

基于深度强化学习的任务卸载和资源分配优化

龚亮亮^{1,2}, 张影^{1,2}, 张俊尧^{1,2}, 许之琛³, 康彬^{3*}

(1. 国网电力科学研究院有限公司, 江苏 南京 210006;

2. 南京南瑞信息通信科技有限公司, 江苏 南京 210006;

3. 南京邮电大学 物联网学院, 江苏 南京 210003)

摘要:移动边缘计算(MEC)可以在网络边缘为用户提供就近的存储和计算服务,从而为移动用户带来低能耗、低时延的优势。该文针对基于超密集网络(UDN)的多用户多MEC场景,从用户侧出发,以最小化用户计算总开销为目的,解决用户在卸载过程中的卸载决策和上传传输功率优化以及MEC计算资源分配问题。具体而言,考虑到该问题是一个具有NP-hard性质的MINLP问题,该文将该问题分解为两个子问题并通过两个阶段的方式进行求解。首先在第一个阶段设计了一种基于深度强化学习(DQN)的任务卸载决策来解决任务卸载子问题,然后在第二个阶段分别使用KKT条件以及黄金分割算法解决MEC计算资源分配和上行传输功率的优化问题。仿真结果表明,所提方案在保证用户时延约束的前提下,有效降低了用户的计算开销,提升了系统性能。

关键词:超密集网络;移动边缘计算;任务卸载;资源分配;深度强化学习

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2024)04-0116-08

doi:10.20165/j.cnki.ISSN1673-629X.2024.0018

Joint Optimization of Task Offloading and Resource Allocation Based on Deep Reinforcement Learning

GONG Liang-liang^{1,2}, ZHANG Ying^{1,2}, ZHANG Jun-yao^{1,2}, XU Zhi-chen³, KANG Bin^{3*}

(1. State Grid Electric Power Research Institute Co., Ltd., Nanjing 210006, China;

2. Nanjing Nanrui Information Communication Technology Co., Ltd., Nanjing 210006, China;

3. School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Mobile edge computing (MEC) can provide users with nearby storage and computing services at the edge of the network, so as to bring the advantages of low energy consumption and low delay to mobile users. Aiming at the multi-user and multi MEC scenario based on ultra-dense network (UDN), starting from the user side and aiming at minimizing the total user computing overhead, we solve the problems of user unloading decision, upload transmission power optimization and MEC computing resource allocation in the unloading process. Specifically, considering that the problem is a NP hard MINLP, we decompose the problem into two subproblems and solves it in two stages. Firstly, in the first stage, a task offloading decision based on deep reinforcement learning (DRL) is designed to solve the task unloading sub problem, and then in the second stage, KKT condition and golden section algorithm are used to solve the optimization problems of MEC computing resource allocation and uplink transmission power respectively. Simulation results show that the proposed scheme effectively reduces the user's computing overhead and improves the system performance on the premise of ensuring the user's delay constraint.

Key words: ultra-dense network; mobile edge computing; task offloading; resource allocation; deep reinforcement learning

0 引言

近年来,随着无线通信技术的飞速发展,催生了一批如人脸识别、虚拟现实和增强现实^[1-2]等新型应用,

智能设备产生的海量数据给5G网络带来了严峻的挑战。此外,5G网络新增加的业务场景对能效、延迟等指标提出了更高的要求。依靠传统的蜂窝无线网络所

收稿日期:2023-04-04

修回日期:2023-08-06

基金项目:国家自然科学基金面上项目(62171232, 62071255, 62001248);国家博士后面上基金(2020M681684);江苏省高校自然科学基金重大项目(20KJA510009)

作者简介:龚亮亮(1987-),男,工程师,研究方向为电力通信;通信作者:康彬(1985-),男,博士,副教授,研究方向为计算机视觉以及多媒体信号处理。

提供的无线接入能力无法满足日益增长的移动设备的接入需要^[3]。为了应对这些严峻的挑战,超密集网络(UDN)被认为是解决这些问题的关键技术之一^[4]。UDN由小基站(SBS)和宏基站(MBS)组成,凭借SBS的部署,UDN可以为用户设备(UE)提供巨大的接入能力^[5]。同时,移动边缘计算(MEC)技术可以弥补移动设备计算能力不足的缺点,通过在网络的边缘部署MEC服务器,为移动设备提供丰富的计算资源,并通过计算卸载减少用户设备的计算时延和能耗。因此,与MEC集成的UDN被视为未来5G发展的趋势之一。

MEC技术目前在学术界引起了广泛的关注。在MEC研究中,一些工作研究了基于MEC的移动设备如何做出卸载决策,以改善时延、能耗或者系统效用。文献[6]基于强化学习方法提出了一种高效计算卸载方案,解决了多用户的MEC场景中最小化系统成本的问题;在文献[7]中作者设置了一组基于云计算卸载的移动用户,并通过博弈论算法得到最优解;文献[8]的主要工作为考虑MEC支持的IoT中卸载决策问题,并最小化用户的时延和能耗加权和。此外,UDN中的资源分配问题也受到了高度关注。一些工作研究了UDN中的资源分配问题,通过分配频谱和信道等资源来减少干扰,从而提高了系统性能;在文献[9]中,作者考虑多类型用户卸载的场景,将任务卸载表述为一个整数非线性规划问题,提出了一种基于差分进化算法的高效任务卸载和信道资源分配方案;文献[10]研究基于SDN的超密集网络中的任务卸载问题,将优化问题转化为两个子问题,即任务卸载子问题和MEC计算资源分配子问题并加以解决,实验结果验证了其算法的有效性;文献[11]在多用户边缘计算网络中,将资源分配策略与计算任务卸载决策进行了联合优化,针对多用户多边缘服务器场景,考虑边缘服务器有限的无线资源、存储资源和计算资源,提出了基于多回合双向拍卖的边缘服务器资源分配算法;文献[12]针对正交频分多址的多用户移动边缘计算系统,以任务时延为主要优化目标,设置任务时限、设备能量、通信资源等约束条件,提出了一个结合通信和边缘服务器计算资源分配的部分卸载方法。该方法在满足用户最低时延的条件下设置初始卸载比和分配通信资源,然后根据服务器计算能力来分配剩余计算资源,最后根据资源分配情况优化卸载比;文献[13]应用了一种基于遗传算法的深度学习方案对任务卸载和资源分配策略进行优化以最小化系统总成本。首先,通过遗传算法对任务卸载和资源分配策略进行优化,获得一个在当前场景模型下的最优策略;然后,将该场景模型通过特征提取存储为特征矩阵,将最优策略存储为对

