

基于多智能体协作的移动边缘计算任务卸载及调度策略

石家琪^{1,2}, 张旭龙^{1*}, 瞿晓阳¹, 倪晓俊³, 田晖⁴, 王健宗¹

1. 平安科技(深圳)有限公司, 深圳 518000;

2. 中国科学技术大学, 合肥 230000;

3. 中国科学院计算技术研究所, 北京 100190;

4. 华侨大学, 泉州 362021

摘要

随着移动互联网和物联网的普及, 用户终端应用对网络传输速率和低时延的需求日益增长。为解决用户设备计算能力受限的难题, 我们提出了一套全新的基于A2C算法的多智能体任务卸载及资源分配方案。该策略利用用户终端、多个接入点和边缘服务器, 构建了一个去蜂窝大规模多输入多输出(MIMO)无线网络。核心思想是将计算任务从用户终端转移到附近的边缘服务器上执行, 从而显著降低用户时延和能耗。该方案将任务卸载问题分解为两个子问题, 并采用强化学习中的长期奖励概念来建模。其中, 一个子问题采用凸优化求解, 另一个则使用深度强化学习算法A2C解决, 共同形成了任务卸载与传输功率分配、以及计算节点资源分配的联合策略。仿真结果显示, 该基于深度强化学习的联合策略在用户时延和能耗方面优于仅考虑凸优化的策略。此外, 研究还分析了用户数量、接入点集群配置及规模等因素对系统性能的影响。该研究为移动边缘计算环境中的高效任务卸载和资源管理提供了有价值的解决方案。

关键词

边缘计算; 任务卸载; 深度强化学习

中图分类号: TP391

文献标志码: A

doi: 10.11959/j.issn.2096-0271.2026061

Multi-Agent Cooperative Task Offloading and Scheduling Strategy for Mobile Edge Computing

Shi Jiaqi^{1,2}, Zhang Xulong^{1*}, Qu Xiaoyang¹, Ni Xiaojun³, Tian Hui⁴, Wang Jianzong¹

1. Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China;

2. University of Science and Technology of China, Hefei, China;

3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China;

4. Huaqiao University, Quanzhou, China

Abstract

With the proliferation of mobile Internet and the Internet of Things (IoT), user-side applications demand ever higher throughput and lower latency. To address the limited computing capability of user devices, we propose a novel multi-agent task offloading and resource allocation scheme based on the Advantage Actor-Critic (A2C) algorithm. The

strategy leverages user equipment (UEs), multiple access points (APs), and edge servers (ESs) to form a cell-free massive MIMO wireless network. The core idea is to offload computation tasks from UEs to nearby edge servers, thereby significantly reducing user-perceived latency and energy consumption. We decompose the offloading problem into two subproblems and model it using the long-term reward framework from reinforcement learning: one subproblem is solved via convex optimization, while the other is addressed with the A2C deep reinforcement learning algorithm. Together they yield a joint policy for task offloading, transmit-power allocation, and compute-node resource allocation. Simulation results show that this DRL-based joint strategy outperforms approaches that rely solely on convex optimization in terms of both latency and energy consumption. In addition, we analyze how factors such as the number of users, AP clustering configurations, and network scale affect system performance. This study offers a valuable solution for efficient task offloading and resource management in mobile edge computing environments.

Key words

Edge computing, task offloading, deep reinforcement learning

0 引言

随着物联网(IoT)与移动网络的快速发展,需要增加边缘计算的计算资源以支持低时延通信和高密度计算的需求。面向无人驾驶场景,车辆平台通常集成多种感知硬件,如激光雷达、摄像机、超声阵列与毫米波雷达等。上述设备以高频率采集并输出庞大的数据流,用于完成环境建模、障碍物检出、路牌与红绿灯等交通要素的感知与判别。随着传感器数量和分辨率的增加,产生的数据量也随之增长,这种增长趋势对数据存储、传输、处理和分析等方面提出了巨大挑战。与此同时,这些数据需要及时传输到车载计算平台进行处理和分析。由于数据量巨大,对于实时处理要求很高,因此需要保证数据传输的时延尽可能地短。多接入边缘计算(MEC)作为移动网络的技术之一,通过提供靠近用户设备的丰富计算资源,满足了自动驾驶领域对超低时延和高密度计算任务的要求。

MEC允许用户设备将超出设备自身计

算能力的大量计算任务卸载到附近的MEC服务器,借助服务器提供的较高算力完成设备的密集计算任务。但是,MEC服务器的计算资源相比云中心还是有限的,因此当出现过多用户设备同时要求卸载任务到MEC服务器时,可能会造成部分计算任务无法在约定时间内完成以及网络和服务器拥塞,从而降低用户设备的体验与服务质量。因此,高效的任务卸载策略对于降低系统总体时延和能耗意义重大。

与此同时,将用户设备的任务传输到MEC服务器需要借助无线通信网络来实现。传统的蜂窝大规模多输入多输出(Cell Massive MIMO)网络需要在移动终端附近部署基站,借助基站中的无线设备将计算数据卸载至MEC服务器,因此MEC服务器与无线通信网络的传输性能紧密联系。然而在传统的蜂窝MIMO网络中,位于多个基站交界处的用户设备会受到多个区域间的信号干扰,从而导致无线传输速率降低。处于该位置的用户设备只能将计算任务在本地处理,这将降低有时延要求的用户设备体验并严重威胁到处于移动环境中(如智能驾驶环境)乘客的安全。去蜂窝大

规模 MIMO 无线通信网络为解决传统蜂窝 MIMO 网络存在的信号衰减问题，通过部署大量接入点 (APs) 替代蜂窝网络中的基站，减少信号间的干扰，提高了信号传输速率和可靠性。为了满足自动驾驶领域对于海量数据的低处理时延要求以及减少信号干扰的安全性要求，本文构建基于去蜂窝大规模 MIMO 无线通信网络的边缘计算框架，综合考虑终端任务数据量、边与端计算资源以及通信模型，探索基于深度学习的任务卸载决策与资源分配的联合优化策略。

1 国内外研究现状

1.1 边缘计算场景中的任务卸载研究

针对卸载请求的决策和资源管理可以有效降低 MEC 系统的整体时延与能耗，满足 5G 场景下用户设备对于数据处理的要求。在以前的研究中，一些任务卸载算法假定 MEC 服务器上有无限的计算资源用于卸载操作[1]。然而，爆炸式增长的计算密集型任务和不平衡的请求分布正在挑战承载网络中 MEC 服务器的计算能力。有限的 MEC 资源要求卸载策略考虑长期回报，并实现多个 MEC 服务器之间的协作。周杰钦提出基于动态协作卸载任务并且混合智能算法的技术创新来同时解决无人机部署的离散优化问题和任务卸载的连续决策问题[2]。除此以外，还有分支与边界法、启发式算法和基于强化学习的算法。解决物联网系统计算卸载问题最常用的强化学习方法之一是 Q 学习，例如 Chen 等人提出的基于强化学习和李雅普诺夫优化理论的卸载算法，用来解决云-边-端网络中的通信和计算资源分配问题[3]。此外，Shen 等人提出了基于 Q 学习的进化博弈算

法[4]，研究了在动态多用户设备环境下的计算卸载问题，并且考虑了无线信道在多个设备使用时可能会产生的干扰现象。尽管文中考虑了基于时间间隔的通信，但并没有考虑其他方面的动态情景，并且场景简单，只有一个基站和一个边缘服务器。

除了应用 Q 学习解决卸载问题的思路外，深度强化学习技术也可以被应用在边缘计算任务卸载问题中。Khani 等人应用了深度强化学习中的 DQN 算法，他们的方法包括两个步骤：首先，根据卸载任务的优先级对设备进行集中集群划分；其次，基于分布式 DQN 的计算卸载算法优化选择每个设备要卸载的任务[5]。这并不是解决物联网场景中计算卸载问题的唯一基于 DQN 的方案，在已发表的文章中也有多个实例。除了应用 DQN 算法外，还有些研究使用基于 DQN 的衍生框架来设计卸载策略。如 Bolourian 等人提出了一个基于 DROO 的在线卸载框架，该框架可以在无线通信随时间变化的 MEC 系统中选择最优的卸载和资源分配策略[6]。此外，为了进一步降低复杂性，文献中还提出了一种自适应程序来动态调整算法的参数[6]。然而在这篇文献中也只考虑了一个相对简单的场景，只有一个基站和边缘服务器，不具备模型通用性。Huang 考虑了一个具有多个用户设备和多个边缘服务器的 MEC 系统[7]。面向上述场景，作者提出一套基于 DDPG 的方法；由于引入了 Dirichlet 分布，命名为 Dirichlet-DDPG。近期，Ren 等人采用“互信息增强”的多智能体强化学习 (MARL) 来做协作式任务卸载决策[8]。Tang 等人面向车联网 V2V/V2I 的图强化学习 (Graph RL) 任务卸载方法，目标最小化端到端时延[9]。Hao 虽然聚焦边缘调度而非直接“卸载”，但其分层 + 安全多智能体 DRL 框架与卸载决策问题高

度同构,常作为卸载策略设计的参考[10]。

1.2 去蜂窝大规模 MIMO 通信网络研究

自从引入多输入多输出(MIMO)技术以来,在无线通信中使用多天线一直是抵消无线信道不利影响的一种方法,例如在时空编码中,或在多路复用技术中用于提高比特率。不过,这些方法大多针对单用户设备应用。在为多用户设备提供高比特率的系统中,必须使用另一种方法。大规模多输入多输出(Massive MIMO)使用大量天线为同一频段的大量用户设备同时提供高比特率,可采用波束成形技术来实现这一目的。实现这一点的代价是需要了解信道状态信息(Channel State Information, CSI)。

