值为:

$$fit_{\text{success}} = T_{\text{MEC}, \text{total}} \quad (15)$$

当卸载失败时, 适应度值为:

$$fit_{\text{fail}} = T_{V, \text{local}} + D \quad (16)$$

其中 D 为惩罚因子。由此, $g_n(t)$ 表示为:

$$g_n(t) = e^{-((fit_{\text{success}} \cdot w(p) + fit_{\text{fail}} \cdot (1-w(p))) \cdot \alpha + (1-\alpha) \cdot T_{V, \text{local}}) \ln \gamma} \quad (17)$$

3.2 任务优先级定义

当任务卸载成功时, 将为其定义优先级, 并根据任务的优先级分配 MEC 的计算资源^[5]。优先级 κ 的定义如下:

$$\kappa = \left\lceil \frac{T_{\text{max}} - T_{V_a, \text{up}}}{\rho} \right\rceil \quad (18)$$

其中: T_{max} 表示当前任务的截止时间; ρ 是一个常数, 用于对不同类型的延迟敏感任务进行分类。根据 3GPP 对延迟应用中不同类型任务的定义, 本文参照文献[25]的定义将任务分为 4 类, 即 $\kappa = 1, 2, 3, 4$, κ 值越小, 优先级越高, 分配的计算资源越多。

本文将 MEC 计算资源划分为计算能力相同的整数块, 每块的计算能力 $C = 2$ MHz, 卸载至 MEC 的任务分配至少 C 的计算能力, 任务在 MEC 上分配的计算资源表示为:

$$F_{V_a} = \begin{cases} C & \kappa = 4 \\ (4 - \kappa) \cdot C & \kappa < 4 \end{cases} \quad (19)$$

3.3 A3C 算法

A3C 作为一种深度强化学习算法, 支持在单机上使用标准的多核 CPU 或 GPU 进行多线程异步执行。在该架构下, 每个线程中的智能体独立地与环境交互, 不断学习并优化卸载策略。具体而言, 在每一轮训

练过程中：首先，每个异步智能体根据其当前状态和策略选择动作；随后环境根据该动作转换至新的状态，并给予相应的即时奖励；接着，智能体利用这些反馈信息更新全局网络参数，以提高策略的有效性。其核心目标是最大化代理的长期累积奖励，以实现最优决策。A3C 模型结构如图 2 所示。