应的标签;接着,将特征矩阵和标签用于训练卷积神经网络;最后,通过训练好的神经网络很快地解算出与输入场景对应的优化策略。

然而文献[6]只关注用户卸载决策的优化,没有考虑计算资源、频谱资源的相关优化;文献[7]降低了计算复杂度,但当用户数增大时计算量没有明显变化;文献[8]没有考虑计算任务时延的约束;文献[9]优化目标仅关注用户的时延而忽略用户的能量消耗,且没有考虑卸载过程中产生的资源分配问题;文献[10]只考虑MEC计算资源分配,而忽略了用户上行传输功率等其他因素对任务卸载的影响;文献[11-13]的方案没有考虑所描述的海量用户数据的情况。因此,上述文献无法确保在资源有限以及时延约束的场景下,综合考虑用户时延以及能量消耗,联合优化任务卸载决策以及资源分配,从而高效地利用UDN-MEC场景下的无线资源和计算资源。

针对以上问题,该文研究了一个多用户多MEC服务器场景下的任务卸载和资源优化问题,从用户侧出发,以最小化用户计算总开销(时延、能耗加权和)为目标,解决用户在卸载过程中的卸载决策和上传传输功率优化以及MEC计算资源分配问题。具体而言,考虑到该问题是一个具有NP-hard性质的混合整数非线性规划(Mixed-Integer Nonlinear Programming, MINLP)问题,该文将该问题分解为两个子问题并通过两个阶段的方式进行求解。首先,在第一个阶段设计了一种基于DQN的任务卸载决策来解决任务卸载子问题,按照任务属性按比例分配通信资源求得最优的任务卸载策略;然后,在第二个阶段将资源分配问题分解为MEC计算资源分配子问题和用户上行传输功率分配子问题,并分别采用凸优化理论、KKT(Karush-Kuhn-Tucker)条件以及黄金分割法解决。仿真结果表明,提出的卸载决策和资源分配联合优化算法能够有效降低用户的计算开销。

1 系统模型及问题描述

1.1 系统模型

该文考虑多用户多MEC场景,系统模型如图1所示。该超密集网络由一个宏基站(MBS)和S个小基站(SBS)以及N个UE(如智能手机、平板电脑和可穿戴设备)组成,其中UE的集合表示为 $\mathcal{N} = \{1, 2, \dots, N\}$,SBS的集合表示为 $\mathcal{S} = \{1, 2, \dots, S\}$ 。每个SBS都部署了一台MEC服务器(用 $\mathcal{S} = \{1, 2, \dots, S\}$ 同样来表示MEC服务器的集合),可以为UE提供更强的计算能力,并且可以通过相应的SBS提供的无线信道与UE通信,上行传输机制采用OFDMA技术,因此同一个小区内不存在信道干扰,B表示系统带宽。

在图 1 所示的网络场景中,UE 根据与基站的距离随机地分布在多个 MEC 服务器服务的范围内,每个用户可以选择将计算任务在本地执行或者卸载到附近的一个 MEC 服务器上执行,且最多只能关联一个 SBS 并将计算任务卸载至所对应的 MEC 服务器上。UE 的卸载决策表示为 $x_{i,j} \in \{0,1\}$, $\forall i \in \mathcal{N}, j \in \mathcal{S}$, 若 $x_{i,j} = 1$ 表示任务 M_i 将会通过信道卸载至 MEC 服务器上

完成, $x_{i,j} = 0$ 则表示在本地执行。定义卸载至 MEC_j 的 UE 集合为 $\mathcal{N}_j = \{1, 2, \dots, N_j\}$, $\forall j \in \mathcal{S}$ 。每个 UE 都有一个待处理的计算任务需要在指定的时间限制内完成,并且每个任务具有原子性。用三元模型 $M_i = \{d_i, \lambda_i, T_i^{\max}\}$ 表示计算任务,其中 d_i 表示完成任务 M_i 需要的总 CPU 周期数即完成任务的计算量, λ_i 表示任务的数据量, T_i^{\max} 表示任务可容忍的最大时延。

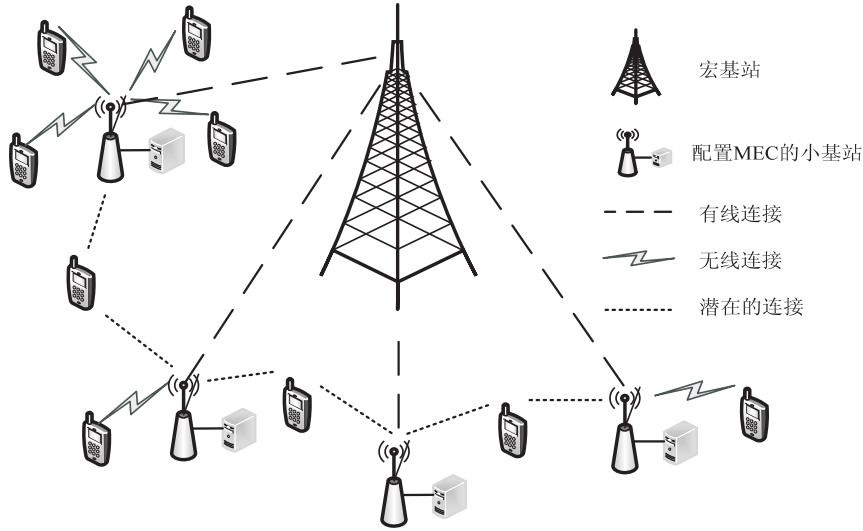


图 1 系统模型

1.2 本地计算模型

当 $x_{i,j} = 0$ 时,表示 UE 选择将任务 M_i 在本地执行。用 f_i^l 表示设备 i 的本地计算能力(即单位时间内执行的 CPU 周期)。则任务 M_i 在本地执行所需的计算时延为:

$$T_i^l = \frac{d_i}{f_i^l} \quad (1)$$

在执行该任务的同时,任务 M_i 的本地计算能耗为:

$$E_i^l = \kappa_i d_i \quad (2)$$

其中, $\kappa_i = \rho^l (f_i^l)^2$ 表示执行单位 CPU 周期所需要消耗的能量^[14], ρ^l 的大小取决于 UE 芯片架构的系数,通常取 5×10^{-27} ^[15]。

为了表述用户的计算代价,定义本地计算的开销为本地计算时延和本地计算能耗的加权和,根据式 1、式 2,表示为:

$$z_i^l = w_i^t T_i^l + w_i^e E_i^l \quad (3)$$

其中, w_i^t 和 w_i^e 定义为时延权重和能耗权重,并且满足 $w_i^t, w_i^e \in [0, 1]$, $w_i^t + w_i^e = 1$ 。具体来说,电池电量低的用户可以增加 w_i^t ,降低 w_i^e ,从而在消耗较长的任务完成时间的同时节省更多的能量。在实际应用中,移动用户可以设定不同的 w_i^e 值来控制 UE 的不同电量模式,比如当 $w_i^e = 1$ 时,表示电量模式为“低电量模式”,表示此时只关注 UE 的能耗,从而尽可能地节省能耗;

当 $w_i^e = 0$ 时,表示“高性能模式”,表示此时不关注 UE 的能耗,尽可能的以最快的速度执行任务。

1.3 MEC 计算模型

当 $x_{i,j} = 1$ 时,表示由用户 i 产生的任务 M_i 将会通过子信道卸载至 MEC 服务器 j 上执行。通过此方式,可以有效地节省本地能耗,与此同时,此过程将会产生额外的时间与能耗用于任务数据的上传。此时,典型的 MEC 计算模型主要分为三个阶段:

- (1) 任务上传阶段: UE_i 通过子信道将计算任务 M_i 上传至 MEC 服务器 j ;
- (2) 任务执行阶段: MEC 服务器给任务 M_i 分配相应的计算资源,任务在 MEC 服务器上执行;
- (3) 结果返回阶段: MEC 服务器将执行完的计算结果返回给 UE。

一般来说,忽略第三阶段任务执行结果的回程延迟,因为任务执行结果远远小于输入数据量。上行传输阶段考虑 OFDMA 模型,连接到同一个 SBS 的 UE 之间不存在小区间的干扰。任务 M_i 的上行传输速率为:

$$r_{i,j} = \frac{B}{N_j} \log_2 \left(1 + \frac{P_i g_{i,j}}{\sigma^2 + \sum_{n \in \mathcal{N}/\{i\}} \sum_{s \in \mathcal{S}/\{j\}} P_n g_{n,s}} \right) \quad (4)$$

其中, N_j 表示卸载至第 j 号 MEC 服务器的用户数量, p_i 表示 UE_i 的上行传输功率, $g_{i,j}$ 表示 UE_i 和 MEC-SBS_j 之间的信道增益, σ^2 表示背景高斯噪声,

$\sum_{n \in \mathcal{N} \setminus \{i\}} \sum_{s \in \mathcal{S} \setminus \{j\}} p_n g_{n,s}$ 表示计算卸载过程中,其他用户的接入对 UE_i 的干扰。因此,任务 M_i 的上行传输时延为:

$$T_{i,j}^{\text{tr}} = \frac{\lambda_i}{r_{i,j}} \quad (5)$$

UE 在卸载任务时需要消耗能量,传输能耗表示为:

$$E_{i,j}^c = p_i T_{i,j}^{\text{tr}} \quad (6)$$

在任务执行阶段假设 MEC 服务器为任务 M_i 分配计算资源 $f_{i,j}^c$, 则任务 M_i 在 MEC 服务器 j 的计算时延为:

$$T_{i,j}^{\text{exe}} = \frac{d_i}{f_{i,j}^c} \quad (7)$$

任务 M_i 在 MEC 服务器 j 的总的处理时延表示为:

$$T_{i,j}^c = T_{i,j}^{\text{tr}} + T_{i,j}^{\text{exe}} \quad (8)$$

此外,此模型是以用户为中心,没有考虑 MEC 服务器上的计算能耗,只考虑用户的传输能耗^[16]。那么可以得到远端计算的总开销,根据式 7 和式 8,表示为:

$$z_i^c = w_i^t T_{i,j}^c + w_i^e E_{i,j}^c \quad (9)$$

1.4 问题描述

综上所述,由式 3、式 9 可以得到 UE 的计算开销,由式 10 表示:

$$z_i = (1 - x_{i,j}) z_i^l + x_{i,j} z_i^c \quad (10)$$

将所有 UE 的卸载决策表示为 $X = \{x_{i,j}, \forall i \in \mathcal{N}, \forall j \in \mathcal{S}\}$, 同理,将每个 MEC 服务器分配给 UE 的计算资源和 UE 的上行传输功率表示为 $F = \{f_{i,j}^c, \forall i \in \mathcal{N}, \forall j \in \mathcal{S}\}$ 和 $P = \{p_i, \forall i \in \mathcal{N}\}$ 。该文的优化目标为通过优化卸载决策 X 和 MEC 计算资源分配 F 以及用户上行传输功率分配 P 来最小化所有 UE 的计算开销。根据上述讨论,联合计算卸载和资源分配的优化问题 P 表示为:

$$\begin{aligned} \text{P: } \min_{X, C, F} Z(X, P, F) &= \sum_{i=1}^N \sum_{j=1}^S z_i \\ \text{s. t. } (C_1) \quad &x_{i,j} \in \{0, 1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{S} \\ (C_2) \quad &\sum_{j=1}^S x_{i,j} \leq 1, \forall i \in \mathcal{N} \\ (C_3) \quad &f_{i,j}^c > 0, \forall i \in \mathcal{N}, \forall j \in \mathcal{S} \\ (C_4) \quad &\sum_{i \in \mathcal{N}} f_{i,j}^c \leq f_m, \forall j \in \mathcal{S} \\ (C_5) \quad &0 \leq p_i \leq P_i^{\max}, \forall i \in \mathcal{N} \\ (C_6) \quad &(1 - x_{i,j}) T_i^l + x_{i,j} T_{i,j}^c \leq T_i^{\max}, \forall i \in \mathcal{N}, \forall j \in \mathcal{S} \end{aligned} \quad (11)$$

其中,约束 C_1 表示卸载决策 $x_{i,j}$ 是一个二进变量,每个任务只能在本地执行或者卸载到 MEC 服务器上执行;约束 C_2 表示任务卸载最多只能选择其中一个

MEC 服务器; C_3 和 C_4 确保每个 MEC 服务器分配给所有用户的总计算资源不得超过 MEC 服务器的最大计算能力,且任务卸载时必须得到一个正的计算资源;约束 C_5 限制了 UE 的上行传输功率不得超过 UE 的最大上行传输功率 P_i^{\max} ;约束 C_6 确保任务总执行时延不能超过任务的可容忍的最大时延。

2 卸载决策和资源分配联合优化算法

通过观察分析,式 11 是一个具有 NP-hard 性质的非凸的 MINLP,难以直接求解。为降低问题的复杂度,该文将其分解成两个子问题并分成两个阶段来进行求解。第一个阶段解决卸载策略优化问题,一旦卸载策略确定,就可以确定 UE 是否进行卸载以及卸载哪个 MEC 服务器来执行计算任务;第二阶段根据第一阶段得到的卸载策略,对 MEC 的计算资源和 UE 的上行传输功率进行了联合分配。

2.1 卸载决策的优化

在卸载决策的优化中,本小节设计了一种基于 DQN 的卸载决策算法来解决该子问题。在本场景中, DQN 的三要素(状态、动作和奖励函数)具体定义为:

(1) 状态。智能体通过从状态中提取有用信息进行学习,针对该模型,状态由用户的代价函数和超时率组成,其中超时即任务完成时延大于任务最大可容忍时延, $\frac{\eta}{N}$ 为超时率, η 为超时 UE 的数目,通过状态变化可以实时监控卸载策略优化的过程。

$$s = \left(\sum_{i=1}^N \sum_{j=1}^S z_i, \frac{\eta}{N} \right) \quad (12)$$

(2) 动作。此模型将用户卸载决策作为智能体的动作,决定任务的卸载位置。定义动作 $a_i = \{0, 1, \dots, S\}$ 表示用户的卸载决策。如果 UE 在本地执行,则采取动作 $a_i = 0$, 对应的卸载决策 $x_{i,j} = 0$; 如果 UE 卸载至 MEC 服务器 j 执行,则采取动作 $a_i = j$, 对应的卸载决策 $x_{i,j} = 1$ 。那么系统的动作空间为:

$$\mathcal{A} = [a_0, a_1, \dots, a_N] \quad (13)$$

(3) 奖励。该文的优化目标为最小化用户计算开销,所以此处的奖励函数设置为用户计算总开销的倒数,其中奖励函数中添加了惩罚项, φ 为惩罚因子,会对超时的 UE 给予一定的惩罚,且超时数越大则惩罚力度越大。

$$r = \frac{1}{\sum_{i=1}^N \sum_{j=1}^S z_i + \varphi \frac{\eta}{N}} \quad (14)$$

基于 DQN 的卸载决策算法的详细实现见算法 1。在此算法中,首先对网络参数进行初始化,包括经验池 D 的容量 G , 动作值函数 $Q(s, a; \theta)$ 的参数 θ 和目标动

作值函数 $Q(s, a; \theta^-)$ 的参数 $\theta^- = \theta$, 以及初始状态 s 。然后在每个 episode 中, 每个 UE 从可行动作空间 \mathcal{A} 中根据 Epsilon-Greedy 算法以概率 ε 选择一个随机动作或者一个满足 $a = \operatorname{argmax}_{a \in \mathcal{A}} Q(s, a; \theta)$ 的动作。

接下来根据 UE 选择的动作进行资源分配。如果 UE_i 选择本地执行任务, 则直接按照本地计算模型公式计算用户代价函数。如果 UE 选择将任务卸载至 MEC 服务器 j , 则令 UE_i 上行传输功率为最大上行传输功率, 由式 15 表示; MEC 计算资源按照式 16 分配, MEC_j 按任务属性 d_i 占有所有卸载至此 MEC 的任务计算量的比例为 UE_i 分配相应的计算资源。

$$p_i = P_i^{\max}, \forall i \in \mathcal{N} \quad (15)$$

$$f_{i,j}^c = \left(\frac{d_i}{\sum_{n \in \mathcal{N}_j} d_n} \right) f_m, \forall i \in \mathcal{N}_j \quad (16)$$