去蜂窝大规模 MIMO 是一种分布式的大规模 MIMO,所有天线都在一个范围广泛的区域,没有蜂窝边界的概念,也就没有用户设备边界的概念。虽然这个概念对于提高边界用户设备的通信效率有帮助,但是在具体实时过程中却面临连接天线和中心单元的回传链路容量有限的问题。为了解决这一问题,赵迪等人针对为不同优先级用户设备提供不同服务质量的目标,建立长期功率约束下的下行链路加权功率控制优化问题模型[11]。将该问题转化为凸优化问题,求解得出一组优化的下行链路功率控制系数。Mobini 等人假设天线对信道矩阵有不准确的估计,而用户设备对有效信道有即时的了解,推导并解决了功率权重分配和量化的联合优化问题[12]。

同时,近来有学者尝试把 MEC 与分布式网络 MIMO 相整合,实现二者的协同。Interdonato 在基于去蜂窝大规模 MIMO 系统框架下,考虑具有 MEC 服务器的接入点和具有云计算能力的中央服务器,

利用随机几何和队列理论推导出通信概率后,提出了目标计算时延下的成功边缘计算概率(SRCP)[13]。结果表明,就 SECP 而言,卸载到 CS 的最佳概率随着接入点覆盖率的增加而降低。Abdulla 利用海鸥优化深度学习算法(TORA-DLSGO)设计了一种新的基于物联网的 MEC 任务卸载和资源分配算法[14]。TORA-DLSGO 面向 MEC 侧的资源编排与调度,能够产出最优卸载方案,从而把系统整体开销压至最低[14]。

2 背景知识

2.1 去蜂窝大规模 MIMO 通信网络

由于用户设备对高吞吐量、低时延、可靠性以及即使连接需求的快速增长,5G 无线通信网络已经日趋成熟。同时,来自工业界和学术界的目光聚焦在了 6G 无线通信网络。如图 1 所示,在去蜂窝大规模 MIMO 通信系统中,覆盖区域内分布着大量的接入点(AP),每个接入点与用户设备间配备了大量的天线元件。蜂窝(Cell)的概念,即特定区域内的所有用户设备(UE)仅由特定接入点(AP)服务,不复存在。每个用户设备(UE)由所有接入点(AP)以时分双工(TDD)模式使用相同的时频资源联合服务。由于天线和接入点在地理上是分散的,因此与集中式系统相比这种系统受阴影衰落的影响更小。并且 UE 到最近的 AP 的平均距离更小,因此数据传输速率也会有所提高。但是,去蜂窝大规模 MIMO 网络需要通过光纤或者无线回传链路将 AP 与服务器连接起来。

在 AP 处使用很多天线时,可实现有利传播与信道硬化。其中,有利传播指在某些情况下,无线信号在传输过程中会经历

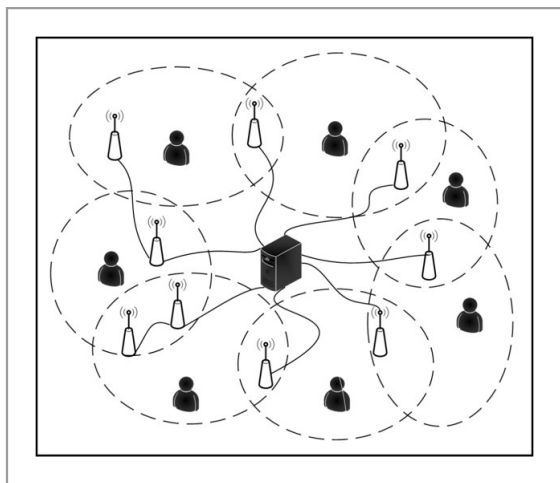


图1 去蜂窝 MIMO 网络架构示意图

一些环境或信道条件的优势，使得信号的接收端能够获得较高质量的信号。这种优势可能是由于信号在传播过程中遇到了反射、折射、散射等现象，使得接收信号的能量增强或多个路径的信号合成效应增强。信道硬化指在大规模天线系统(Massive MIMO)等多天线通信系统中，由于信号在多个天线间的独立传播路径，导致接收信号受到的干扰和衰减效应相对较小。因此，即使在信道条件变化或随机性较大的情况下，接收信号的表现也相对稳定，其干扰和衰减效应被平均或抵消掉了，从而使得信道的特性变得更加可预测和稳定。

去蜂窝大规模 MIMO 网络中会根据用户设备(UE)的地理位置和网络环境动态地分配一个 AP 集群。当用户设备(UE)需要卸载任务时，用户设备通过上行链路将计算任务上传至为该用户设备分配的 AP 集群中，AP 集群内的所有 AP 都会提供协作上行传输服务。当 AP 集群接收到用户设备数据后，再通过回传链路将信号上传至 MEC 服务器中。在后续的网络建模部分，上行链路的建模部分同时参考了 Buzzi[15] 和 Zhao[16] 的方法，回传链路的建模部分参考了 Femenias 等人的方法[17]。

由于在去蜂窝大规模 MIMO 系统中，不同的用户设备同时由相同的时频资源服务，控制用户设备间的干扰很重要。功率分配在控制这种干扰并优化上行链路性能方面发挥着十分重要的作用。尽管许多现有的功率分配算法在仿真和理论分析中表现出优异的性能，但它们在现实系统中的实现面临着严重的障碍。特别是，过高的计算复杂度使得在功率分配算法在落地应用时成为最主要的问题。由于信道的状态随时间变化，因此功率分配应与该状态保持一致。目前实现这一目标的方法通常以迭代方式找到接近最优的解决方案，但代价是相当大的计算复杂性，可能使功率分配算法的计算时间过长，从而使得用户设备的低时延需求得不到满足。例如，常见的加权最小均方误差(WMMSE)算法在每次迭代中都需要复杂的操作。而强化学习算法可以在智能体与环境交互的过程中，不断优化控制策略，实时给出合适的功率选择。

因此，在设计基于深度强化学习的任务卸载策略的同时也考虑了上行链路中的传输功率控制，设置用户设备最大传输功率，对用户设备传输功率进行控制。为了降低算法复杂度，将连续空间的功率控制问题转化为离散空间的功率控制。

2.2 强化学习

强化学习的过程即为智能体(Agent)与环境的交互过程，其中智能体下一个时间点所采取的动作，并将该动作与环境交互，获得交互后的下一个状态以及该动作所得收益。状态(state)用于描述智能体在该环境中的处境，可以用 s 表示。描述所有状态的集合称为状态空间，用 S 表示。假设有 n 个状态 $s_1, s_2, \dots, s_j$ ，则状态空间 S 可表示为

$S=s_1, s_2, \dots, s_j$ 。动作(Action)表示智能体在某个状态下可以采取的动作或决策, 通常用符号 a 表示。动作空间 A 是所有可能行动的集合。在强化学习中, 智能体根据当前状态选择一个行动, 作为对环境的响应。动作可以是离散的(如移动一个棋子)或连续的(如调节机器人的速度), 同样取决于问题的特性。对于任务卸载问题, 可以将其定义为动作离散的问题。策略是一个映射函数, 它将状态映射到行动的概率分布。在状态 s 下, 策略 π 给出了智能体选择行动 a 的概率, 通常表示为 $\pi(a|s)$ 。策略可以是确定性的(即对于每个状态只选择一个行动)或随机性的(即对于每个状态可能选择不同的行动)。收益是智能体在执行某个行动后从环境中获得的数值反馈, 通常用符号 R 表示。在状态 s 下采取行动 a 后获得的即时收益为 $R(s, a)$ 。在某一状态下执行一个动作后, 代理会从环境中获得一个奖励(用 r 表示)作为反馈。奖励是状态 s 和行动 a 的函数, 它的值可以是正或负的实数或零。不同的奖励对代理最终学习的政策有不同的影响。最终学习的策略产生不同的影响。一般来说, 如果奖励为正数, 则鼓励代理采取相应的行动。采取相应的行动。对于负奖励, 需要阻止代理采取该行动。

满足马尔可夫特性的强化学习任务称为马尔可夫决策过程(MDP)。如果状态空间和行动空间都是有限的, 则称为有限马尔可夫决策过程(Finite MDP)。一个特定的有限 MDP 是由其状态和行动集以及环境的一步动态变化决定的。给定任何状态和行动 s 和 a , 下一个状态和奖励的每一对可能的概率 s', r 表示为:

$$p(s', r|s, a) Pr = S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a$$

可以根据上述公式计算当前状态-动作组的预期收益, 计算公式如下:

$$r(s, a) = E[R_{t+1} | S_t = s, A_t = a] = \sum_{r \in R} r \sum_{s' \in I} p(s', r|s, a)$$

基于预期收益 $r(s, a)$, 可将状态转移概率公式转换为:

$$p(s'|s, a) Pr = S_{t+1} = s' | S_t = s, A_t = a = \sum_{r \in R} p(s', r|s, a)$$

同时, 三元组 (s, a, s') 的预期收益可表示为:

$$r(s, a, s') = E[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] = \frac{\sum_{r \in R} r p(s', r|s, a)}{p(s'|s, a)}$$

几乎所有的强化学习算法都涉及对状态(或状态动作对)的价值函数进行估计, 以估算代理处于给定状态的好处(或在给定状态下执行给定动作的好处)。策略 π 可视为一种映射: 对任一状态 $s \in S$ 及其可选动作 $a \in A(s)$, 给出在 s 处选择 a 的概率 $\pi(a|s)$ 。在遵循该策略时, 状态价值函数记作 $v_\pi(s)$, 表示从状态 s 出发按 π 行动所能获得的预期回报。对于任意马尔可夫决策过程(MDP), 可将 $v_\pi(s)$ 定义为:

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right]$$

记 $E_\pi[\cdot]$ 为在策略 π 下, 时刻 t 的随机变量期望; 相应地, v_π 称为该策略对应的状态价值函数。

同样, 定义动作价值 $q_\pi(s, a)$: 表示从状态 s 出发、在 π 约束下先选择动作 a 时所能获得的期望回报。

$$q_\pi(s, a) = E_\pi[G_t | S_t = a, A_t = a] = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right]$$

可将前文定义的 v_π 与 q_π 通过交互数据

来近似评估：令智能体按策略 π 行动，记录每次访问到的状态后的实际回报，并取其样本均值，作为对应价值的估计量。那么，当遇到该状态的次数接近无穷大时，平均值就会趋近于状态值 $v_{\pi(s)}$ 。如果在一个状态下采取的每一个行动都保留不同的平均值，那么这些平均值同样会收敛到行动值 $q_{\pi}(s, a)$ 。可以建立上述状态价值函数 $v_{\pi}(s)$ 和动作价值函数 $q_{\pi}(s, a)$ 之间的关系：

$$v_{\pi}(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_{\pi}(s')]$$

其中， γ 表示折扣因子，用于衡量未来收益的重要性。

2.3 A2C算法

强化学习的常见范式可大致分为两路：一类以估计价值函数为核心（value-based），另一类直接优化行为策略（policy-based）。基于值的算法试图找到近似最佳值函数，而基于策略的算法试图直接找到最佳策略，而无需Q值作为中间步骤。Q-Learning是最经典的基于值的方法。它通过学习状态-动作值函数 $Q(s, a)$ ，直接估计在给定状态 s 下选择动作 a 所能获得的期望回报。DQN 将深度神经网络纳入 Q-Learning 框架，用以应对维度庞大的状态表示与估值问题。PPO 属于当下应用广泛的策略优化（policy-based）强化学习方法，通过裁剪和信任域约束平衡探索和利用，限制策略更新步长以保证训练稳定性。而 Actor-critic 框架试图将两类主流算法融合，其思想是将模型分为两部分：基于状态计算动作和用于估计动作的Q值。Actor 网络将状态作为输入并输出最佳动作，通过控制智能体的行动来学习最佳策略。Critic 网络通过计算值函数评估动作。让两个网络相互竞争的思想在机器学习领

域广泛流行，如生成对抗式网络(GAN)。

AC(Actor-Critic) 和 A2C(Advantage Actor-Critic) 都是强化学习中常用的算法，属于 Actor-Critic 架构的一部分。它们在许多方面具有相似之处，但也存在一些关键的差异。标准的 AC 算法使用状态值函数来评估当前状态的价值。更新策略时，AC 使用从 Critic 估计的状态值来计算策略梯度。AC 的训练效率较低，因为它在每个时间步后进行更新，可能导致高方差和不稳定的训练过程。为了解决该问题，A2C 算法框架引入一个基线的方式，即在计算期望的时候用累计奖励减去基线，该方法可以让梯度减小，因此梯度下降的趋势也更加平缓，从而使训练过程更加稳定。将 $A_{\pi}(s, a)$ 定义为优势函数，用于度量在状态 s 下执行动作 a 相对于该状态基准选择所带来的相对增益。它定义为动作价值函数 $Q_{\pi}(s, a)$ 和状态价值函数 $V^*(s)$ 之间的差异：

$$A_{\pi}(s, a) = Q_{\pi}(s, a) - V_{\pi}(s, a)$$

A2C 使用策略梯度方法来更新演员的参数 θ ，目标是最大化预期回报。策略梯度的更新规则可以表示为：

$$\nabla_{\theta} J(\theta) = E_{\pi} [\nabla_{\theta} \log \pi(a|s; \theta) A_{\pi}(s, a)]$$

其中， $J(\theta)$ 是预期回报， E_{π} 表示在策略 π 下的期望。更新参数 θ 的一种常见方法是使用梯度上升法。

3 基于强化学习的边端协同的 MEC 任务卸载策略

3.1 系统模型

基于去蜂窝大规模 MIMO 通信网络的边缘计算任务卸载模型如图 2 所示，该架构去除云-边-端结构中的云中心计算部分，考虑用户设备端与边缘节点之间的任

务卸载与资源分配。设在该网络中有 M 个用户设备和 N 个接入点 AP，根据用户设备的地理位置和网络环境动态地分配一个 AP 集群。当用户设备需要卸载任务时，用户设备通过上行链路将计算任务上传至为该用户设备分配的 AP 集群中，AP 集群内的

所有 AP 都会提供协作上行传输服务。当 AP 集群接收到用户设备数据后，再通过回传链路将信号上传至 MEC 服务器中。通过卸载策略，MEC 服务器可以决定通过有线接口将工作负载重定向到其他已连接的服务器。

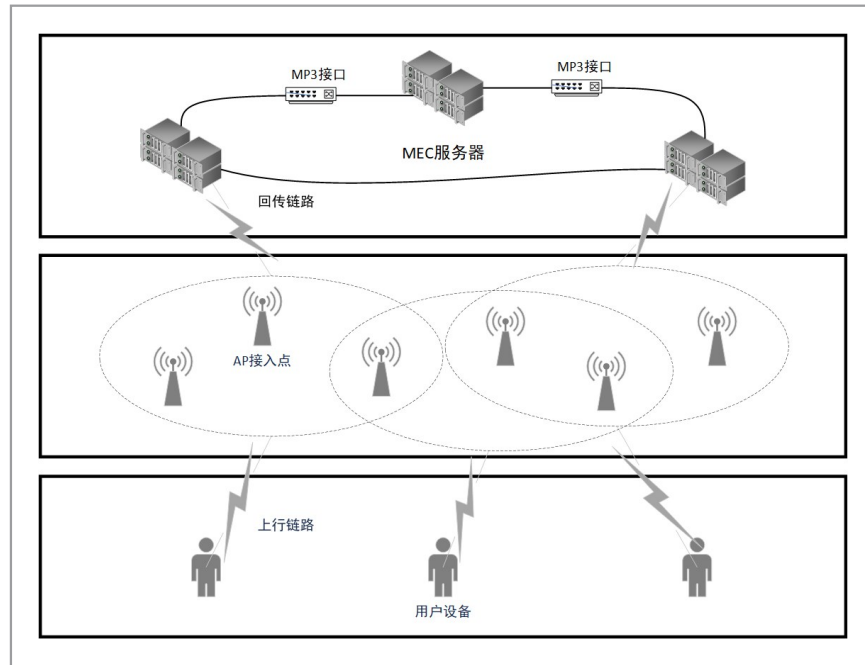


图2 边缘计算任务卸载架构示意图

将第 i 个用户设备表示为 UE_i ，第 j 个接入点表示为 AP_j ，第 k 个 MEC 服务器表示为 MEC_k 。设单位时间为 Δt ，整个执行时间由连续的单位时间组成，表示为 T 。在第 t 个单位时间，用户设备 UE_i 产生一个计算任务 $Task_i(t) = \{C_i(t), D_i(t)\}$ ，其中 $C_i(t)$ 表示执行任务 i 所需时钟周期数， $D_i(t)$ 表示该任务数据大小，单位为比特。假设在每个单位时间内信道状态信息(CSI)保持不变。假设每个任务都有一个最大的可容忍时延 ω ，终端可有两种执行路径：要么在本机直接计算任务，要么经接入点 (AP)

把作业转交到边缘计算 (MEC) 服务器处理。设以 CPU 周期频率衡量的本地计算能力为 f^{UE} ，MEC 服务器的计算能力为 f^{MEC} ，单位 CPU 周期/秒。

3.2 通信模型-上行链路

在蜂窝大规模 MIMO 系统中有 M 个用户设备和 N 个接入点。每个接入点都配备有 u 个天线，每个用户设备均为单天线设备[15]。将 t 时刻用户设备 UE_i 和接入点 AP_j 之间的信道向量表示为 $g_{ij}^a(t)$ ：

$$g_{ij}^a(t) = \sqrt{\beta_{ij}} \cdot h_{ij}(t)$$

其中, 信道向量 $g_{ij}^a(t)$ 是大尺度信道衰落 $\beta_{ij}(t)$ 和小尺度衰落 h_{ij} 的乘积。假设小尺度衰落在每个时隙区间内都是静态的, 并且从一个时隙区间到下一个时隙区间之间独立变化。大规模衰落的变化要慢得多, 并且根据用户设备的移动性在几个间隔内保持不变。Ngo 等人将 $\beta_{ij}(t)$ 定义为[18]:

$$\beta_{ij}(t) = 10^{\frac{PL_{ij}(t)}{10}} \cdot 10^{\frac{\sigma^2 z_{ij}(t)}{10}}$$

其中, $PL_{ij}(t)$ 表示 t 时刻用户设备 UE_i 和接入点 AP_j 之间的路径损失(单位 dB), $10^{\frac{\sigma^2 z_{ij}(t)}{10}}$ 用于计算阴影衰落系数, σ^2 表示标准偏差, 其中 $z_{ij}(t) \sim N(0,1)$ 。采用三梯度模型来计算路径损耗 $PL_{ij}(t)$, 计算公式如下:

$$PL_{ij}(t) = \begin{cases} -L - 35 \cdot \log_{10}(d_{ij}(t)) & , d_{ij}(t) > d_1 \\ -L - 10 \cdot \log_{10}(d_1^{1.5} \cdot d_{ij}^2(t)) & , d_0 < d_{ij}(t) \leq d_1 \\ -L - 10 \cdot \log_{10}(d_1^{1.5} \cdot d_0^2) & , d_{ij}(t) < d_0 \end{cases}$$

其中, d_0 、 d_1 为分界点, L 满足以下公式:

$$L = 45.5 + 33.9 \cdot \log_{10}(f) - 13.82 \cdot \log_{10}(h_a) - [1.11 \cdot \log_{10}(f) - 0.7] \cdot h_u + 1.56 \cdot \log_{10}(f) \#$$