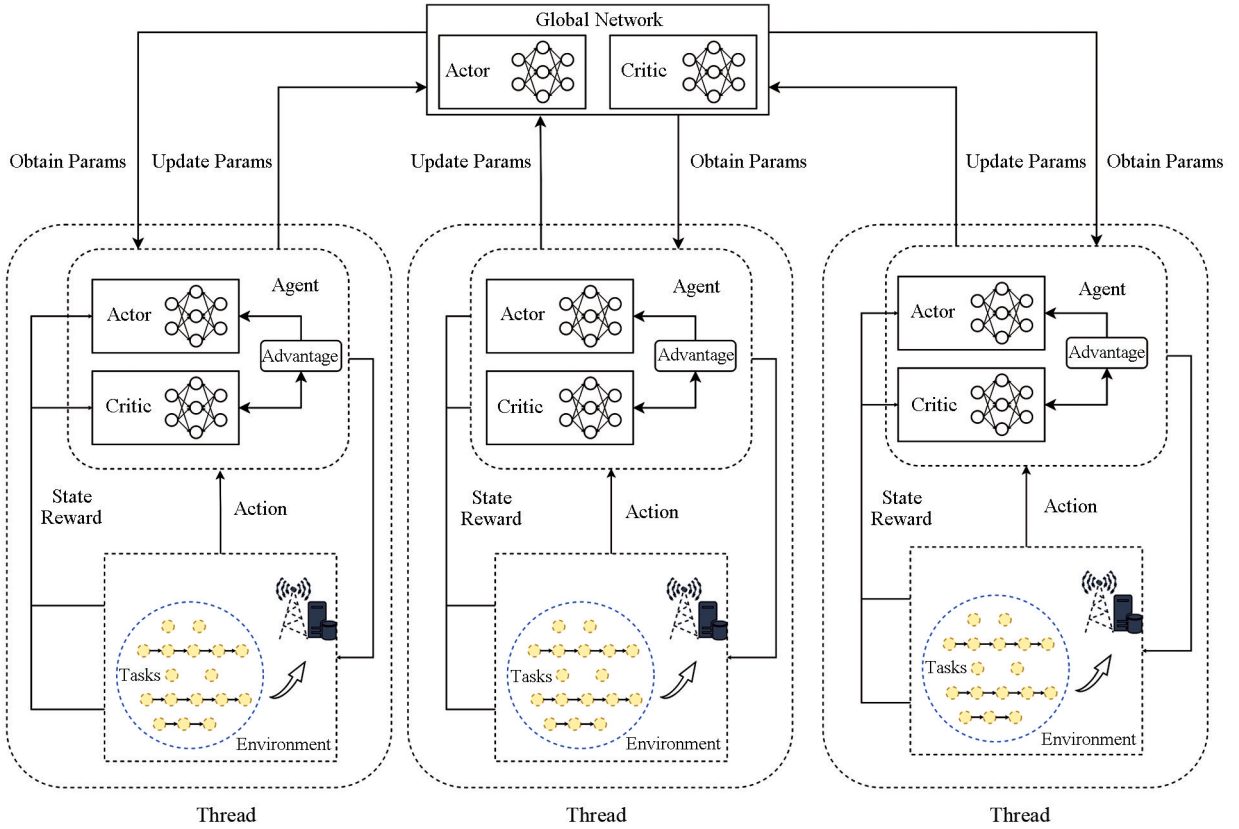


图 2 A3C 模型

本文对状态空间、动作空间和奖励函数定义如下：

1) 状态空间

$S = \{s(t), t \in \tau\}$ 表示系统空间， $s(t)$ 表示在时隙 t 的状态，包括以下参数：

① 在 t 时隙，MDs 和 MEC 之间的信道增益，这里假设上传和下载之间的增益是相同的，均为 $h(t)$ ，仅有带宽不同。

② 任务的大小 $S_a(t)$ ，所需的 CPU 周期数 $C_a(t)$ ，任务发射功率 $P_a(t)$ 和截止时间 $T_{V_a, \max}(t)$ 。

③ 任务的依赖矩阵 \mathbf{g} 。

t 时隙的系统状态可以表示为：

$$s(t) \cong [h(t), S_a(t), C_a(t), P_a(t), T_{V_a, \max}(t), \mathbf{g}] \quad (20)$$

2) 动作空间

$A = \{\alpha(t), t \in \tau\}$ 表示为动作空间。在每一个时隙 t 中， $\alpha(t) \in A$ 包含以下参数：

① 卸载决策 α 。 $\alpha = 1$ 表示卸载至 MEC 计算， $\alpha = 0$ 表示本地计算。

② 根据任务优先级分配的计算资源 $\mathcal{R}(\alpha)$ 。

因此，时隙 t 的动作可以表示为：

$$\alpha(t) \cong [\alpha, \mathcal{R}(\alpha)] \quad (21)$$

3) 奖励函数

奖励值是评价卸载策略和资源分配策略的指标，与优化目标相关。在本文的 MEC 系统中，选择 Tversky 函数作为奖励值。由于奖励值与优化目标负相关，针对时延最小化这一优化目标，我们选择幂函数作为适应度函数，其中 η 是方程(18)中的常数。奖励函数可以表示为：

$$R_{v_a} = u(t) \quad (22)$$

综上所述, PTRORA 的算法过程可以表示为:

算法 1 PTRORA 算法

1. 初始化:

(1) 创建全局共享网络: 初始化全局 Actor 网络参数 θ 和全局 Critic 网络参数 θ_v ;

(2) 初始化异步线程数 N , 全局步数 `global_step` 和线程锁, 每个线程拥有独立的本地 Actor 网络(参数 θ)和本地 Critic 网络(参数 θ_v);

(3) 初始化独立线程的网络参数;

2. FOR EPISODE=1,2,3,⋯, DO

3. 根据式(20)初始化全局训练参数和环境状态;