资源分配完成之后根据设定的奖励函数计算奖励 r , 得到新状态 s' 并将新样本 (s, a, r, s') 存入经验池 D 中并更新状态 $s = s'$ 。然后从经验池 D 中进行随机采样 (s_i, a_i, r_i, s_{i+1}) 形成小批量样本用于算法训练, 并使用这些小批量样本更新参数 θ , 目标 Q 网络采用了延时更新策略, 每隔 C 步更新其参数 $\theta^- = \theta$ 。

算法 1: 基于 DQN 的卸载决策算法。

1: 初始化经验池 D , 设置容量为 G , 动作值函数 Q 的参数 θ , 目标动作值函数 Q 的参数 $\theta^- = \theta$, 目标动作值函数的更新步长 C , UE 的初始状态 s 。

2: for episode = 1 to M do

3: for $n = 1: N$

4: if $\operatorname{rand}(0, 1) < \varepsilon$

5: UE_i 从动作空间 \mathcal{A} 根据 Epsilon-Greedy 算法随机选择动作 a

6: else

7: UE_i 选择满足 $a = \operatorname{argmax}_{a \in \mathcal{A}} Q(s, a; \theta)$ 的动作

8: end if

9: end for

10: 根据 UE 的动作按照公式 15、公式 16 分配相应的资源

11: for $n = 1: N$

12: 获得奖励 r 并得到新状态 s' , 将新样本 (s, a, r, s') 存入经验池 D 中

13: 更新状态 $s = s'$

14: 从经验池 D 中进行随机采样 (s_i, a_i, r_i, s_{i+1}) 形成小批量样本

15: 令 $y_i = r_i + \gamma \max_a (s_{i+1}, a; \theta^-)$

16: 对损失函数 $(y_i - Q(s_i, a_i; \theta))^2$ 关于参数 θ 执行梯度下降

17: 每隔 C 个步长更新一次目标网络参数 $\theta^- = \theta$

18: end for

19: end for

2.2 资源分配的优化

通过上述基于 DQN 的卸载决策算法得到每个用户的卸载方式后, 原目标函数可以化简为资源分配问题 P1:

$$\begin{aligned} \text{P1: } \min_{P, F} Z(P, F) &= \sum_{j=1}^S \sum_{i=1}^{N_j} w_i^t T_{i,j}^c + w_i^e E_{i,j}^c \\ \text{s. t. } (C_1) \quad &f_{i,j}^c > 0, \forall i \in \mathcal{N}, \forall j \in \mathcal{S} \\ (C_2) \quad &\sum_{i \in \mathcal{N}_j} f_{i,j}^c \leq f_m, \forall j \in \mathcal{S} \\ (C_3) \quad &0 \leq p_i \leq P_i^{\max}, \forall i \in \mathcal{N} \\ (C_4) \quad &T_{i,j}^c \leq T_i^{\max}, \forall i \in \mathcal{N}_j, \forall j \in \mathcal{S} \end{aligned} \quad (17)$$

此时只需要考虑卸载的用户, 具体地, 通过式 3、式 9, 可以将问题 P1 展开:

$$Z(P, F) = \sum_{j=1}^S \sum_{i=1}^{N_j} \frac{\eta_i}{f_{i,j}^c} + \frac{\alpha_i + \beta_i p_i}{\log_2(1 + I_{i,j} p_i)} \quad (18)$$

其中, $\eta_i = w_i^t d_i$, $\alpha_i = \frac{w_i^t \lambda_i}{W}$, $\beta_i = \frac{w_i^e \lambda_i}{W}$, $I_{i,j}$ 表示信道对任务 M_i 的有效干扰, 可以表示为:

$$I_{i,j} = \frac{g_{i,j}}{\sigma^2 + \sum_{n \in \mathcal{N} \setminus \{i\}} \sum_{s \in \mathcal{S} \setminus \{j\}} p_n g_{n,s}} \quad (19)$$

此时原资源分配问题分解为计算资源子问题 P2 和上行传输功率子问题 P3, 接下来依次对这两个子问题进行求解。

2.2.1 计算资源的优化

通过式 17, 计算资源分配子问题 P2 能被表述为:

$$\text{P2: } \min_F (F) = \sum_{j=1}^S \sum_{i=1}^{N_j} \frac{\eta_i}{f_{i,j}^c} \quad (20)$$

s. t. C_1, C_2

首先求问题 P2 关于 $f_{i,j}^c$ 的 Hessian 矩阵为:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial f_{1,j}^c} & \frac{\partial^2 f}{\partial f_{1,j}^c \partial f_{2,j}^c} & \cdots & \frac{\partial^2 f}{\partial f_{1,j}^c \partial f_{N,j}^c} \\ \frac{\partial^2 f}{\partial f_{2,j}^c \partial f_{1,j}^c} & \frac{\partial^2 f}{\partial f_{2,j}^c} & \cdots & \frac{\partial^2 f}{\partial f_{2,j}^c \partial f_{N,j}^c} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial f_{N,j}^c \partial f_{1,j}^c} & \frac{\partial^2 f}{\partial f_{N,j}^c \partial f_{2,j}^c} & \cdots & \frac{\partial^2 f}{\partial f_{N,j}^c} \end{bmatrix} \quad (21)$$

其中, 矩阵的每一项具体表示为:

$$\frac{\partial^2 f}{\partial f_{i,j}^c \partial f_{n,j}^c} = \begin{cases} \frac{2d_{n_i}}{(f_{i,j}^c)^3}, & i = n \\ 0, & \text{其他} \end{cases} \quad (22)$$

从式 22 表示的二阶导数可以看出, 式 21 中的 Hessian 矩阵中对角线元素均为正数 (即 Hessian 矩阵的每个特征值都大于 0), 因此 Hessian 矩阵是一个对称正定矩阵, 可以得出 $f(F)$ 是一个凸函数。又由于约束是线性的, 可以得出该问题是一个凸优化问题, 可以通过构造拉格朗日函数和 KKT 条件来求解, 那么