其中, f 表示载波频率(MHz), h_a 和 h_u 表示接入点和用户设备的天线高度。由于用户设备和接入点之间存在障碍物, 为了校正阴影衰落现象, 将 $z_{ij}(t)$ 分为两部分:

$$z_{ij}(t) = \sqrt{\delta} \cdot \dot{z}_j(t) + \sqrt{1-\delta} \cdot \dot{z}_i(t)$$

其中, $\dot{z}_j(t)$ 和 $\dot{z}_i(t)$ 按照高斯分布 $N(0,1)$, 并且 $0 \leq \delta \leq 1$ 为自定义参数, 实验中设置为 0.5。

每个用户设备都由一个特定的 AP 集群

提供服务, 将每个用户设备的 AP 集群大小定义为 S 。一个 AP 集群可以服务多个用户设备。本模型假设每个用户设备所对应的 AP 集群都是动态分布的, 其分配过程如下:

- 首先将用户设备 UE_i 与所有接入点的信道量降序排序。
- 然后选取前 S 个 AP 作为用户设备 i 的 AP 集群。

其中, S 为自定义变量, 用于控制 AP 集群的规模。将 t 时刻为用户设备 i 分配的 AP 集群称为 $\Phi_i(t)$ 。基于瑞利衰落信道模型 [19], 将从用户设备 i 向接入点集的上传数据的传输速率 r_i^a 定义为:

$$r_i^a = B^a \cdot \log_2 \left(1 + \frac{p_i \cdot \left[\sum_{j \in \Phi_i(t)} g_{ij}^a(t) \right]^2}{\sum_{j \in \Phi_i(t)} \sum_{l \neq i} p_l g_{lj}^a(t)^2 + S \cdot N_0^2} \right)$$

其中, B^a 表示上行信道带宽, 在实验中设置为固定参数, p_i 表示从用户设备 i 传输功率, N_0 表示高斯噪声。参考 Lin 的方法, 为了验证优化传输功率分配决策的必要性, 在模拟实验中将用户设备的传输功率进行离散化, 令 p_i^{max} 表示用户设备 i 的最大传输功率, 而 p_i 的值则基于当前策略从 $0, 0.2, \dots, p_i^{max}$ 中选择[20]。

3.3 通信模型-回传链路

从接入点 AP 到 MEC 服务器的传输信道具有较大的可用带宽和高波束成形增益, 如此高的带宽可以处理 MEC 回传链路内的大量数据传输。在通信系统中, 信号传输路径为直线(LoS)的情况下, 信号可以直接从发送端传输到接收端, 而不需要经过任何阻碍。而当发送端和接收端之间存在障碍物或其他物体, 信号传输路径不是直线

(NLoS)的情况下, 信号可能需要绕过障碍物或经过多次反射、散射等过程才能到达接收端。此外还可能会出现系统或服务的中断或停止运行, 特别是指网络或通信系统的暂时性故障或失效(Outage)。因此, 从接入点 j 到 MEC 服务器 k 需要考虑 LoS、NLoS 和 Outage 这三种情况, 将这三种情况的概率表示为:

$$p_{jk}^{out} = \max(0, 1 - e^{-a_{out} \cdot d_{jk} + b_{out}}),$$

$$p_{jk}^{LoS} = (1 - p_{out}(d_{jk})) e^{-a_{LoS} \cdot d_{jk}},$$

$$p_{jk}^{NLoS} = 1 - p_{out}(d_{jk}) - p_{LoS}(d_{jk}).$$

其中, d_{jk} 表示接入点 j 与 MEC 服务器 k 之间的距离, 并令 $\frac{1}{a_{out}} = 30$ 米, $b_{out} = 5.2$, $\frac{1}{a_{LoS}} = 67.1$ 米。对于那些处于中断状态的链路, 将使用固有传播损耗来表示; 而对于非中断状态的链路, 将使用带阴影的标准线性模型来表示传播损耗。

假设接入点和 MEC 服务器的天线阵列进行定向波束成形。主波束指向主要传播路径, 其他方向的辐射能量较小。为了便于分析, 这里采用扇形天线模式[21]。用 φ 表示主叶波束宽, G^i 和 G^s 分别表示主叶和侧叶的方向性增益。则接入点 j 和 MEC 服务器 k 之间的随即天线增益/干扰 G_{jk} 有三种概率不同的模式: $G_{jk} =$

$$\begin{cases} (G^i)^2, \left(\frac{\varphi}{2\pi}\right)^2 \\ G^i \cdot G^s, \left(\frac{2\varphi \cdot (2\pi - \varphi)}{(2\pi)^2}\right)^2 \\ (G^s)^2, \left(\frac{2\pi - \varphi}{2\pi}\right)^2 \end{cases} \#$$

对于路径损失模型, 设参考距离为 1 米, 且在传播距离内的传输功率受因子 $d_{jk}^{-\alpha}$

衰弱, 其中 α 为路径损失指数。当接入点 j 将数据发送至 MEC 服务器 k 时, MEC 服务器的信号与干扰加噪声比(SINR)可以表示为:

$$\zeta_{jk}^f(t) = \frac{p_j^f \cdot G_{jk} \cdot d_{jk}^{-\alpha}}{I_{jk}(t) + \sigma_f^2}$$

其中, P_j^f 为接入点 j 的前行传输功率, 当采用 LoS 链路时, $\alpha = \alpha_{LoS}$, 当采用 NLoS 链路时, $\alpha = \alpha_{NLoS}$ 。 σ_f^2 为链路的噪声方差。 $I_{jk}(t)$ 表示由其他接入点引起的干扰。

如上所述, 对于特定用户设备的卸载数据, MEC 服务器需要根据 AP 集群中所有 AP 的信号来恢复数据。因此, 每个用户设备的 AP 集群中的所有 AP 都需要通过无线回传将数据传输到特定的 MEC 服务器。接入点 j 向 MEC 服务器 k 传输数据时的传输速率可通过香农公式求得:

$$r_{jk}^f(t) = B^f \cdot \log_2(1 + \zeta_{jk}^f(t))$$

其中, B^f 表示回传带宽。

3.4 时延模型

用户设备 i 在 t 时刻本地处理任务的时延 $T_i^l(t)$ 表示公式如下:

$$T_i^l(t) = \sum \left(1 - \varpi_{ik}(t) \cdot \frac{C_i(t)}{f_i^{UE}} \right)$$

当用户设备向 MEC 服务器发送卸载请求时, 用户设备首先通过上行通道将数据信号传输到 AP 集群内的所有 AP。因此, 用户设备 i 在卸载到 AP 集群 $\Phi_i(t)$ 时的传输时延可以表示为:

$$T_i^a(t) = \frac{\varpi_{ik}(t) \cdot D_i(t)}{r_i^a(t)}$$

然后, AP 集群中的所有 AP 都将通过回传通道将所服务用户设备的数据传输到

特定的 MEC 服务器。用户设备 i 的数据传输时延可以表示为集群中所有 ap 的回传传输时延的最大值，它可以表示为：

$$T_i^f(t) = \sum \omega_{ik}(t) \cdot \max_{n \in \Phi_i(t)} \left(\frac{D_i(t)}{r_{jk}^f(t)} \right)$$

假设 MEC 服务器 k 分配给用户设备 i 的计算资源为 $f_{ik}^{MEC}(t)$ ，则用户设备 i 在 MEC 服务器上的处理时延计算如下：

$$T_i^e(t) = \sum \omega_{ik}(t) \cdot \frac{C_i(t)}{f_{ik}^{MEC}(t)}$$

因此，用户设备 i 的卸载时延可以表示为：

$$T_i^o(t) = T_i^a(t) + T_i^f(t) + T_i^e(t)$$

3.5 能耗模型

每个时钟周期的 CPU 能耗由执行频率 f_i^{UE} 决定，可以写为 $\epsilon (f_i^{UE})^2$ 。其中， ϵ 表示取决于硬件属性的能量系数，通常是 10^{-11} [22]。因此，本地能源消耗为：

$$E_i^l = \epsilon \cdot (f_i^{UE})^2 \cdot C_i(t)$$

从用户设备 i 到 MEC 服务器的传输能耗可表示为 [33]：

$$E_{ik}^t = \frac{P_{ij}}{\xi} \cdot T_i^a(t) + \frac{P_{jk}}{\xi} \cdot T_i^f(t)$$

其中， ξ 表示功率放大器效率。

3.6 问题公式化

将 t 时刻用户设备 i 卸载任务的卸载奖励 v_i 表示为时延和能量消耗的相对加权和，其公式为：

$$v_i^t = \omega_{ik}(t) \cdot \left(\beta_t \cdot \frac{T_i^f(t) - T_{ik}^o(t)}{T_i^f(t)} + \beta_e \cdot \frac{E_i^f(t) - E_{ik}^o(t)}{E_i^f(t)} \right)$$

其中， $\beta_t + \beta_e = 1$ 用于控制时延和能耗之间的权重，考虑用户设备端对于低时延的较高需求，在实验中设置时延和能耗比为 4: 1。

为了有效地进行任务卸载和资源分配，用户设备和 MEC 服务器需要将其行动的长期影响纳入其决策中。因此，实验的目标是最大化所有用户设备的长期平均总奖励。设 $\Omega = \omega_{ik}(t), i \in M, k \in K, t \in T$ 为卸载策略的数据集， $P = p_i^d(t), i \in M, t \in T$ 为上行传输功率的数据集， $F = f_{ik}(t), i \in M, t \in T$ 为计算资源分配数据集。可以得到以下的优化问题 P1：