4. 异步线程智能体从全局网络中加载参数, 设置 $\theta' = \theta$ 和 $\theta_v' = \theta_v$;

5. FOR M=1,2,3,⋯, DO

6. 异步线程智能体根据式(17)执行动作;

7. MEC 根据式(18)为异步线程智能体分配计算资源;

8. 异步线程获取奖励和下一状态;

9. 计算本地 Actor 和 Critic 网络的梯度, 异步更新全局 Actor 和 Critic 网络参数(θ, θ_v);

10. IF `global_step mod 3 == 0` DO

11. 异步线程互斥更新全局网络;

12. 同步本地网络: 将更新后的全局网络参数(θ, θ_v)拷贝到本线程的本地网络中;

13. END IF;

14. END FOR;

15. END FOR。

4 仿真结果与讨论

本节使用 Python 进行仿真实验。根据表 1 参数设置, 本实验中 MDs 总共生成了 $M=16$ 个任务, 包括独立任务集和依赖任务集。任务大小 S 和所需的 CPU 周期数 C 均为随机生成。MDs 节点 P 的发射功率随机选择为 0.1 W 或 0.2 W, 任务截止时间 T_{\max} 在 50 ~ 200 ms 之间随机选择, 步长为 50 ms。此外, 随机生成任务的关系矩阵。实验中, 设置异步线程数 $N=3$ 、成功率 $p=0.5$ 。同时探讨了不同带宽以及移动终端路径对系统卸载决策的影响。参考文献[26]表征链路信道增益 h , 将平均信道增益建模为 $\bar{h} = A_d \times \left(\frac{3 \times 10^8}{4\pi \cdot f_c \cdot d} \right)^{d_e}$, 其中: $A_d=4.11$, 载波频率 $f_c=915$ MHz, d 为 MDs 距离, 路径损耗指数 $d_e=3$ 。无线信道增益用瑞利衰落模型表示为 $h_{\text{up}}=0.06 \bar{h}$ 。经典上行链路—下行链路信道模型中^[27], 下行链路信道与上行链路信道相关, 其系数设为 0.7。

为了评估 PTRORA 算法, 在相同实验环境、任务实例下将 PTRORA 算法与全本地计算(All Local, AL)、依赖任务卸载与资源分配(Dependent Task Offloading and Resource Allocation, DTORA)、绝对成功卸载计算(Absolutely Successful Offloading, ASO)和随机卸载计算(Random Offloading, RO)算法在任务执行延迟和系统能耗方面进行了比较。各个算法特点如下:

1) AL: 所有任务都在本地执行计算。

2) DTORA: 基于 DQN 的依赖任务卸载与资源分配^[20]。

3) ASO: 不考虑卸载成功和失败概率的卸载方案。

4) RO: 任务随机选择卸载到 MEC 进行计算或本地计算。

图 3 展示了在不同卸载成功率 p 下, 总奖励值的变化趋势。结果表明, 卸载成功率对任务卸载决策具有显著影响。当卸载成功率较低时, 由于传输失败的概率较高, 系统更倾向于选择本地计算, 以降低因卸

载失败导致任务重试或超时的风险，从而减少整体惩罚。相反，随着卸载成功率的提升，卸载计算能够更稳定地减少任务执行时延，提高资源利用率，从而提升系统获得更高总奖励值的可能性。

图 4 展示了在不同带宽条件下系统总奖励值的变化趋势。结果表明，带宽对卸载决策具有一定影响。在相同卸载成功率下，随着带宽的增加，传输时延显著降低，从而提升系统奖励值。然而，当带宽超过某一阈值时，传输时延的变化趋于平缓，卸载决策更多受到卸载成功率的制约。因此，在低带宽条件下，由于传输时延较长，系统倾向于选择本地计算，以降低因卸载失败带来的惩罚，避免任务时间超过截止时延。而当带宽达到较高水平时，系统将综合考虑本地计算与卸载计算的收益与风险，并基于前景理论进行决策评估，做出更合理的卸载决策。