$f(\mathbf{F})$ 的最优计算资源分配 $f_{i,j}^*$ 和对应的最优目标函数值 $f(\mathbf{F}^*)$ 表示为:

$$f_{i,j}^* = \frac{f_m \sqrt{\eta_i}}{\sum_{i \in N_j} \sqrt{\eta_i}}, \forall i \in N_j, \forall j \in \mathcal{S} \quad (23)$$

$$f(\mathbf{F}^*) = \sum_{j \in \mathcal{S}} \frac{1}{f_m} \left(\sum_{i \in N_j} \sqrt{\eta_i} \right)^2 \quad (24)$$

2.2.2 上行传输功率的优化

卸载用户的上行传输功率分配子问题 P3 能被表述为:

$$\text{P3:} \min_p \sum_{j=1}^S \sum_{i=1}^{N_j} \frac{\alpha_i + \beta_i p_i}{\log_2(1 + I_{i,j} p_i)} \quad (25)$$

s. t. C_3, C_4

首先,找到一个有效干扰 $I_{i,j}$ 的近似值 $\hat{I}_{i,j}$:

$$\hat{I}_{i,j} = \frac{g_{i,j}}{\sigma^2 + \sum_{n \in \mathcal{N}' \setminus \{i\}} \sum_{s \in \mathcal{S}' \setminus \{j\}} P_n^{\max} g_{n,s}} \quad (26)$$

其中, $\hat{I}_{i,j}$ 是 $I_{i,j}$ 一个较为准确的估计值^[17-18],因为该文的卸载决策是倾向于选择合适的用户与 SBS 进行关联,所以 $\hat{I}_{i,j}$ 的值应该是较小的,同时这也意味 $I_{i,j}$ 的变化不应该导致 $r_{i,j}$ 会有很大的偏差。

然后,将约束 C_4 进一步推导:

$$C_4: T_{i,j}^* = \frac{\lambda_i}{r_{i,j}} + \frac{d_i}{f_{i,j}} \leq T_i^{\max} \Rightarrow p_i \geq \frac{2^{\frac{\lambda_i}{w\sigma_i}} - 1}{I_{i,j}} \quad (27)$$

$\forall i \in N_j, \forall j \in \mathcal{S}$

其中, $\sigma_i = T_i^{\max} - \frac{d_i}{f_{i,j}}$ 。通过约束 C_3 和新的约束 C_4' 以及将 $I_{i,j}$ 替换成 $\hat{I}_{i,j}$,可以得到用户 i 的上行传输功率范围为 $\frac{2^{\frac{\lambda_i}{w\sigma_i}} - 1}{\hat{I}_{i,j}} \leq p_i \leq P_i^{\max}$,因此可以将问题 P3 重新改写为:

$$\text{P3:} \min_p \sum_{j=1}^S \sum_{i=1}^{N_j} \frac{\alpha_i + \beta_i p_i}{\log_2(1 + \hat{I}_{i,j} p_i)} \quad (28)$$

s. t. $\frac{2^{\frac{\lambda_i}{w\sigma_i}} - 1}{\hat{I}_{i,j}} \leq p_i \leq P_i^{\max}$

为了方便起见,令:

$$\psi(p_i) = \frac{\alpha_i + \beta_i p_i}{\log_2(1 + \hat{I}_{i,j} p_i)} \quad (29)$$

该文采用黄金分割法来求解问题 P3。具体步骤如算法 2 所示。首先输入用户上行传输功率的初始搜索范围 $[a_i, b_i]$ 和收敛精度 ε , 然后计算 p_i^l, p_i^r 之差的绝对值。当绝对值大于收敛精度时,通过式 29 计算 $\psi(p_i^l)$ 和 $\psi(p_i^r)$ 并比较。如果 $\psi(p_i^l) > \psi(p_i^r)$, 则搜索空间向左减小;如果 $\psi(p_i^l) < \psi(p_i^r)$, 则搜索空间向

右减小,最后输出最优上行传输功率 p_i^{opt} 。该方法通过不断减小搜索空间来寻找用户最优上传功率。

算法 2: 基于黄金分割法的上行传输功率分配算法。

1: 输入初始搜索区间 $[a_i, b_i]$, $a_i = \frac{2^{\frac{\lambda_i}{w\sigma_i}} - 1}{\hat{I}_{i,j}}$, $b_i = P_i^{\max}$,

收敛精度 ε

2: 计算初始试探点 $p_i^l = b_i - 0.618(b_i - a_i)$, $p_i^r = a_i + 0.618(b_i - a_i)$

3: while $|p_i^r - p_i^l| \geq \varepsilon$ do

4: 通过式 29 计算 $\psi(p_i^l)$ 和 $\psi(p_i^r)$

5: if $\psi(p_i^l) > \psi(p_i^r)$ then

6: $a_i = p_i^l, p_i^l = p_i^r$

7: $\psi(p_i^l) = \psi(p_i^r), p_i^r = a_i + 0.618(b_i - a_i)$

8: else

9: $b_i = p_i^r, p_i^r = p_i^l$

10: $\psi(p_i^r) = \psi(p_i^l), p_i^l = b_i - 0.618(b_i - a_i)$

11: end if

12: end while

13: 输出最优解 $p_i^{\text{opt}} = (p_i^l + p_i^r)/2$

3 仿真结果及性能分析

本小节通过使用 Matlab 和 python 仿真软件对所提算法进行仿真分析,其中仿真部分采用 tensorflow 1.14 版本,数据处理及画图部分采用 Matlab。考虑的一个覆盖范围为 500 m×500 m 的超密集网络,设置 SBS 数量为 4,相邻的基站距离为 50 米,并且每个 SBS 中心部署一台 MEC 服务器,其中 UE 随机分布在该网络覆盖范围内。系统带宽为 20 MHz, SBS 和 UE 之间的信道增益模型服从基于距离的路径损失模型 $127 + 30\log d$ 。一般的,设置 $w_i^t = 0.8$, $w_i^e = 0.2$ 。仿真相关参数如表 1 所示。