其中，(a) 表示用户设备最多只能将其任务卸载到一个 MEC 服务器上。(b) 确保分配的计算资源不会超过计算容量。(c) 规定每个用户设备选择的发射功率应属于预定义的功率级别集。(d) 对每个用户设备的限制每个用户设备的 AP 集群的大小。(e) 要求用户设备的所有任务必须在每个时隙内完成，否则将被丢弃。(f) 和 (g) 定义了计算资源分配变量和卸载决策变量的范围。观察可以发现，这是一个混合整数非线性编程(MINLP)问题。优化目标是非凸的，这个问题可以被证明为 NP-Hard 问题。

鉴于边缘侧算力与带宽受限，且各终端彼此独立理性，单个设备在做卸载取舍时通常以自身效用最大化为目标。因此，在本实验中假设用户设备端与 MEC 端除了必要的信息传输外相互独立。分析该优化问题 P1，在该问题中需要对用户设备卸载决策 Ω 、用户设备传输功率选择 P 和 MEC 端资源分配策略 C 同时进行考虑，强行进行计算很困难。如图图 3 所示，可以将 P1

$$\begin{aligned}
\text{P1: } & \max_{\Omega, \mathbb{P}, \mathbf{C}} \sum_{t=1}^T \sum_{i=1}^M v_i(t), \\
\text{s.t. } & \mathbf{a.} \sum_{k \in \mathcal{K}} \omega_{ik}(t) \leq 1, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \\
& \mathbf{b.} \sum_{i \in \mathcal{M}} f_{ik}(t) = F_k^e, \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \\
& \mathbf{c.} p_i^d(t) \in \mathcal{P}^a, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \\
& \mathbf{d.} |\Phi_i(t)| = \Xi, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \\
& \mathbf{e.} T_i(t) \leq \tau_c, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \\
& \mathbf{f.} 0 \leq c_{ik}(t) \leq C_k^e, \forall i \in \mathcal{M}, k \in \mathcal{K}, \forall t \in \mathcal{T}, \\
& \mathbf{g.} \omega_{ik}(t) \in \{0, 1\}, \forall i \in \mathcal{M}, k \in \mathcal{K}, \forall t \in \mathcal{T}.
\end{aligned}$$

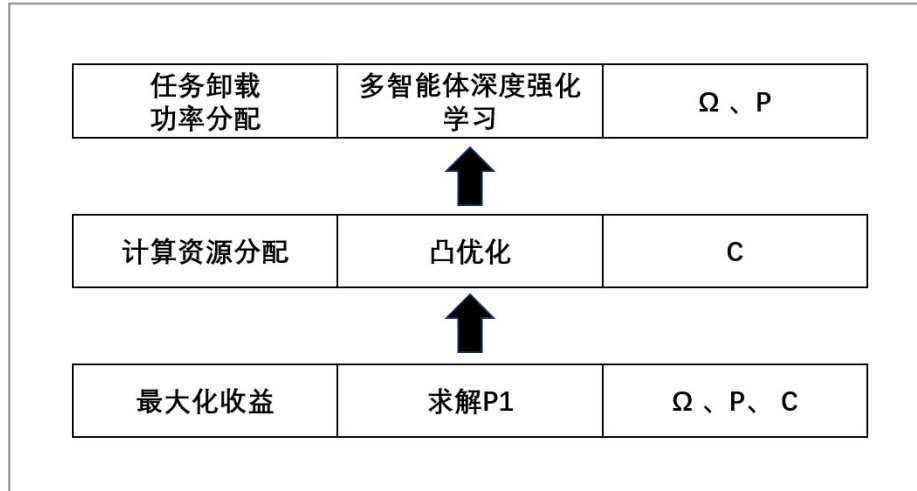


图3 求解P1的两阶段示意图

拆分成两个子问题 P2：MEC 资源分配问题和 P3：任务卸载和功率分配问题。其中，P2 可以在 MEC 侧使用凸优化解决，而 P3 可以在用户设备端使用多智能体深度强化学习来解决[23]。

4 算法设计

对于 P2 中的凸优化问题，可以利用 cvxpy 求解器进行求解。设在第 t 个时隙，MEC 侧共 K 台服务器，每台服务器 k 的可用 CPU 峰值为 $F_k^{\max}$ （单位 Hz），其服务的任务集合为 $U_k = \{i | x_{i,k} = 1\}$ ，其中 $x_{i,k}$ 表示第 i 个任务是否被卸载到服务器 k 。任务 i 的计算需求为 C_i （所需 CPU 周期数），对应服务器 k 上的分配频率为 $f_{i,k} \geq 0$ 。排除排队/抢占等复杂度，单任务的计算时

延近似为 $T_{i,k}^{cmp} = C_i/f_{i,k}$ 。考虑任务重要性，令 $w_i \geq 0$ 。则凸目标是 minimized 加权计算时延（或时延与能耗的组合项）之和：

$$\text{minimize} \sum_{k=1}^K \sum_{i \in U_k} w_i \cdot (C_i/f_{i,k}) + \rho \cdot \sum_{k=1}^K \sum_{i \in U_k} f_{i,k}^2$$

其中 $\rho \geq 0$ 为光滑化正则项。注意函数 $1/f$ 在 $f > 0$ 上为凸函数，因此上述目标与线性约束构成标准的凸规划。由于我们需要在边缘设备的有限资源中进行凸优化计算，因此我们选用 ECOS（二阶锥）作为求解器。ECOS 实现了一套原始-对偶内点法，配合自对偶嵌入与 NT 尺度来求解锥规划；每步解一个对称不定 KKT 线性系统得到牛顿方向，并用预测-校正策略加速。

P3 中的联合优化问题则通过在用户设备端部署多智能体深度强化学习算法，以获得任务卸载和功率分配的决策。我们采用多智能体 Actor-Critic (MA2C) 算法来解决联合优化问题。每个用户被视为一个独立的智能体，其目标是最大化长期回报，最小化长期成本。第一步，将终端在任务卸载与功率控制上的决策流程，形式化为一个 POMDP（部分可观测马尔可夫决策模型），然后根据每个智能体的决策目标，建立多智能体 A2C 优化算法以获得近最优策略。

在该设定下，将每台用户终端视作独立决策智能体，拥有一个本地的策略网络 (Actor)，仅基于局部观测来做出决策，而其余终端被抽象为外部环境的一部分。围绕卸载与功率控制的建模，则以相关属性为核心展开：全局状态 S 、动作空间 $\mathcal{A}$ 、观测空间 $\mathcal{O}$ 、观察函数 $\mathcal{F}$ 、状态转移概率 $\mathcal{P}$ 、奖励函数 r 、折扣因子 γ 。其中，在 t 时刻中， S 表示全局状态集 $s(t) \in S$ ，该属性对单个智能体不可见。

$\mathcal{A} = \mathcal{A}_i | i \in M$ 为一组动作空间，其中 $a_i(t)$ 智能体 i 在 t 时刻选择执行的动作组。

动作组由两个变量构成，可以表示为 $a_i(t) = \omega_{ik}(t), p_i^d(t) | \forall i \in M, \forall t \in T$ 。用 $o_i(t)$ 表示在 t 时刻智能体 i 的观测数据， $o_i(t)$ 由两部分组成，前一部分为当前信息，包括当前时刻中的任务数据大小和计算密度；后一部分为历史信息，包括智能体上一时刻所做决策和获得的收益。用公式可以表示为： $o_i(t) = \{D_i(t), f_i(t), a_i(t-1), v_i(t-1)\}, t \in T$ 。令 $\mathcal{O} = \{o_i(t) | \forall i \in M, \forall t \in T\}$ 表示所有智能体在时段 T 内的一组观测数据。观察函数用于将全局空间 S 中的信息提取局部信息作为相应智能体的观测信息，可以表示为 $\mathcal{F} = F_i: s(t) \rightarrow o_i(t), \forall i \in M, \forall t \in T$ 。 $\mathcal{P} = \{P(s(t+1)|s(t), a(t)), \forall t \in T\}$ 描述了一组状态转移概率集合。在根据智能体的目标做出决策后，智能体可以从环境中获得在当前状态 $s(t)$ 下做出动作 $a(t)$ 的奖励 $r_i(s(t), a(t))$ 。

为了获得每个智能体的最优策略，对每个智能体使用 A2C 算法框架。每个智能体应用 A2C 来学习独立的策略。A2C 作为基于策略梯度的 DRL 算法，引入优势函数来衡量每个动作相对于平均值的优势，以确保训练稳定性。每个 A2C 智能体都有两个深度神经网络，actor 网络以当前状态作为输入，生成动作作为输出近似最优策略；critic 网络评估策略并近似值函数，在 A2C 中使用更新后的策略而不是原来的策略与环境进行交互。基于 A2C 的优化框架如图 4 所示。每个时隙伊始，终端先读取任务队列与历史状态等观测量，将其组装后送入 actor 网络，进而生成当下的动作输出一涵盖卸载选择与功率配置。在任务被处理完成后，计算这段时间的奖励 $r_i(t)$ ，并将观测值、动作和奖励存入记忆缓冲池中。每个智能体间隔 B 时刻更新一次 Actor 和 Critic 网络的参数。具体来说，智能体将从记忆缓冲池中采样小批量数据