图 5 展示了终端移动距离对系统决策的影响。结果表明，随着 MDs 远离边缘服务器，路径损耗加剧，信道增益逐渐降低，导致传输时延增加，从而影响卸载决策。当距离超过一定阈值时，网络带宽显著下降，此时卸载计算的时延和成本可能高于本地计算。在前景理论的视角下，系统倾向于采取更保守的本地计算策略，以降低潜在风险并减少因卸载失败带来的惩罚。

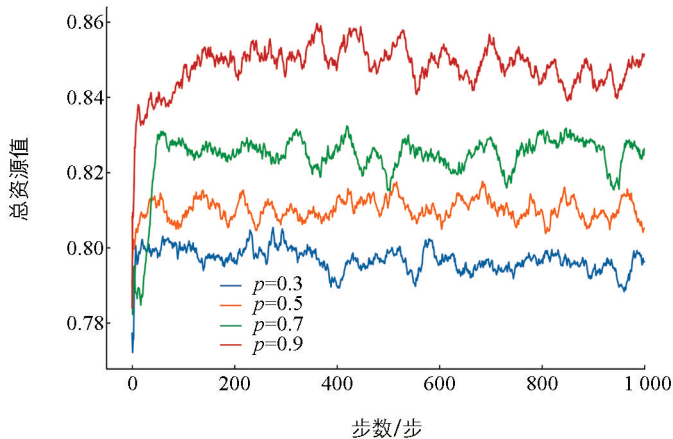


图 3 总奖励随卸载成功率的变化趋势

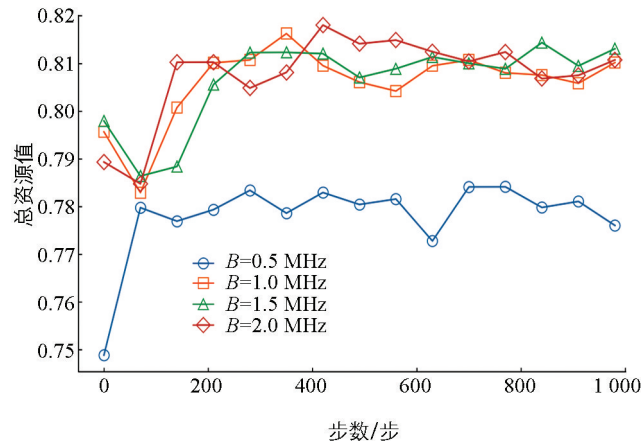


图 4 不同带宽条件下系统总奖励趋势

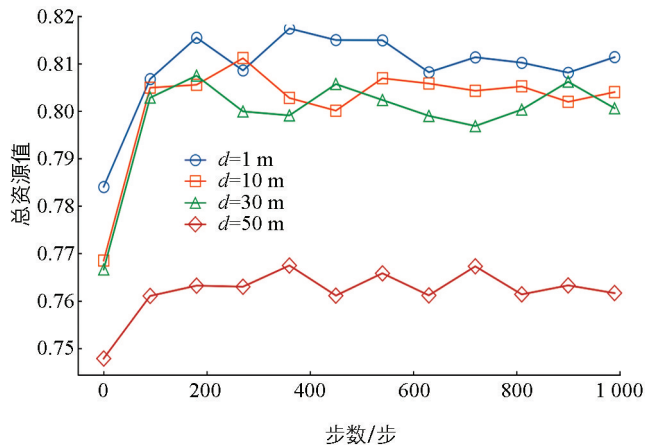


图 5 路径衰减对系统奖励值影响

图 6 展示了 DTORA 和 PTRORA 算法在完全独立决策和完全依赖任务决策两种不同环境下的累积奖励趋势。在这两种环境中，两种算法的奖励趋势表现出高度相似性，表明它们在不同环境中做出了可比较的决策。

图 7 展示了混合环境中 DTORA 和 PTRORA 的总奖励值趋势变化。结果表明，DTORA 的奖励值约为 0.8，而 PTRORA 的奖励值约为 0.82。这表明在混合环境下，尽管 DTORA 在全独立和全依赖任务场景中与 PTRORA 的决策表现相似，但在包含不同依赖关系的混合任务环境中，其决策效果略显不足。PTRORA 由于在卸载决策过程中能够更精准地权衡任务间的依赖关系与资源分配策略，因此在混合环境下展现出更强的适应性，从而获得更高的系统奖励值。说明针对复杂任务依赖结构的优化机制在提升整体性能方面具有重要作用，进一步验证了前景理论驱动的卸载策略在动态环境中的有效性。