表 1 仿真参数

参数	参数值
能量系数 ρ^l	5×10^{-27}
系统总带宽 B / MHz	20
子信道数目 K	20
噪声功率 σ^2 / dBm	-100
最大传输功率 P_i^{\max} / W	0.5
本地 CPU 计算能力 f_i^l / GHz	1
MEC 最大计算能力 f_m / GHz	10
任务所需 CPU 周期数 d_i / Gigacycle	0.5 ~ 2.5
任务数据量 λ_i / MB	0.1 ~ 0.5
最大容忍时延 T_i^{\max} / s	1 ~ 2

对于 DQN 的模型参数,不同参数的设置会对算法最终的性能产生巨大的影响。例如 ε 贪婪策略的设置是为了避免 DQN 模型陷入局部最优,如果概率设

置过大,就会导致 DQN 过多地探索未知的动作,减低了从经验池中选择动作的概率,将会导致算法收敛速度变慢甚至无法收敛;如果概率设置过小,就会导致模型陷入局部最优。所以设置 $\varepsilon = 0.9$,奖励因子 $\gamma = 0.9$,学习率设置为 0.001。

首先分析公式 3、公式 9 中时延权重和能耗权重对于算法的影响。

图 2 描述的是当设置不同的时延权重 w_t^i 时,不同数量的 UE 的平均计算时延的变化,其中 UE 数量设置为 [10, 20, 30]。从图中可以观察到,随着时延权重的增大,任务的平均计算时延不断降低,不同数量的 UE 都符合该趋势,并且 UE 数量越大,平均计算时延越大。这是因为时延权重决定了 UE 对于计算时延的要求,时延权重越大,UE 对于计算时延的要求越苛刻。

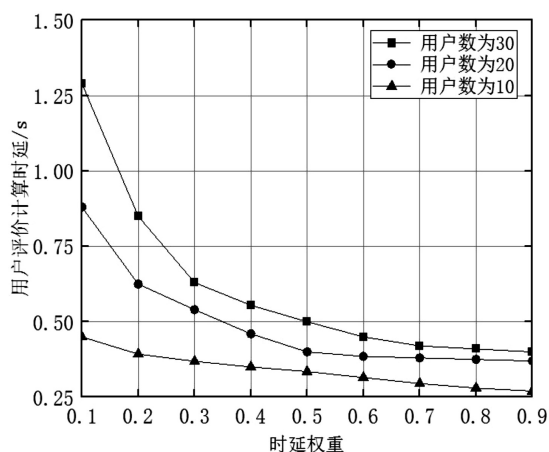


图2 时延权重对用户平均计算时延的影响

图 3 描述的与图 2 类似,表示设置不同的能耗权重 w_e^i 时,不同数量 UE 的平均计算能耗的变化趋势。图 3 表明,平均计算能耗随着时延权重的增大而降低。这是因为能耗权重体现了 UE 对于计算能耗的要求,且权重越大用户越希望能够降低计算能耗,提高设备使用时间。

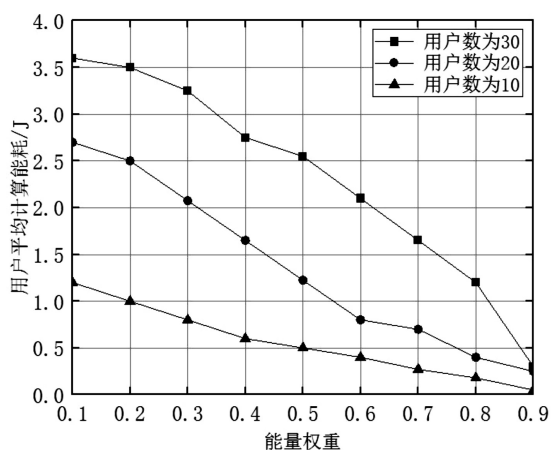


图3 能量权重对用户平均计算能耗的影响

图 4 为所提算法与其他算法在不同用户数下的用

户计算开销比较。其中比较的算法为本地计算策略、全部卸载策略和随机卸载策略。从图 4 中可以观察到,所提算法的用户计算总开销比其余三种算法的低。并且当用户数较少时,全部卸载执行算法和所提算法在计算开销上大致相同,但随着用户不断增加,全部卸载执行的计算开销迅速增加,这是由于多个用户选择卸载,产生不同小基站之间的用户通信干扰,卸载至同一个 MEC 服务器的用户也会产生计算资源的竞争。而所提算法综合优化了用户的卸载决策、MEC 计算资源以及用户的上行传输功率,能够在资源有限的情况下,对任务卸载和资源分配做出合理的优化,使得 MEC 能够为不同任务属性的卸载任务分配最优的资源,用户也能根据自身情况设置最优的上行传输功率,有效降低了任务传输能耗,所以所提算法在系统能效方面具有出色的表现。