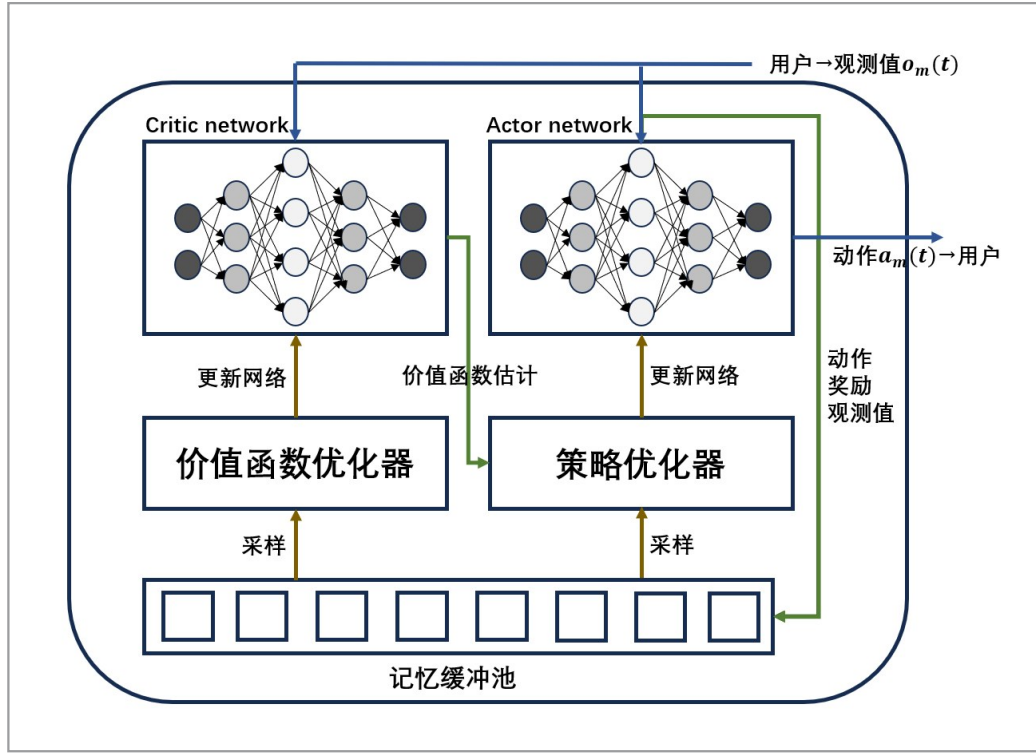


图4 A2C强化学习算法的网络框架

进行处理，并估计 Actor 和 Critic 网络的参数梯度。随后，智能体根据梯度更新网络参数，然后清除缓冲区。

对于智能体 i ，假设其策略参数为 $\pi_i(\theta_i)$ 每个智能体的策略更新过程即使为状态-价值函数的累计收益最大化：

$$\theta_i^* = \underset{\theta_i}{\operatorname{argmax}} \left[\mathcal{Q}_{\pi_i(\theta_i)}(o_i, a_i | \pi_i(\theta_i)) \right]$$

其中， $\mathcal{Q}_{\pi_i(\theta_i)}$ 表示基于当前状态和动作的价值函数。针对随机策略，其策略梯度如下所示：

$$\nabla_{\theta_i} \{ \pi_i(\theta_i) \} L_i = \mathbb{E}_{o_i \sim \mathcal{D}} \left[\nabla_{\theta_i} \log \pi_i(\theta_i) \mathcal{Q}_{\pi_i(\theta_i)}(o_i, a_i) \right]$$

用于更新 critic 网络的损失函数定义为：

$$J_i(\phi_i) = \mathbb{E}_{o_i} \left[-V_{\phi_i}(o_i) + \mathbb{E}_{o'_i \sim \pi_i} [r + V_{\phi_i}(o'_i)] \right]^2$$

其中， ϕ_i 表示 critic 网络的参数， o'_i 表示智能体 i 在下一个时刻的观测值。采用小批量随机梯度下降法更新了批评家网络的参数：

$$\phi_i(t) \leftarrow \phi_i(t) - l_i^c \nabla_{\phi_i} J_i'$$

其中， l_i^c 表示 critic 网络的学习速率。actor 网络的网络模型与 critic 网络类似，但是在最后另外添加了一层分布分类函数，用于多离散动作的生成。

基于 A2C 的任务卸载和功率分配方案的训练过程如下所示。于每个时刻 t 的起点，终端先感知外部环境（含任务队列与历史记录）。随后将上述观测整理后馈入 Actor 网络，生成当下的卸载选择与功率分配动作。最后，终端将该动作连同当前状态一并上报至 MEC 边缘服务器。MEC 服务器整合所有智能体信息来计算资源分配，获得计算奖励的当前全局状态和动作。随后，MEC 服务器将观测值、动作和奖励

存储到经验回放池中。MEC 服务器每隔 B 个时间间隔从经验回放池中取出所得数据，用于训练 Actor 和 Critic 网络，更新网络

参数。并发送消息通知各智能体的 Actor 网络更新其参数。

算法 1 基于 A2C 的 MEC 任务卸载和功率分配策略

Input: 系统环境和模型参数

Output: 卸载决策和功率分配

```

1: for  $t \in T$  do
2:   for  $i \in M$  do
3:     获取环境信息，并更新  $o_i(t)$ ;
4:     输入 Actor 网络获得任务卸载和功率分配决策;
5:      $a_i(t) = \omega_{ik}(t), p_i^d(t), \forall k \in \mathcal{K}$ ;
6:     if  $t \% B == 0$  then
7:       更新 Actor 网络  $\theta_i$ ;
8:     end if
9:   end for
10:  for MEC 服务器  $k \in \mathcal{K}$  do
11:    计算当前奖励  $r(t)$ ;
12:    将数据集  $s(t), a(t), r(t), s(t+1)$  存入回放缓冲池;
13:    if  $t \% B == 0$  then
14:      计算 Actor 网络和 Critic 网络梯度  $\nabla_{\theta_i} L_i$  和  $\nabla_{\phi_i} L_i$ ;
15:      更新 Critic 网络  $\Phi_i(t)$ ;
16:      将 Actor 网络梯度传给各用户设备;
17:    end if
18:  end for
19:  任务处理完成后返回结果给用户设备;
20: end for

```

5 仿真实验

5.1 仿真参数设置

在仿真实验中，以二维数组表示经纬度用于定位用户设备、AP 位置，在考虑实际情况简化模型，将实验范围设置为 1000 米×1000 米的区域，使用随机函数生成用户设备和 AP 的位置，并设置 3 个位置固定的边缘服务器。设定每个用户设备限制其计算能力在 3GHz 到 5GHz 之间，每个 MEC 服务器的计算能力在 10GHz 到 20GHz 之间。按细粒度划分计算任务，每个用户设备的计算任务在 50kb 到 100kb 之间。系统模型中信道参数设置如表 1 所

示。该实验中的上行信道带宽参考赵俊皓等人提供的数据[24]，其他部分数据参考 Ngo 在其实验中提供的数据[18]。

每次实验均迭代 1×10^6 次，平均每 100 次迭代作为一个 episode，共计 100 个 episodes，实验证明该迭代数可使模型收敛。

5.2 主实验分析

为了比较该方法与其他方法的优越性，我们对比了 MAPPO 和 MADDPG 算法。在保证相同实验设置的情况下，采用不同的强化学习算法，得到的实验结果如下图所示。结果显示，我们的方法相比 MAPPO 可以提高 5% 的整体收益，相比 MADDPG 可以提高 2% 的整体收益。

表1 信道参数列表

参数	符号	值
时隙持续时间(s)	Δt	0.1
路径损失阈值	d_0, d_1	10、15
阴影衰落标准偏差(dB)	σ^s	8
4G上行信道带宽(Hz)	B^a	20M
载波频率(Hz)	f	1.9G
用户设备、AP天线高度(米)	h_u, h_a	1.65、15
5G回传信道带宽(Hz)	B^f	100M
阻塞密度系数	ϵ	6×10^{-4}
主叶方向增益	G^i	63.1
侧页方向增益	G^s	0.631
LoS路径损失指数	α^l	2.5
NLoS路径损失指数	α^j	2.5
接入点配置天线数	u	20

为了检验加入基于强化学习算法的联合功率控制策略是否可以降低传输时延和能耗，令所有用户设备均使用最大功率传输卸载任务作为对照组，对比考虑功率控制前后系统的性能。如图5所示，考虑功率控制后的联合卸载模型的长期收益优于使用最大功率的长期收益。该结果证明引入基于强化学习的联合功率分配策略可以实现用户设备低时延和低能耗的双重收益。

5.4 基于系统模型的结果分析

利用控制变量法设计三个实验，分别用于检验在不同用户设备数、APs的数目、AP集群大小的条件下，长期奖励的变化情况。其中，设置不同用户设备数为10、15和20；设置不同APs的数目为100、200和300；设置不同AP集群大小为3、6和9。

图6展示了在接入点集群数控制为100，集群大小为5，改变用户设备数为10、15、20的情况下，迭代100个episode后的模型结果。可以从结果看出，随着用户设备端数目增加，长期累计的时

延和能耗将增加。这说明在网络模型不变的情况下，用户设备数需要控制在一定范围内以满足用户设备对于低时延和低能耗的要求。这需要通过增加基建规模、提高通信带宽以及MEC服务器算力来实现。

图7展示了在用户设备数为10，集群大小为5的条件下，改变接入点集群数为100、200、300的情况下，迭代100个episode后的模型结果。可以从结果看出，当APs集群数控制在一定范围内时，随着APs的数目增加，长期累积的时延和能耗总计减少。但是当APs集群数超过一定范围后，规模效应的出现逆增长，APs集群数为300时的长期奖励值反而小于200时。该现象可以与图6中所出现的情况相互印证。证明通过增加基建数，即增加用户设备可选择接入点集群的数目，来减少累计时延和能耗总计，满足用户设备对于低时延、低能耗的要求。但不可以盲目增加接入点集群的数目，需要在可控的范围内提高模型性能。

图8展示了在用户设备数为10，接入点集群数为200条件下，改变接入点集群大小为3、6、9的情况下，迭代100个episode后的模型结果。可以从结果看出，改变接入点集群大小并不能优化长期累积的时延和能耗。因此在实际应用中，仅考虑增加接入点集群大小无法提高模型整体性能。