图 8 展示了不同方法的总延迟趋势比较。AL 计算受限于自身计算能力，其总时延最高。这是由于所有任务均需在本地图理处理，缺乏并行计算能力，从而导致任务执行时间大幅增加。随机卸载由于具有不

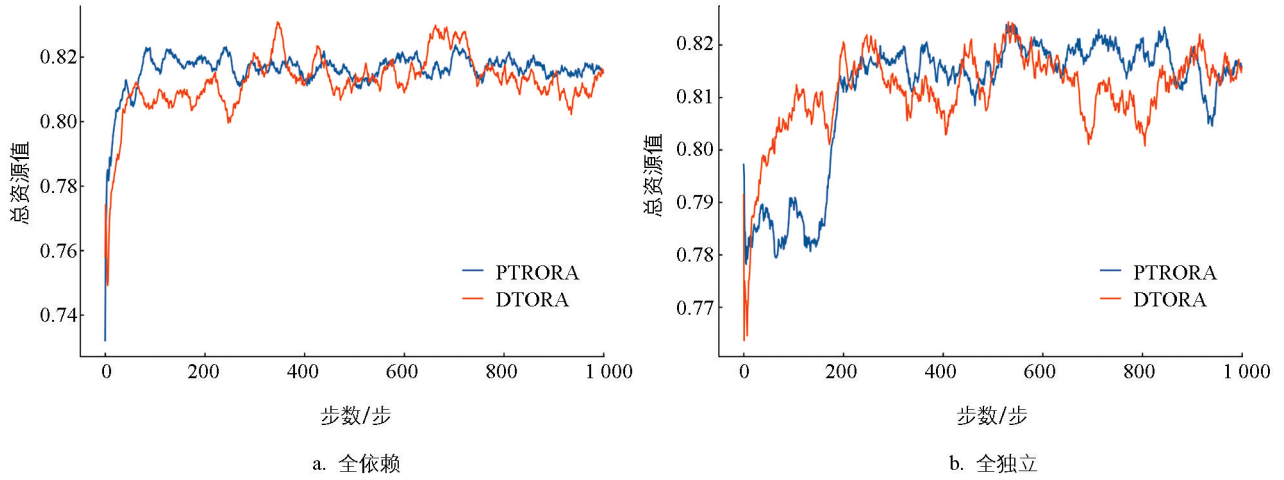


图 6 全依赖和全独立任务奖励值趋势

稳定性,在此不做具体对比。混合环境下,DTORA 由于未能充分优化任务卸载策略,导致其总时延高于 PTRORA。相比之下,PTRORA 结合前景理论优化决策,使得系统能够做出更为理性的决策,并根据任务优先级合理分配计算资源。因此,PTRORA 在降低系统总时延方面表现更优。值得注意的是,PTRORA 的最终延迟趋势接近绝对成功条件下的理想时延,这表明本文算法可以在不确定网络环境下做出有效的卸载决策。