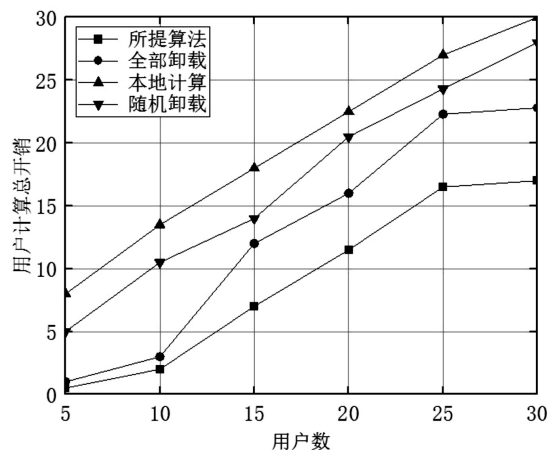


图4 用户数对用户计算总开销的影响

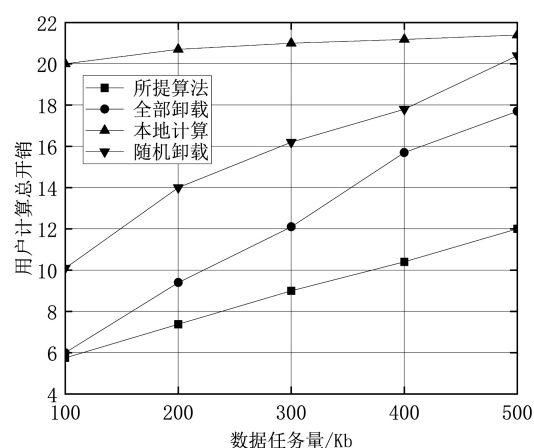


图5 任务数据量对用户计算总开销的影响

图 5 描述的是任务不同的计算量对用户计算总开销的影响。从图中可以看出,除了本地计算策略,全部卸载策略、随机卸载策略以及该文所提策略的计算开销都随着任务数据量的增大而递增,而本地计算的计算开销主要与任务所需的计算量有关,所以在任务计算量一定的前提下,本地计算的计算开销维持不变。

而其他卸载方案,用户需要进行任务卸载,任务传输速率与任务的数据量有关,所以随着任务数据量的增大,其用户计算开销不断增大。同时,所提算法的计算开销是所有方案中最好的,这是因为所提算法会根据任务的属性制定不同的卸载决策,同时也会对用户传输功率进行了优化,从而有效地减少了计算开销。

4 结束语

该文考虑了多用户多 MEC 的超密集网络场景下的联合任务卸载和资源分配的优化方案。首先描述了多用户多 MEC 的系统模型,然后以最小化用户计算总开销为优化目标,确定最佳的卸载决策和最优的资源分配策略。由于优化目标是一个 MINLP,考虑将目标函数进行子问题分解。其中,设计了一种基于 DQN 的任务卸载决策来解决任务卸载子问题,然后分别使用 KKT 条件以及黄金分割算法解决 MEC 计算资源分配和上行传输功率的优化问题。仿真结果表明,所提算法可以根据权重因子的不同,准确地选择降低时延或者节省能耗。除此之外,还可以有效地降低用户计算的总开销,且性能明显优于其他算法。

参考文献:

- [1] 朱雪田,夏旭,齐飞. 5G 网络关键技术和业务[J]. 电子技术应用,2018,44(9):1-4.
- [2] 杨鑫,时晓厚,沈云,等. 5G 工业互联网的边缘计算技术架构与应用[J]. 电子技术应用,2019,45(12):25-28.
- [3] ORSINI G, BADE D, LAMERSDORF W. CloudAware: a context-adaptive middleware for mobile edge and cloud computing applications [C]//2016 IEEE 1st international workshops on foundations and applications of self * systems (FAS * W). Augsburg: IEEE, 2016: 216-221.
- [4] ZHOU Y, FADLULLAH Z, MAO B, et al. A deep-learning-based radio resource assignment technique for 5G ultra dense networks[J]. IEEE Network, 2018, 32(6): 28-34.
- [5] TENG Y, LIU M, YU F R, et al. Resource allocation for ultra-dense networks: a survey, some research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2019, 21(3): 2134-2168.
- [6] 杨戈,张衡. 面向移动边缘计算基于强化学习的计算卸载算法[J]. 电子技术应用, 2021, 47(2): 36-40.
- [7] WANG Z, DU H, YE Q. HTR: a joint approach for task offloading and resource allocation in mobile edge computing [C]//ICC 2021 - IEEE international conference on communications. Montreal: IEEE, 2021: 1-6.
- [8] WANG J, FENG D, ZHANG S, et al. Joint computation offloading and resource allocation for MEC-enabled IoT systems with imperfect CSI[J]. IEEE Internet of Things Journal, 2020, 8(5): 3462-3475.
- [9] CHEN X, LIU Z, CHEN Y, et al. Mobile edge computing based task offloading and resource allocation in 5G ultra-dense networks [J]. IEEE Access, 2019, 7(6): 184172-184182.
- [10] CHEN M, HAO Y. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597.
- [11] 杨卓. 移动边缘计算网络中基于资源联合配置的计算卸载策略研究[D]. 重庆: 重庆大学, 2021.
- [12] 刘耀,何岳园,周红静,等. 移动边缘计算中基于资源联合分配的部分计算卸载方法[J]. 物联网学报, 2023, 7(1): 140-148.
- [13] 辜航,张敏娟,潘钰文. 移动边缘计算中任务卸载和资源分配算法优化[J]. 现代电子技术, 2023, 46(7): 67-72.
- [14] ZHANG J, XIA W, YAN F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing [J]. IEEE Access, 2018, 6: 19324-19337.
- [15] ZHANG W, WEN Y, GUAN K, et al. Energy-optimal mobile cloud computing under stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2013, 12(9): 4569-4581.
- [16] GUO F, ZHANG H, JI H, et al. Energy efficient computation offloading for multi-access MEC enabled small cell networks [C]//2018 IEEE international conference on communications workshops (ICC workshops). Kansas City: IEEE, 2018: 1-6.
- [17] DU Y, DE VECIANA G. "Wireless networks without edges": dynamic radio resource clustering and user scheduling [C]//IEEE INFOCOM 2014 - IEEE conference on computer communications. Toronto: IEEE, 2014: 1321-1329.
- [18] PHAM Q V, LEANH T, TRAN N H, et al. Decentralized computation offloading and resource allocation for mobile-edge computing: a matching game approach[J]. IEEE Access, 2018, 6: 75868-75885.