6 总结与展望

随着5G和6G技术的不断推进，涌现了许多新技术的讨论。近年来，Cell-free massive MIMO (CF-mMIMO) 成为热点方向。该体系通过分布式部署大量接入点，可在广域内并行为多终端提供链

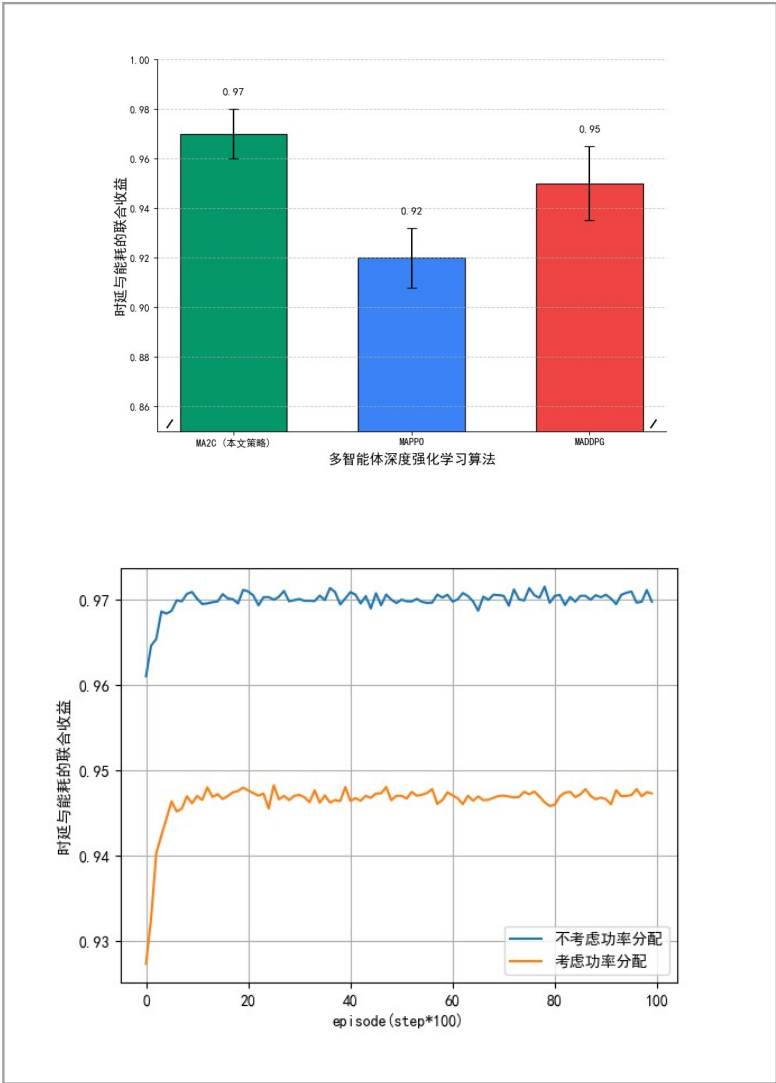


图5 不同功率分配的长期奖励值

路支持，从而显著提升总体吞吐与服务可靠性。为充分释放边缘侧能力，本文将CF-mMIMO的网络形态与MEC深度耦合，系统考察新架构下的任务卸载机制。我们的目标是在终端能量预算受限时，联合优化以压缩时延并降低功耗，构建综合代价最小的决策与算法。主要工作概述如下：

(1) 调研了现有的边缘计算卸载策略的实现方式以及去蜂窝大规模MIMO通信系统的网络架构，并结合已有文献考察基于

去蜂窝大规模MIMO系统的边缘计算任务卸载策略的可行性，明确了在用户设备与边缘节点计算能力有限的情况下优化卸载任务与资源分配的问题范式。

(2) 将去蜂窝大规模MIMO通信架构于边缘计算模型结合，搭建了基于去蜂窝大规模MIMO的边缘计算网络模型，该模型可以整合端设备与边缘节点的算力资源，通过去蜂窝模型获得更高的频率带宽以传递更多的计算信息。

(3) 设计实现边缘计算中任务卸载与资

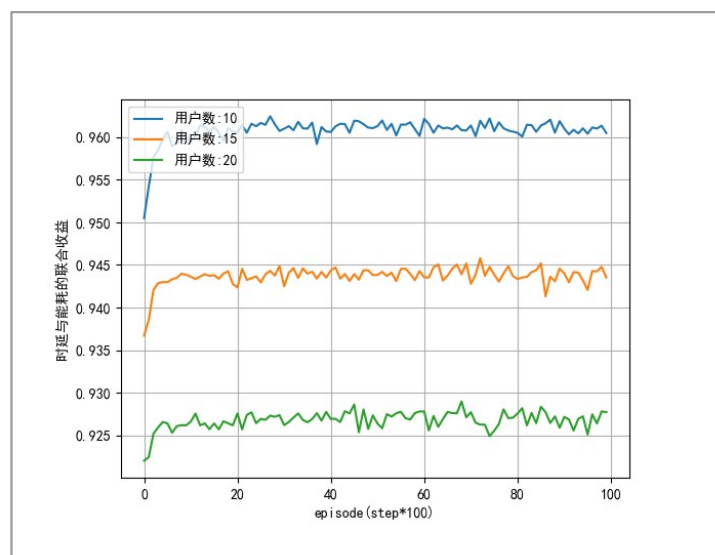


图6 不同用户设备数的长期奖励值

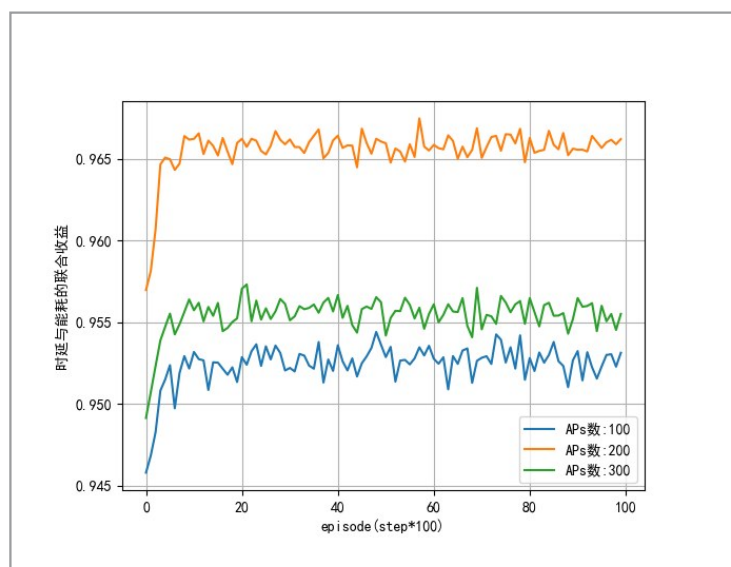


图7 不同Aps数的长期奖励值

源分配联合优化问题的解决方案，将多智能体深度强化学习算法和传统凸优化方法结合，以最大程度降低累计时延与能耗为目的，进行算法设计与任务拆解，最终完成问题的求解策略。

未来的系统将更加智能化，能够根据实时网络状态、用户设备需求以及资源利用情况等因素，动态地调整任务卸载策略。

这样的智能化决策能够使得任务卸载更加高效、灵活，从而优化系统性能。随着边缘计算和去蜂窝大规模MIMO网络技术的不断发展，未来的任务卸载策略将更加智能化和自适应。边缘节点将具备更强大的计算和通信能力，能够处理更多的任务并实现更低的时延。同时，强化学习等智能算法的应用将使得系统能够根据实时环境

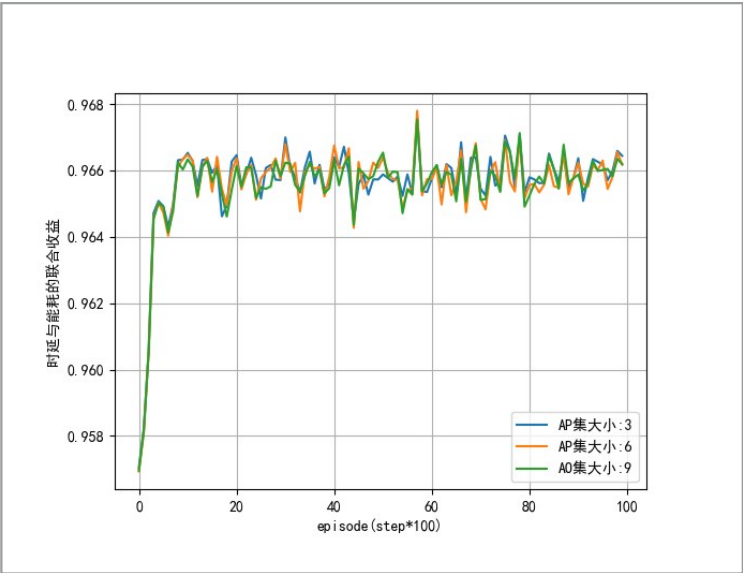


图8 不同集群大小的长期奖励值

和用户设备需求进行智能化决策，进一步提升任务卸载的效率和性能。

此外，随着多媒体和大数据应用的普及，未来的边缘计算任务卸载策略将更加注重对这些应用的优化。可能会针对不同类型的任务，设计专门的卸载策略，以满足对于低时延、高带宽和高可靠性的需求。例如，针对视频流等实时传输应用，可以设计特定的任务调度和传输机制，以确保视频数据的实时性和流畅性。对于大数据处理任务，可以采用分布式计算和数据预处理等技术，以提高数据处理的效率和速度。

未来的任务卸载策略需要更加注重安全性和隐私保护。可能会引入加密和身份验证等技术，以保护用户设备数据在卸载过程中的安全性和隐私性。随着边缘计算系统与物联网的深度融合，涉及到的用户设备数据种类和数量将会急剧增加，因此保护用户设备隐私将成为任务卸载策略设计的重要考虑因素。未来可能会出现更加复杂的安全和隐私保护机制，例如基于区块链技术的数据交换和验证系统，以确保用

户设备数据在边缘计算环境中的安全传输和处理。

总的来说，未来的边缘计算任务卸载策略将更加智能化、灵活和安全，能够更好地满足未来应用的需求，并发挥更大的作用。随着技术的不断进步和创新，边缘计算将成为未来网络 and 智能系统的重要组成部分，为人们的生活和工作带来更多的便利和效率。

参考文献：

[1] Wang F, Xu J, Wang X, et al. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[J]. IEEE transactions on wireless communications, 2017, 17(3): 1784-1797.