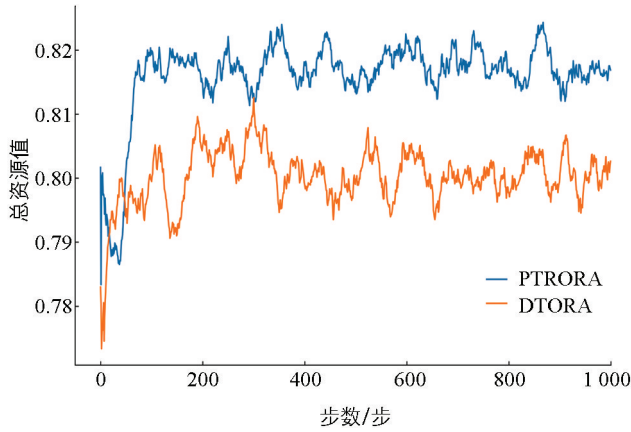


图 7 混合环境奖励值对比

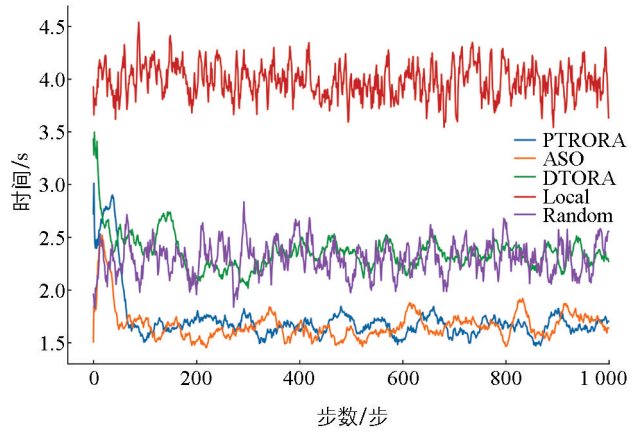


图 8 总时延趋势对比

图 9 展示了混合环境下 5 种方法的能耗对比情况。其中 AL 计算能耗最高,这主要是由于本地计算受限于设备计算能力,导致任务执行时间延长,从而增加了总能耗。RO 计算紧随其后。ASO 将全部任务卸载至 MEC,显著降低了 MDs 任务执行时的能耗,因此其能耗最低。在混合任务环境中,DTORA 由于未能有效优化卸载策略,导致其能耗高于 PTRORA。相比之下,PTRORA 通过前景理论评估卸载的风险与收益,合理分配 MEC 的计算资源,从而减少了总能耗。实验结果显示,与 DTORA、AL 和 RO 相比,PTRORA 的能耗分别降低了 27.3%、68.0%和 36.0%。



图 9 能耗对比

5 结语

本文提出了一种基于前景理论的混合任务风险卸载与资源分配算法。针对任务卸载失败带来的挑战以及异构任务环境的复杂性,将前景理论引入卸载决策过程,以优化风险与收益的权衡。通过 A3C 深度强化学习算法,依据任务的最高可容忍时延对任务进行分类,并据此分配计算资源,从而提升任务执行效率和用户体验。仿真结果表明,与现有研究成果相比,所提算法有效减少了卸载失败的影响,同时显著降低了任务时延与能耗,验证了其在动态计算环境下的优越性。未来的研究方向将进一步聚焦信道状态的时变性和终端移动性对卸载决策的影响,针对 MEC 网络中无线信道相关的优化问题进行拓展。

参考文献:

- [1] LAI P, HE Q, GRUNDY J, et al. Cost-Effective App User Allocation in an Edge Computing Environment [J]. *IEEE Transactions on Cloud Computing*, 2022, 10(3): 1701-1713.
- [2] SHARGHIVAND N, DERAKHSHAN F, MASHAYEKHY L, et al. An Edge Computing Matching Framework with Guaranteed Quality of Service [J]. *IEEE Transactions on Cloud Computing*, 2022, 10(3): 1557-1570.
- [3] WOLF T, DEBUT L, SANH V, et al. Transformers: State-of-the-Art Natural Language Processing [C] // *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online. Stroudsburg: ACL, 2020: 38-45.
- [4] GERLA M, LEE E K, PAU G, et al. Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds [C] // *2014 IEEE World Forum on Internet of Things (WF-IoT)*. New York: IEEE Press, 2014: 241-246.
- [5] ZHANG Y, HE X Y, XING J, et al. Balanced Offloading of Multiple Task Types in Mobile Edge Computing [C] // *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*. New York: IEEE Press, 2023: 2746-2747.
- [6] CHEN X, JIAO L, LI W Z, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing [J]. *IEEE/ACM Transactions on Networking*, 2016, 24(5): 2795-2808.
- [7] TALEB T, SAMDANIS K, MADA B, et al. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration [J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1657-1681.
- [8] DU J B, CHENG W J, LU G Y, et al. Resource Pricing and Allocation in MEC Enabled Blockchain Systems: An A3C Deep Reinforcement Learning Approach [J]. *IEEE Transactions on Network Science and Engineering*, 2022, 9(1): 33-44.
- [9] ZHANG K, CAO J Y, LENG S P, et al. Mining Task Offloading in Mobile Edge Computing Empowered Blockchain [C] // *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*. New York: IEEE Press, 2019: 234-239.
- [10] SCHMEIDLER D. Subjective Probability and Expected Utility without Additivity [J]. *Econometrica*, 1989, 57(3): 571.
- [11] DONG S, TANG J X, ABBAS K, et al. Task Offloading Strategies for Mobile Edge Computing: A Survey [J]. *Computer Networks*, 2024, 254: 110791.
- [12] ISLAM A, DEBNATH A, GHOSE M, et al. A Survey on Task Offloading in Multi-Access Edge Computing [J]. *Journal of Systems Architecture*, 2021, 118: 102225.
- [13] BEHERA S R, PANIGRAHI N, BHOI S K, et al. A Distributed Fuzzy Optimal Decision Making Strategy for Task Offloading in Edge Computing Environment [J]. *IEEE Access*, 2023, 11: 33189-33204.
- [14] LV Z H, QIAO L. Optimization of Collaborative Resource Allocation for Mobile Edge Computing [J]. *Computer Communications*, 2020, 161: 19-27.

- [15] FENG C, HAN P C, ZHANG X, et al. Computation Offloading in Mobile Edge Computing Networks: A Survey [J]. *Journal of Network and Computer Applications*, 2022, 202: 103366.
- [16] YANG J M, SHAH A A, PEZAROS D. A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks [J]. *Electronics*, 2023, 12(17): 3548.
- [17] EBRAHIM M A, EBRAHIM G A, MOHAMED H K, et al. A Deep Learning Approach for Task Offloading in Multi-UAV Aided Mobile Edge Computing [J]. *IEEE Access*, 2022, 10: 101716-101731.
- [18] CHAKRABARTI A, GUÉRIN R, LU C Y, et al. Real-Time Edge Classification: Optimal Offloading under Token Bucket Constraints [C] // 2021 IEEE/ACM Symposium on Edge Computing (SEC). New York: IEEE Press, 2021: 41-54.
- [19] 李强, 仪晋辉, 杜婷婷, 等. 移动边缘计算中基于 A3C 的依赖任务卸载与资源分配 [J]. *计算机工程*, 2023, 49(6): 42-52.
- [20] TANG M, WONG V W S. Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems [J]. *IEEE Transactions on Mobile Computing*, 2022, 21(6): 1985-1997.
- [21] 任潇扬, 于秀兰. Stackelberg 博弈的 MEC 资源分配策略 [J]. *重庆邮电大学学报(自然科学版)*, 2024, 36(3): 469-477.
- [22] 杨金松, 孙三山, 刘莉, 等. MEC 架构下基于 DDPG 的车联网任务卸载和资源分配 [J]. *重庆邮电大学学报(自然科学版)*, 2024, 36(2): 259-267.
- [23] TANG C G, YAN G, WU H M, et al. Computation Offloading and Resource Allocation in Failure-Aware Vehicular Edge Computing [J]. *IEEE Transactions on Consumer Electronics*, 2024, 70(1): 1877-1888.
- [24] KAHNEMAN D, TVERSKY A. Prospect Theory: An Analysis of Decision under Risk [M] // MACLEAN L C, ZIEMBA W T. *Handbook of the Fundamentals of Financial Decision Making: Part I*. Hackensack: World Scientific Publishing, 2013: 99-127.
- [25] 3GPP. Technical Specification Group Services and System Aspects [EB/OL]. (2020-11-23) [2025-01-01]. <https://www.3gpp.org/technologies/5g/23-888>.
- [26] ZENG S, HUANG Y, LIU X. Privacy-Preserving Communication for Vanets with Conditionally Anonymous Ring Signature [J]. *International Journal of Network Security*, 2015, 17(2): 135-141.
- [27] PERAHIA E, COX D C. Shadow Fading Correlation between Uplink and Downlink [C] // IEEE VTS 53rd Vehicular Technology Conference. New York: IEEE Press, 2002: 308-312.

责任编辑 张宥
柳剑