[2] 周杰钦,冯毅雄,金柯兵,等.多UAV协作下的边缘计算部署、卸载与资源优化[J/OL]. 计算机工程, 1-22[2025-10-09].<https://doi.org/10.19678/j.issn.1000-3428.0252376>.

[3] Chen Z, Xiong B, Chen X, et al. Joint

- computation offloading and resource allocation in multi-edge smart communities with personalized federated deep reinforcement learning[J]. IEEE Transactions on Mobile Computing, 2024, 23 (12): 11604–11619.
- [4] Shen B, Gu Q, Yang G. Dynamic task offloading and online scheduling for Edge-enabled IoT with a hierarchical framework[J]. Computer Networks, 2025: 111486.
- [5] Khani M, Sadr M M, Jamali S. Deep reinforcement learning - based resource allocation in multi - access edge computing[J]. Concurrency and Computation: Practice and Experience, 2024, 36 (15): e7995.
- [6] Bolourian M, Shah-Mansouri H. Deep Q-learning for minimum task drop in SWIPT-enabled mobile-edge computing [J]. IEEE Wireless Communications Letters, 2024, 13(3): 894–898.
- [7] Huang B, Chen X, Li J, et al. Deep reinforcement learning for optimizing computation latency in wireless-powered Multi-Access Edge Computing systems: A partial offloading approach [J]. Ad Hoc Networks, 2025: 103971.
- [8] Ren T, Hu Z, Niu J, et al. ExplabOff: Towards Explorative and Collaborative Task Offloading via Mutual Information-Enhanced MARL[C]//IEEE INFOCOM 2025–IEEE Conference on Computer Communications. IEEE, 2025: 1–10.
- [9] Tang H, Du M, Wu H, et al. Link Topology-Adaptive Offloading Method On Vehicular Edge Computing[C]//IEEE INFOCOM 2024–IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2024: 01–02.
- [10] Hao Y, Yang S, Li F, et al. Edgetimer: Adaptive multi-timescale scheduling in mobile edge computing with deep reinforcement learning[C]. IEEE INFOCOM 2024–IEEE Conference on Computer Communications. IEEE, 2024: 671–680.
- [11] 赵迪, 郭爱煌, 宋春林. 去蜂窝大规模MIMO系统功率控制算法[J]. 系统工程与电子技术, 2021, 43(03): 854–860.
- [12] Mobini Z, Gokceoglu A H, Wang L, et al. Fronthaul-Aware User-Centric Generalized Cell-Free Massive MIMO Systems[J]. arXiv preprint arXiv: 2506.14494, 2025.
- [13] Interdonato G, Buzzi S. Joint optimization of uplink power and computational resources in mobile edge computing-enabled cell-free massive MIMO[J]. IEEE Transactions on Communications, 2023, 72(3): 1804–1820.
- [14] Abdullaev I, Prodanova N, Bhaskar K A, et al. Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning[J]. Computers, Materials & Continua, 2023, 76(2).
- [15] Buzzi S, D’Andrea C, Zappone A, et al. User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach[J]. IEEE Transactions on Wireless Communications, 2019, 19(2): 1250–1264.
- [16] Zhao Y, Niemegeers I G, De Groot S M H. Dynamic power allocation for cell-free massive MIMO: Deep reinforcement learning methods[J]. Ieee Access, 2021, 9: 102953–102965.
- [17] Femenias G, Riera-Palou F. Cell-free millimeter-wave massive MIMO systems with limited fronthaul capacity[J]. IEEE Access, 2019, 7: 44596–44612.
- [18] Ngo H Q, Ashikhmin A, Yang H, et al. Cell-free massive MIMO versus small cells[J]. IEEE transactions on wireless communications, 2017, 16(3): 1834–

1850.

[19] Ma H, Huang P, Zhou Z, et al. GreenEdge: Joint green energy scheduling and dynamic task offloading in multi-tier edge computing systems[J]. IEEE Transactions on Vehicular Technology, 2022, 71(4): 4322–4335.

[20] Lin H, Ishibashi K, Shin W Y, et al. A simple random access scheme with multi-level power allocation[J]. IEEE Communications Letters, 2015, 19(12): 2118–2121.

[21] Di Renzo M. Stochastic geometry modeling and analysis of multi-tier millimeter wave cellular networks[J]. IEEE Transactions on Wireless Communications, 2015, 14(9): 5038–5057.

[22] Chen X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(4): 974–983.

[23] Li H, Assis K D R, Yan S, et al. DRL-based long-term resource planning for task offloading policies in multiserver edge computing networks[J]. IEEE Transactions on Network and Service Management, 2022, 19(4): 4151–4164.

[24] 赵俊皓,张俊文,迟楠.面向下一代移动通信的前传网络关键技术[J].中兴通讯技术,2023,29(05):45–55.

作者简介

石家琪（2002– ），女，中国科学技术大学硕士研究生，平安科技（深圳）



张旭龙（1988– ），男，博士，平安科技（深圳）有限公司，算法研究员，复旦大学计算机理学博士，2023年入选上海市东方英才计划青年项目。兼任清华大学深圳研究院及中国科学技术大学先进技术研究院校外导师，以及中国指挥与控制学会具身智能专业委员会委员、中国自动化学会联邦数据与联邦智能委员会委员、中国电子音响行业协会声音与音乐技术专委会常务委员。目前是IEEE、中国自动化学会以及中国计算机学会会员。主要研究方向包括大模型、具身智能、跨模态智能计算等。发表学术论文80余篇，获得国家发明专利授权20余项。联系邮箱：zhangxulong@ieee.org



瞿晓阳（1988– ），男，博士，平安科技前沿机器学习算法分组负责人，华中科技大学博士，兼任清华大学深圳国际研究院、中国科学技术大学先进技术研究院、哈尔滨工业大学校外研究生导师，以及美国中佛罗里达大学访问学者。长期从事机器学习、大数据与计算体系结构方向的研究，在大模型推理加速、具身智能、联邦学习、持续学习、高性能计算与存储等领域具有丰富的研究与实践经验。近年来，在体系结构方向（如INFOCOM、DAC、TPDS、IPDPS）和人工智能方向（如NeurIPS、AAAI、ACL、ICCV、IJCAI）等国际顶级会议及期刊上发表论文60余篇，多篇论文获评最佳论文奖；担任多家国际顶级期刊审稿人。已授权国家发明专利100余项，出版专著2本，译著1本。联系邮箱：quxiaoy@gmail.com



倪晓俊（1973–），男，硕士，现任中科院计算技术研究所助理研究员。主要从事数据库安全、大数据分析、隐私计算等方面的研究工作。曾获得北京市科技进步一等奖等奖励。jamesni@sina.com



田晖，男，博士，教授，博士生导师；现任华侨大学计算机科学与技术学院院长、网络与信息安全产业学院院长、厦门市数据安全与区块链技术重点实验室主任；入选福建省青年拔尖人才、福建省百千万人才工程省级人选、福建省高校新世纪优秀人才、福建省高校杰出科研人才培养计划。现为IEEE 高级会员（Senior Member）、中国计算机学会（CCF）杰出会员、CCF 互联网专委会执行委员、CCF 区块链技术专委会执行委员、中国中文信息学会大数据安全与隐私保护专委会常务委员、福建省计算机学会副理事长、福建省网络空间安全学会理事长。主要从事网络与信息安全、人工智能安全，数据安全、多媒体内容安全等方面的研究工作。近年来，主持各类科技项目20余项，其中包括国家自然科学基金项目4项，国家重点研发课题和子课题各1项；在IEEE TDSC、IEEE TIFS、IEEE TASLP、ACM MM等国内外著名期刊和知名学术会议上发表学术论文120余篇；获得授权发明专利40余项，软件著作权8项；部分研究成果曾荣获中国电子学会科学技术奖二等奖、海南省自然科学二等奖、福建省科学技术进步奖三等奖和福建省自然科学奖三等奖等奖励。htian@hqu.edu.cn



王健宗（1983–），男，博士，正高级工程师，平安科技（深圳）有限公司，智能金融前沿技术研究院院长。美国佛罗里达大学人工智能博士后，美国莱斯大学和华中科技大学联合培养博士，中国计算机学会杰出会员，中国计算机学会大数据专家委员会委员，中国自动化学会联邦数据和联邦智能专业委员会副主任。主要研究方向为具身智能、大模型、联邦学习和深度学习等。荣获中国专利优秀奖、广东省科技进步二等奖、吴文俊人工智能科技进步一等奖、人民银行金融发展一等奖、深圳高层次专业人才领军人才、福田英才等奖项。联系邮箱：jzwang@188.com

有限公司算法工程师，主要研究方向为强化学习，边缘计算，多模态大模型等。

收稿日期: XXXX-XX-XX
通信作者: 张旭龙, zhangxulong@ieee.org
基金项目: 深港联合资助项目(A类)(No. SGDX20240115103359001)
Foundation Items: Shenzhen-Hong Kong Joint Funding Project (Category A) under Grant No. SGDX20240115103359001